UNIVERSITÄT PASSAU

Multi-Stage Fault Attacks

Applications to the Block Cipher PRINCE

Philipp Jovanovic

Department of Informatics and Mathematics
University of Passau

March 27, 2013

Outline

1. Motivation
2. The PRINCE Block Cipher
3. A Multi-Stage Fault Attack on PRINCE

Cryptology

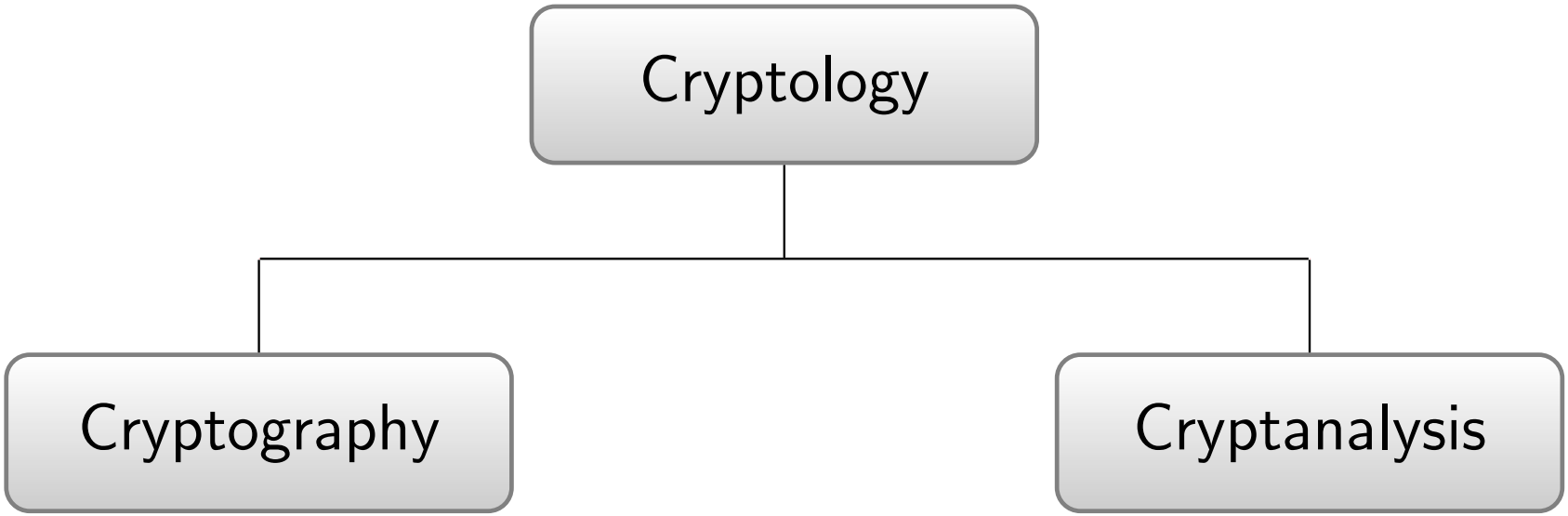


Figure: Overview on the field of cryptology.

