Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon

Christina Boura, María Naya-Plasencia & Valentin Suder

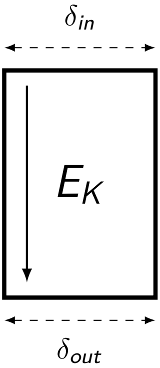
ASIACRYPT 2014, Kaohsiung, Taiwan

December 8th, 2014

informatics mathematics

UNIVERSITÉ DE VERSAILLES SAINT-QUENTIN-EN-YVELINES

Impossible Differential Cryptanalysis



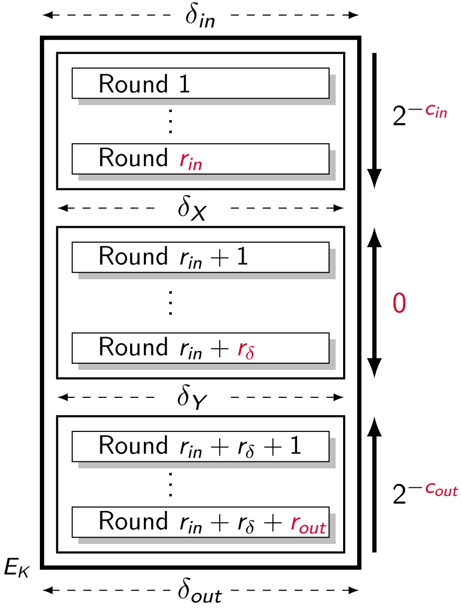
Impossible Differential Cryptanalysis:

- was introduced by **Knudsen** in 1998, and **Biham**, **Biryukov** & **Shamir** in 1999;
- is part of the **Differential Cryptanalysis** family...
- ...but uses a **distinguisher** of probability 0;
- is very efficient against **iterated block ciphers**.

L. R. Knudsen,
DEAL – A 128-bit cipher,
1998.

E. Biham, A. Biryukov and A. Shamir,
Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials,
EUROCRYPT'99.

Impossible Differential Cryptanalysis: Scenario



- **place** an impossible differential (δ_X, δ_Y) on r_δ rounds;
- **extend** it by differentials $(\delta_{in} \to \delta_X)$ and $(\delta_{out} \to \delta_Y)$;
- **evaluate** the parameters:

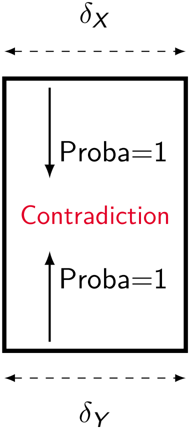
r_{in}, r_{out}: **number** of rounds

c_{in}, c_{out}: log of the **probabilities**

k_{in}, k_{out}: involved **subkeys**

$|k_{in} \cup k_{out}|$: key **entropy**

Finding an Impossible Differential



- Miss-in-the-middle technique [BBS99];
- $\mathcal{U}$-method [Kim *et al.* 03];

J. Kim and S. Hong and J. Sung and C. Lee and S. Lee,
Impossible Differential Cryptanalysis for Block Cipher Structures,
INDOCRYPT'03.

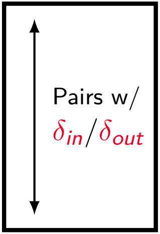
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA'08.

Early-Abort Technique

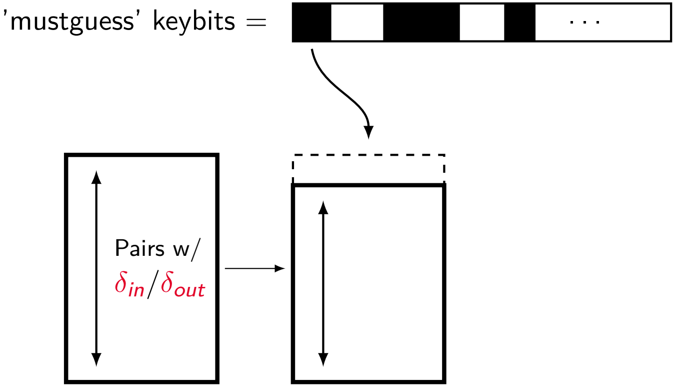
J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA'08.