Fields of Cryptanalysis

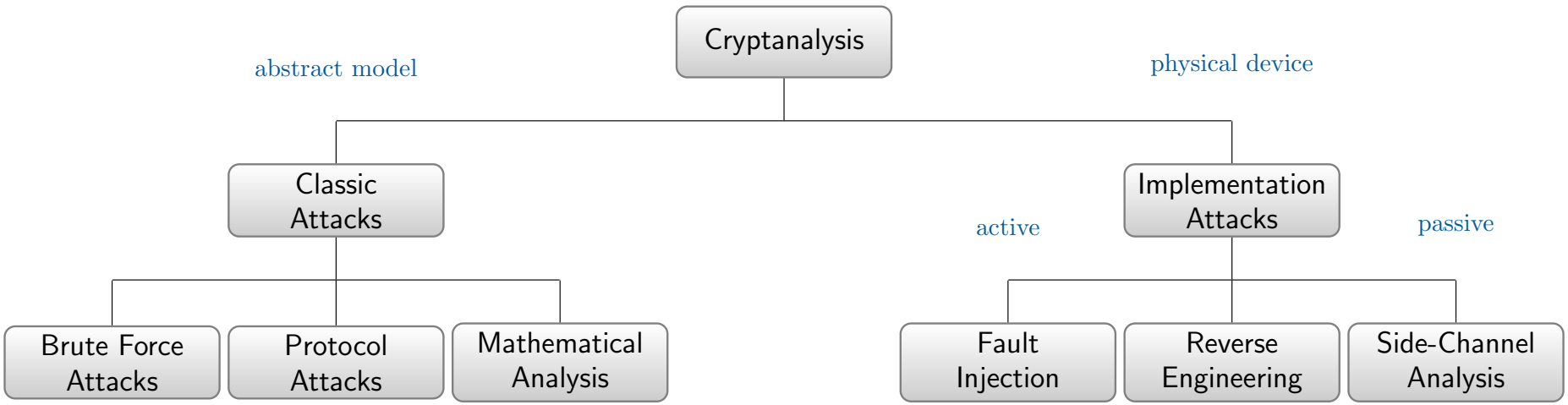


Figure: Overview on the different fields of cryptanalysis.

UNIVERSITÄT PASSAU

At a Glance ...

At a Glance ...


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!

At a Glance ...

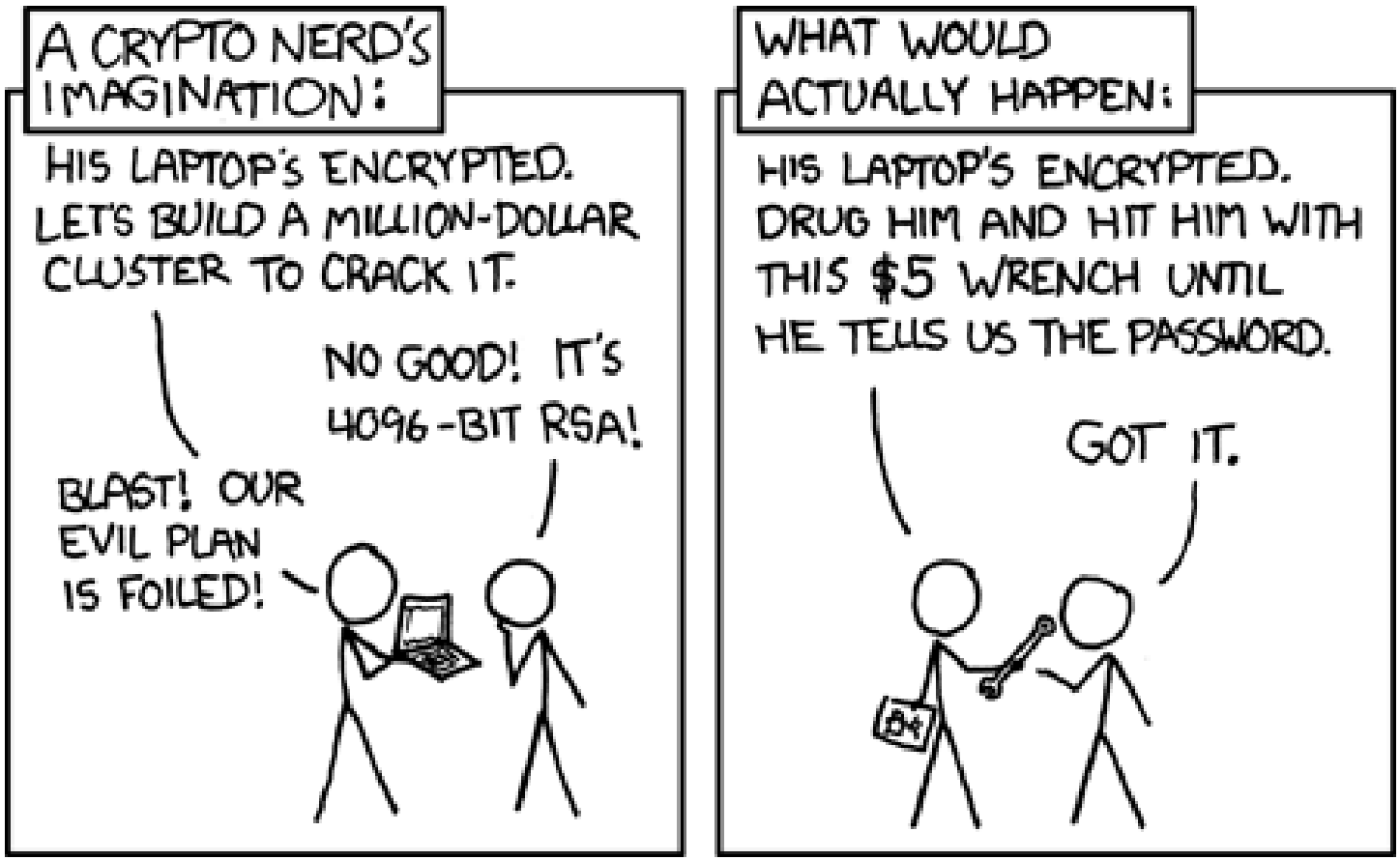


Figure: http://xkcd.com/538

Fields of Cryptanalysis

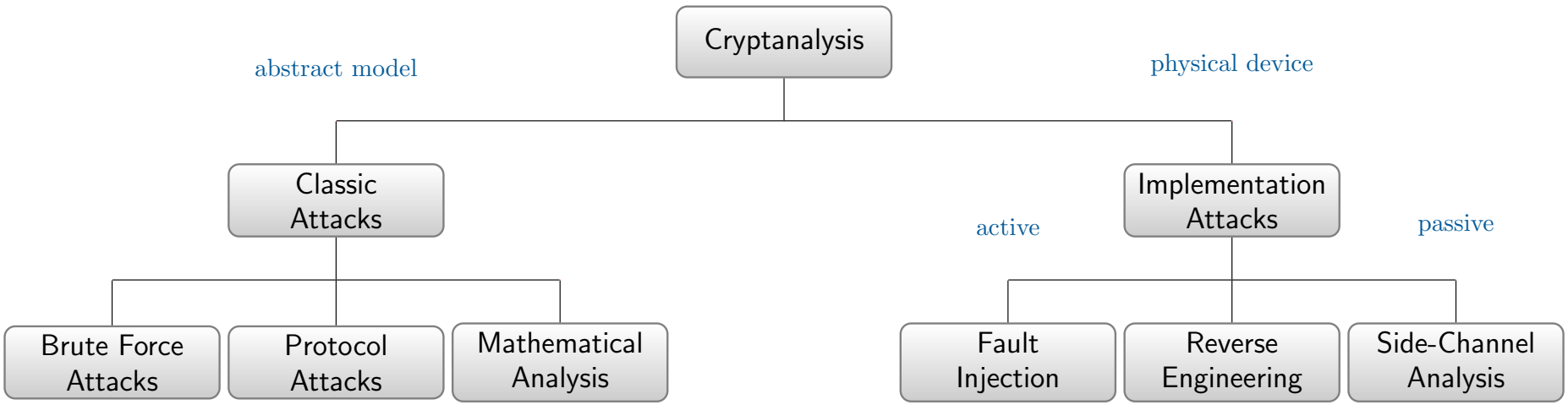


Figure: Overview on the different fields of cryptanalysis.