'mustguess' keybits =



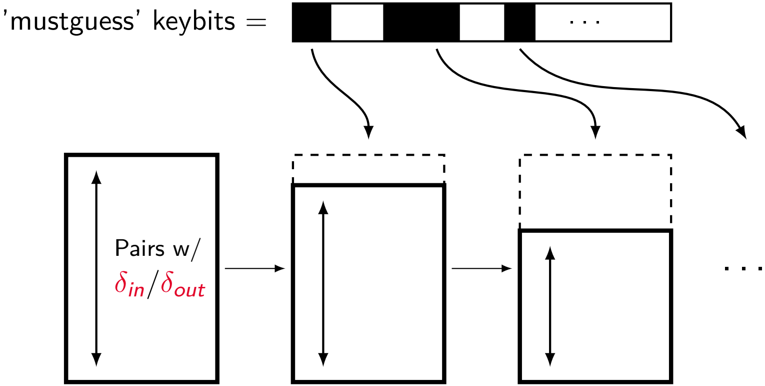
Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA'08.



Early-Abort Technique

J. Lu, J. Kim, N. Keller and O. Dunkelman,
Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1,
CT-RSA'08.



Existing Flaws

Algorithm	Ref.	Type	Gravity
CLEFIA-128	[ZH08]	data	✗
CLEFIA-128	[T10]	unverifiable	-
Camellia	[WZF07]	big flaw	✗
Camellia-128	[WZZ08]	big flaw	✗
Camellia	[LKKD08]	small flaws	✓
LBlock	[MN1208]	small flaw	✓
SIMON	[ALLW13,14]	big flaw	✗
SIMON	[AL13]	data	✗

Objectives

- **Formalize** the evaluation of the complexities;
- **Automate** the whole process;

Objectives

- **Formalize** the evaluation of the complexities;
- **Automate** the whole process;

Results

- **Optimization** of previous attacks;
- **Development** of new techniques;
- **Application** to block ciphers (CLEFIA, Camellia, LBlock, SIMON)
  $\Rightarrow$ Best Cryptanalysis.

Amount of Memory needed

$$\frac{1}{2^{c_{in}+c_{out}}}$$

Amount of Memory needed

$$\left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)$$

Amount of Memory needed

$$\mathcal{P} = \left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)^{N} \quad < \quad \frac{1}{2^{|k_{in} \cup k_{out}|}}$$

Amount of Memory needed

$$\mathcal{P} = \left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)^{N} \quad < \quad \overset{\frac{1}{2}}{\cancel{\frac{1}{2^{|k_{in} \cup k_{out}|}}}}$$

Amount of Memory needed

$$\mathcal{P} = \left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)^N < \frac{1}{2}$$

$$\cancel{\frac{1}{2^{|k_{in} \cup k_{out}|}}}$$

Since $\mathcal{P} \simeq e^{-N(2^{-(c_{in}+c_{out})})}$, we will **consider** that $N_{\min} = 2^{c_{in}+c_{out}}$.

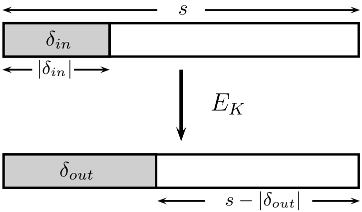
Amount of Memory needed

$$\mathcal{P} = \left(1 - \frac{1}{2^{c_{in}+c_{out}}}\right)^N < \cancel{\frac{1}{2^{|k_{in} \cup k_{out}|}}} \ \frac{1}{2}$$

Since $\mathcal{P} \simeq e^{-N(2^{-(c_{in}+c_{out})})}$, we will **consider** that $N_{\min} = 2^{c_{in}+c_{out}}$.

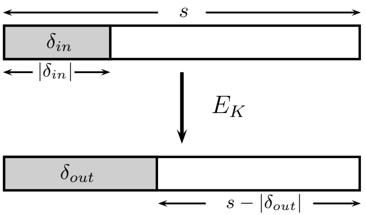
Memory Complexity: $\min\left\{\mathbf{N}, 2^{|k_{in} \cup k_{out}|}\right\}$.

Amount of Data needed



- **To build** these N pairs, we **need** $C_N < 2^s$ plaintexts.

Amount of Data needed



- **To build** these N pairs, we **need** $C_N < 2^s$ plaintexts.