Fields of Cryptanalysis

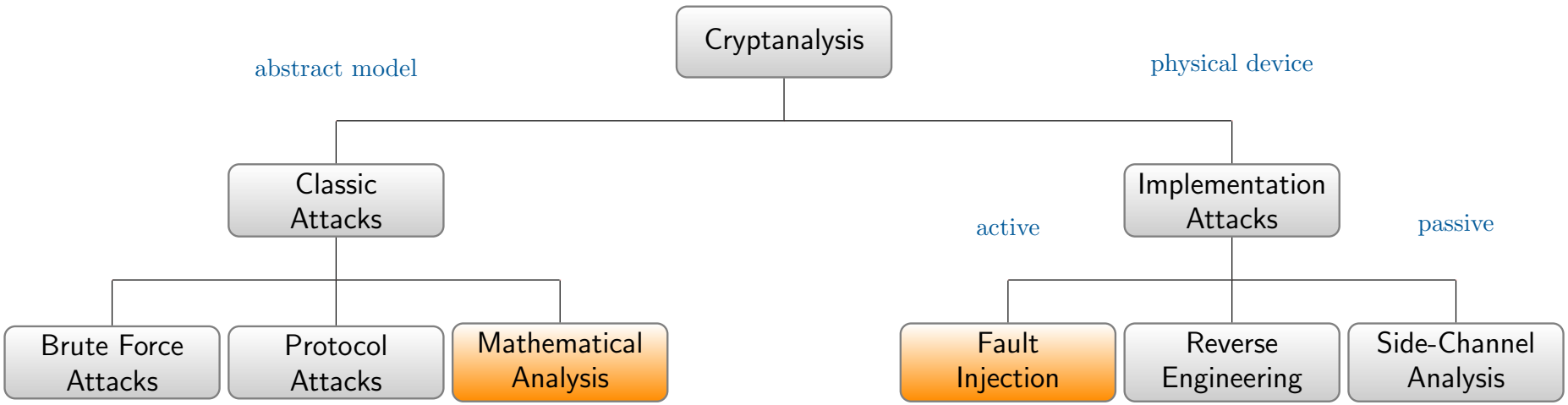
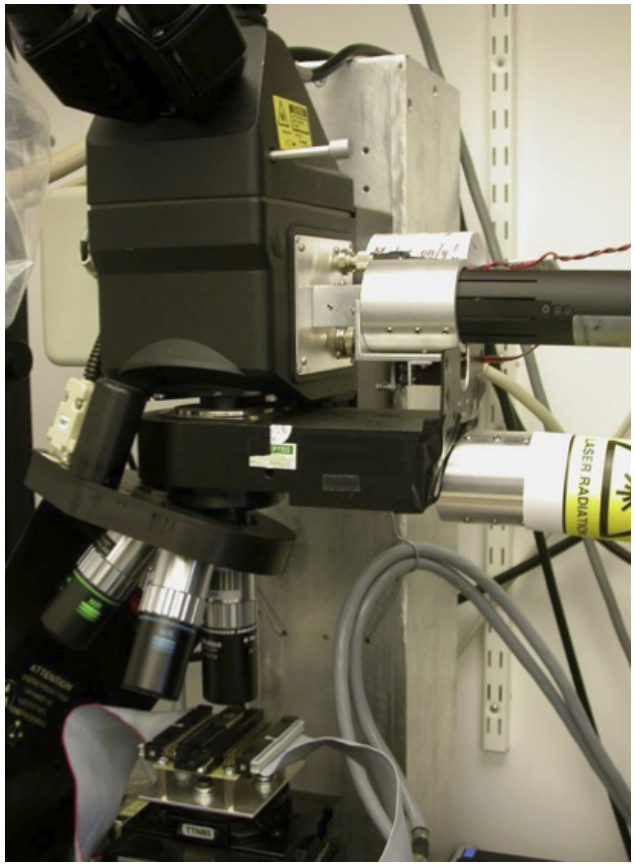


Figure: Overview on the different fields of cryptanalysis.

Fault Attacks

Characteristics

- First appearances in 1998* and 2001†.
- By injecting faults into the electronical circuit, the attacker tries to extract secret informations from the latter (e.g. a key).
- Lead to powerful new attack techniques and chip manufacturers were forced to rethink their designs.

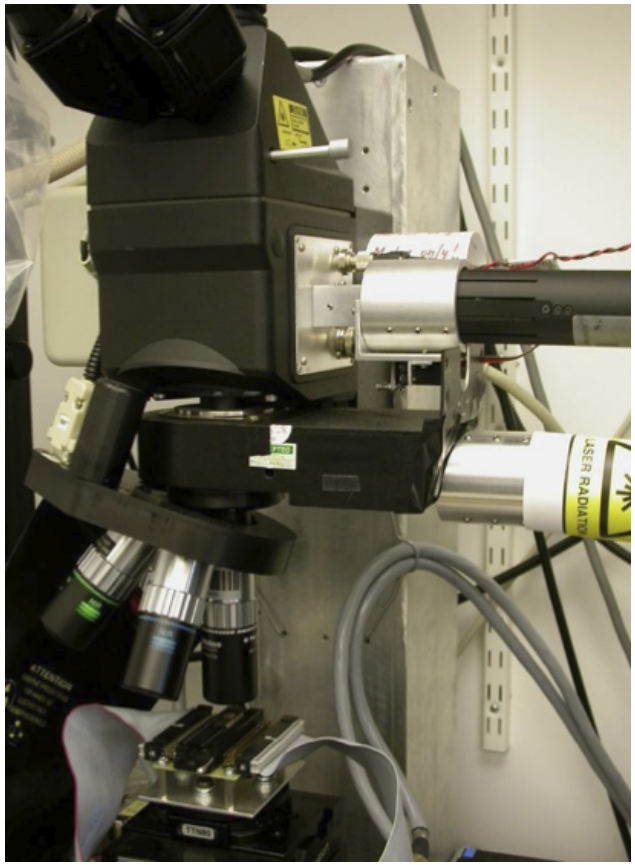


* E. Biham and A. Shamir, Differential Fault Analysis of Secret Key Cryptosystems, In: Burton S. Kaliski Jr. (ed.) *CRYPTO* 1997, LNCS, vol. **1294**, Springer Heidelberg 1997, pp. 513–525.

† D. Boneh, R.A. Demillo, R.J. Lipton, On the Importance of Eliminating Errors in Cryptographic Computations, Journal of Cryptology **14** (2001), 101–119.

Characteristics

- First appearances in 1998* and 2001†.
- By injecting faults into the electronical circuit, the attacker tries to extract secret informations from the latter (e.g. a key).
- Lead to powerful new attack techniques and chip manufacturers were forced to rethink their designs.



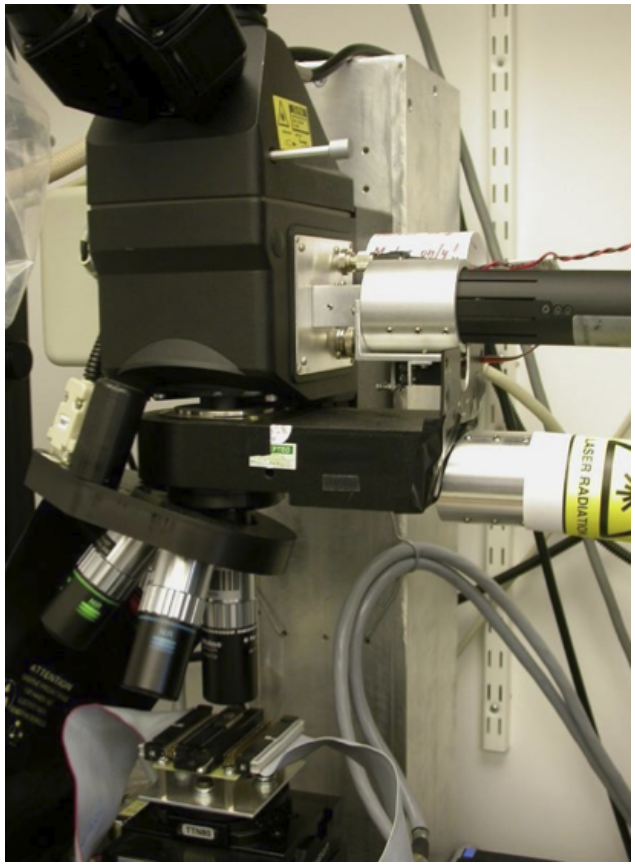
* E. Biham and A. Shamir, Differential Fault Analysis of Secret Key Cryptosystems, In: Burton S. Kaliski Jr. (ed.) *CRYPTO* 1997, LNCS, vol. **1294**, Springer Heidelberg 1997, pp. 513–525.

† D. Boneh, R.A. Demillo, R.J. Lipton, On the Importance of Eliminating Errors in Cryptographic Computations, Journal of Cryptology **14** (2001), 101–119.

UNIVERSITÄT PASSAU

Characteristics

- First appearances in 1998* and 2001†.
- By injecting faults into the electronical circuit, the attacker tries to extract secret informations from the latter (e.g. a key).
- Lead to powerful new attack techniques and chip manufacturers were forced to rethink their designs.