Data Complexity: C_N.

$$C_N = \max\left\{ \min_{\delta \in \{\delta_{in}, \delta_{out}\}} \left\{ \sqrt{N 2^{s+1-|\delta|}} \right\}, N 2^{s+1-|\delta_{in}|-|\delta_{out}|} \right\} < 2^s.$$

Time Complexity

$$T_{comp} = C_N C_E +$$

- **Encrypt** all the data;

Time Complexity

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E$$

- **Encrypt** all the data;
- *Early-Abort* Technique

Time Complexity

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E$$

- **Encrypt** all the data;
- *Early-Abort* Technique
  - Check **each** key step by step;

Time Complexity

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E$$

- **Encrypt** all the data;
- *Early-Abort* Technique
  - Check **each** key step by step;
  - Decrease the number of pairs in the list;

Time Complexity

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}}\right) C'_E + \frac{2^{|K|}}{2^{|k_{in} \cup k_{out}|}} \mathcal{P} 2^{|k_{in} \cup k_{out}|} C_E.$$

- **Encrypt** all the data;
- *Early-Abort* Technique
  - Check **each** key step by step;
  - Decrease the number of pairs in the list;
- Test every key remaining in the **candidate key set**

Time Complexity

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}}\right) C'_E + \frac{2^{|K|}}{2^{|k_{in} \cup k_{out}|}} \mathcal{P} 2^{|k_{in} \cup k_{out}|} C_E.$$

- **Encrypt** all the data;
- *Early-Abort* Technique
  - Check **each** key step by step;
  - Decrease the number of pairs in the list;
- Test every key remaining in the **candidate key set**

$$T_{comp} < 2^{|K|} C_E.$$

Uniformized Formulas

$$T_{comp} = C_N C_E + \left(2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E + \mathcal{P} 2^{|K|} C_E .$$

⇒ **easy to use** formulas;

⇒ more **trade-offs**;

⇒ **automatic** tool & **systematic** search;

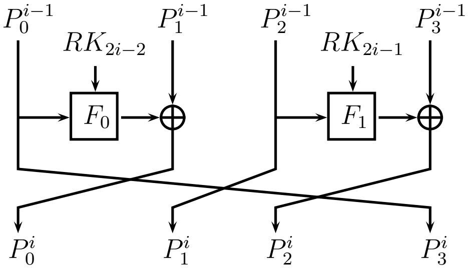
⇒ development of **new techniques**;

New Techniques

- **Multiple Impossible Differentials**
- *State-Test* **Technique**

Example of an Application to **CLEFIA**

- block size:
  $4 \times 32 = 128$ bits
- key size:
  128, 192, 256 bits
- # of rounds:
  18, 22, 26

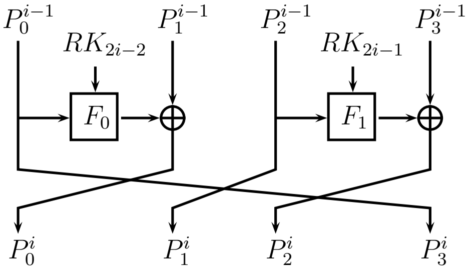


T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata,
The 128-Bit Blockcipher CLEFIA (Extended Abstract),
FSE'07.

Multiple Impossible Differentials

Formalize the idea of [Tsunoo et al. 08]

$\delta_X \nrightarrow_9 \delta_Y$	
δ_X	δ_Y
$(0,0,0,A)$	$(0,0,0,B)$
$(0,A,0,0)$	$(0,B,0,0)$

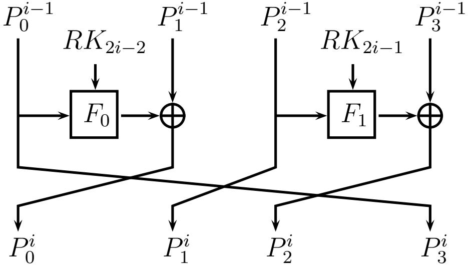


A	B		
$(0,0,0,\alpha)$	$(0,0,\beta,0)$	$(0,\beta,0,0)$	$(\beta,0,0,0)$
$(0,0,\alpha,0)$	$(0,0,0,\beta)$	$(0,\beta,0,0)$	$(\beta,0,0,0)$
$(0,\alpha,0,0)$	$(0,0,0,\beta)$	$(0,0,\beta,0)$	$(\beta,0,0,0)$
$(\alpha,0,0,0)$	$(0,0,0,\beta)$	$(0,0,\beta,0)$	$(0,\beta,0,0)$

Multiple Impossible Differentials

Formalize the idea of [Tsunoo et al. 08]

$\delta_X \nrightarrow_9 \delta_Y$	
δ_X	δ_Y
$(0,0,0,A)$	$(0,0,0,B)$
$(0,A,0,0)$	$(0,B,0,0)$