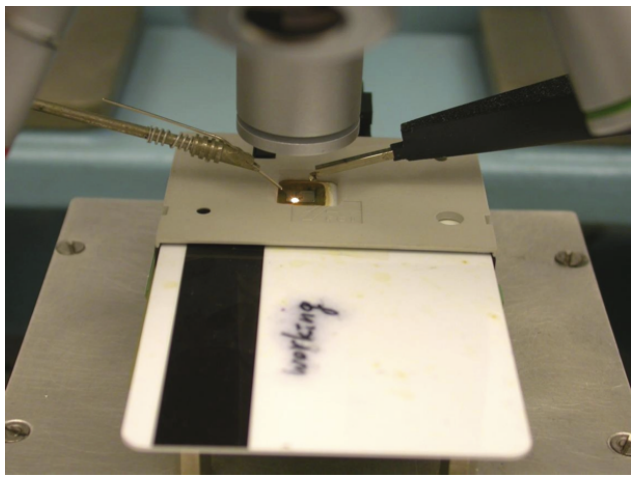


* E. Biham and A. Shamir, Differential Fault Analysis of Secret Key Cryptosystems, In: Burton S. Kaliski Jr. (ed.) *CRYPTO* 1997, LNCS, vol. **1294**, Springer Heidelberg 1997, pp. 513–525.

† D. Boneh, R.A. Demillo, R.J. Lipton, On the Importance of Eliminating Errors in Cryptographic Computations, Journal of Cryptology **14** (2001), 101–119.

Techniques to Induce Faults

- Manipulation of the power-supply voltage to cause miscalculations.
- Manipulation of the circuit's clock.
- Parasitic charge-carrier generation by a laser beam.



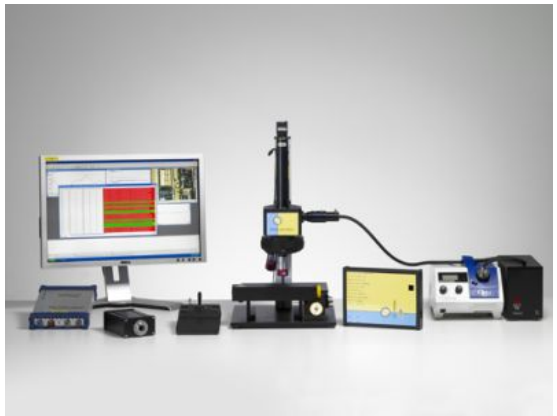


Figure: www.riscure.com

UNIVERSITÄT PASSAU

Some Attacks using Fault Injections

- C. Aumueller et al. Fault Attacks on RSA With CRT: Concrete Results and Practical Countermeasures, *CHES* 2002.
- M. Mohamed, S. Bulygin and J. Buchmann, Improved Differential Fault Analysis of Trivium, *COSADE* 2011.
- M.S. Pedro, M. Soos and S. Guilley, FIRE: Fault Injection for Reverse Engineering, In: C.A. Ardagana and J. Zhou (eds.) *Security and Privacy of Mobile Devices in Wireless Communication* 2011.

Overview

UNIVERSITÄT PASSAU

1. Motivation

2. The PRINCE Block Cipher

3. A Multi-Stage Fault Attack on PRINCE

Definition

Given a block size of n_1 bits and a key size of n_2 bits a block cipher is specified by an encryption function

$$E : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^{n_1}, (m,k) \mapsto c$$