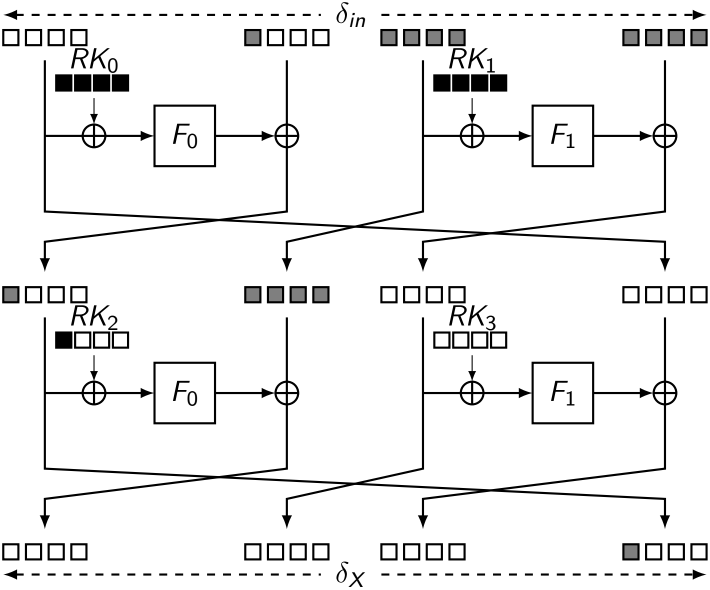


A	B		
$(0,0,0,\alpha)$	$(0,0,\beta,0)$	$(0,\beta,0,0)$	$(\beta,0,0,0)$
$(0,0,\alpha,0)$	$(0,0,0,\beta)$	$(0,\beta,0,0)$	$(\beta,0,0,0)$
$(0,\alpha,0,0)$	$(0,0,0,\beta)$	$(0,0,\beta,0)$	$(\beta,0,0,0)$
$(\alpha,0,0,0)$	$(0,0,0,\beta)$	$(0,0,\beta,0)$	$(0,\beta,0,0)$

$$C_N = 2^{113} \quad \Rightarrow \quad C_N = 2^{113-\log_2(24)}$$

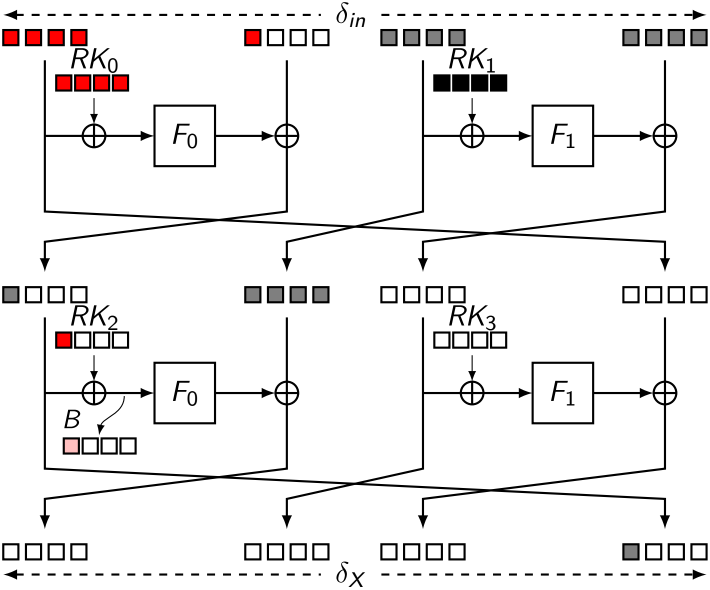
State-Test Technique

Decrease the number of key bits to guess



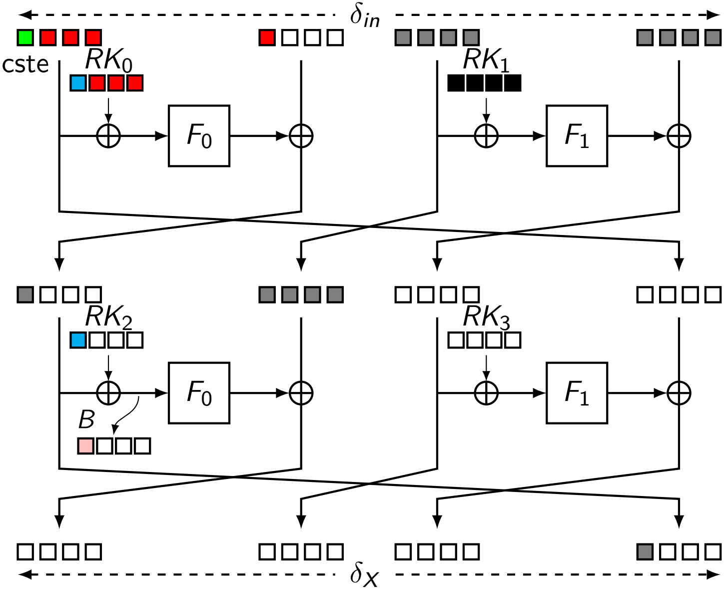
State-Test Technique

Decrease the number of key bits to guess



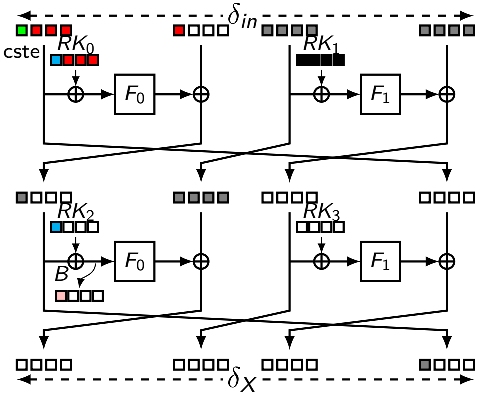
State-Test Technique

Decrease the number of key bits to guess



State-Test Technique

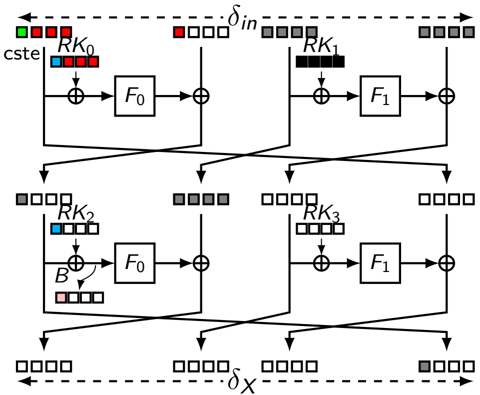
Decrease the number of key bits to guess



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

State-Test Technique

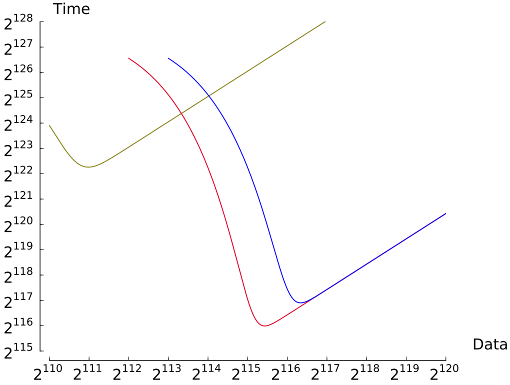
Decrease the number of key bits to guess



$$B' = B \bigoplus \blacksquare \quad = \quad \blacksquare \oplus S_0(\blacksquare \oplus \blacksquare).$$

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

Comparison



Algorithm	Rounds	Time	Data	Memory	Ref.
CLEFIA-128	13/18	121.2	117.8	86.8	[MDS11]
state-test	13/18	116.90	116.33	83.33	
multiple	13/18	122.26	**111.02**	**82.60**	
multiple & *state-test*	13/18	**116.16**	**115.38**	**83.16**	

Camellia

128-bit block

Algorithm	Rounds	Time	Data	Memory	Ref.
Camellia-128	11/18	122	122	98	[LLGWLCL12]
	11/18	**118.43**	**118.4**	**92.4**	
Camellia-192	12/24	187.2	123	155.41	[LLGWLCL12]
	12/24	**161.06**	**119.70**	**150.70**	
Camellia-256	13/24	251.1	123	203	[LLGWLCL12]
	13/24	**225.06**	**119.71**	**198.71**	
Camellia-256 †	14/24	250.5	120	125	[LLGWLCL12]
state-test	14/24	**220**	**118**	173	

LBlock

64-bit block, 80-bit key

Algorithm	Rounds	Time	Data	Memory	Ref.
LBlock	22/32	79.28	58	72.67	[KDH12]
	22/32	**71.53**	60	**59**	
	23/32	74.06	59.6	74.6	

SIMON

Algorithm	Rounds	Time	Data	Memory
SIMON-32/64	**19**/32	62.56	32	44
SIMON-48/72	**20**/36	70.69	48	58
SIMON-48/96	**21**/36	94.73	48	70
SIMON-64/96	**21**/42	94.56	64	60
SIMON-64/128	**22**/44	126.56	64	75
SIMON-96/96	**24**/52	94.62	94	61
SIMON-96/144	**25**/54	142.59	96	77
SIMON-128/128	**27**/68	126.6	126	61
SIMON-128/192	**28**/69	190.56	128	77
SIMON-128/256	**30**/72	254.68	128	111

Perspectives

- Extend results to **Substitution Permutation Network** ciphers (AES,. . .);
- Generalize the *State-test* technique;