and a decryption function

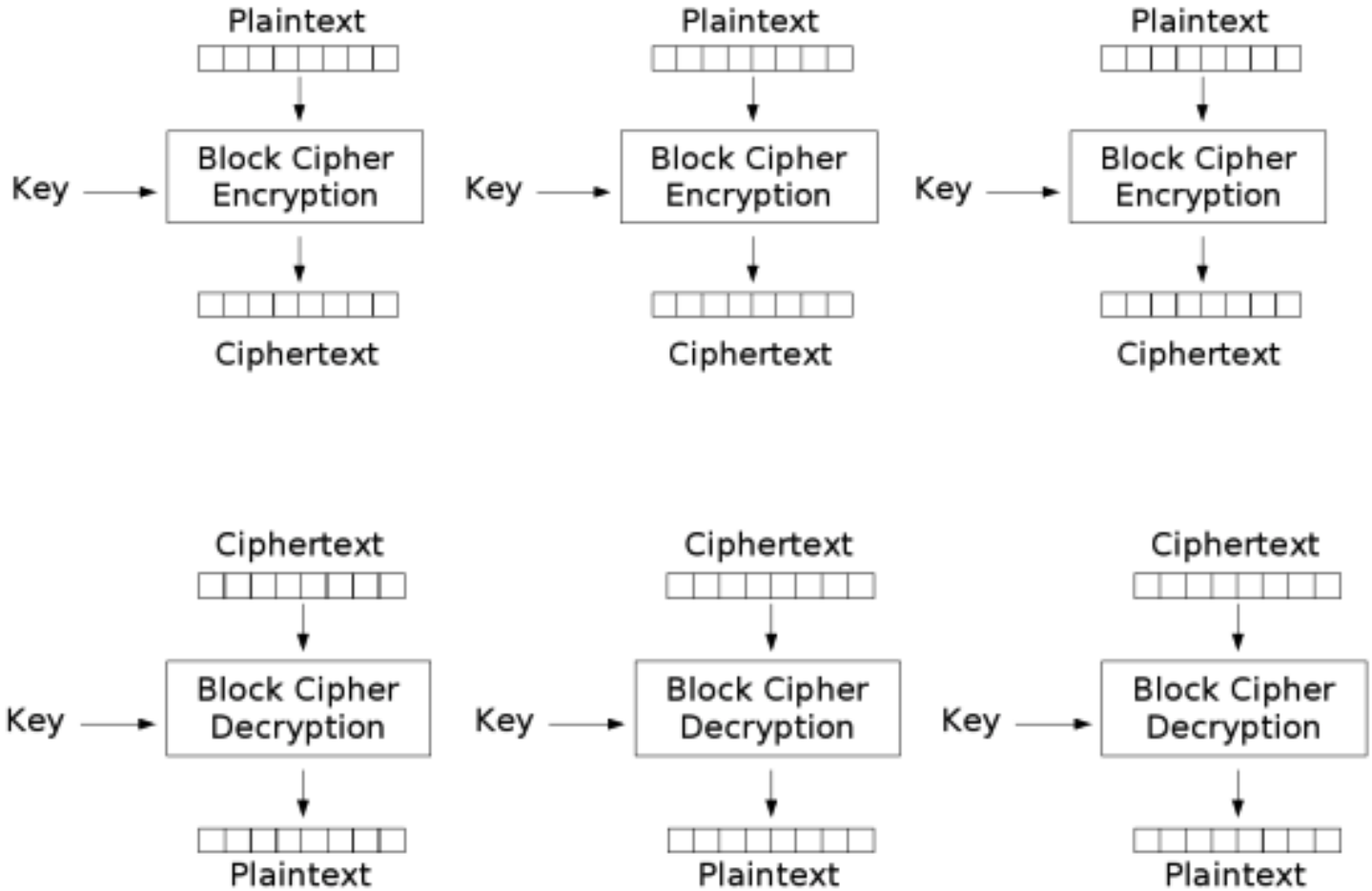
$$D : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^{n_1}, (c,k) \mapsto m$$

such that

$$D_k(E_k(m)) = m$$

for all plaintext messages $m \in \{0,1\}^{n_1}$ and all keys $k \in \{0,1\}^{n_2}$.

Overview on Block Ciphers



General Features

- Uses a 64-bit state and a 128-bit key.
- Based on the so-called *FX* construction.
- Core of the cipher is based on a Substitution-Permutation Network (SPN) and has 10 encryption rounds divided by a middle layer.
- Furthermore the core of PRINCE features the so-called α-reflection property. Due to this it holds that:

$$D_{(k_0 \| k_0' \| k_1)}(\cdot) = E_{(k_0' \| k_0 \| k_1 \oplus \alpha)}(\cdot)$$

  where $\alpha = \texttt{c0ac29b7c97c50dd}$.
- This keeps the hardware costs low and produces only small overheads. (Applications: smart cards, sensor networks, "internet-of-things" etc.)

* J. Borghoff et al., PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications, In: K. Sako and X. Wang (eds.) *ASIACRYPT* 2012, LNCS, vol. **7658**, Springer Heidelberg 2012, pp. 208–225.

The 128-bit key k is split into two parts k_0 and k_1 of 64 bit each,

$$k = k_0 \parallel k_1$$

and extended to 192 bits by the following mapping:

$$(k_0 \parallel k_1) \mapsto (k_0 \parallel k_0' \parallel k_1) := (k_0 \parallel (k_0 >>> 1) \oplus (k_0 >> 63) \parallel k_1)$$

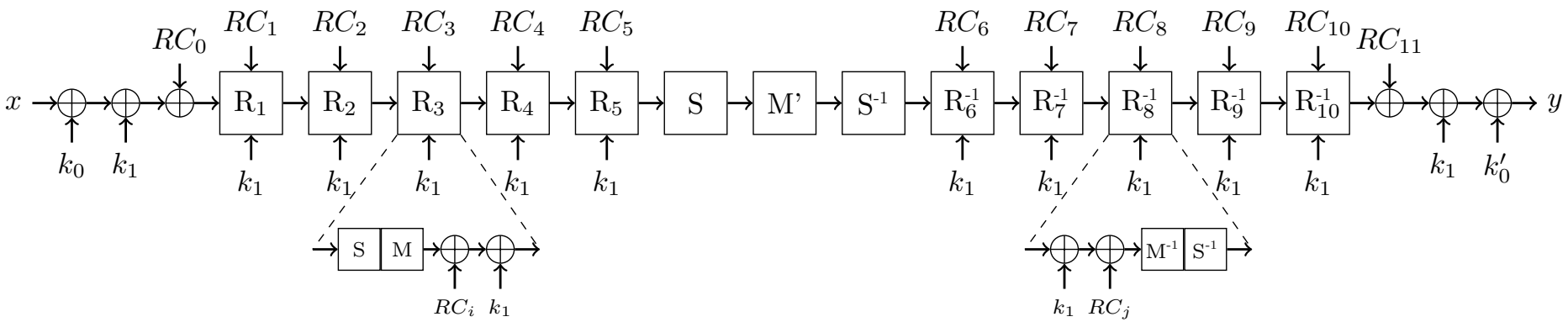
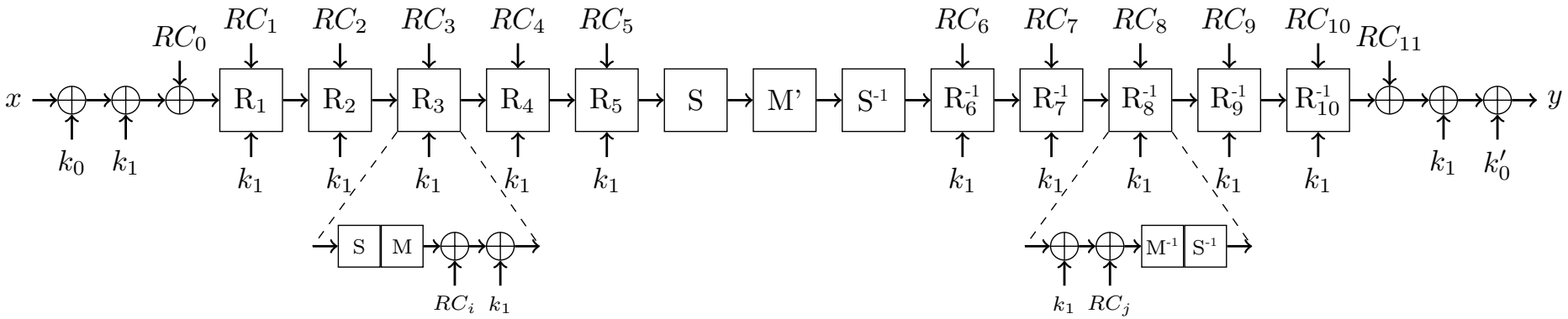


Figure: Layout of PRINCE.

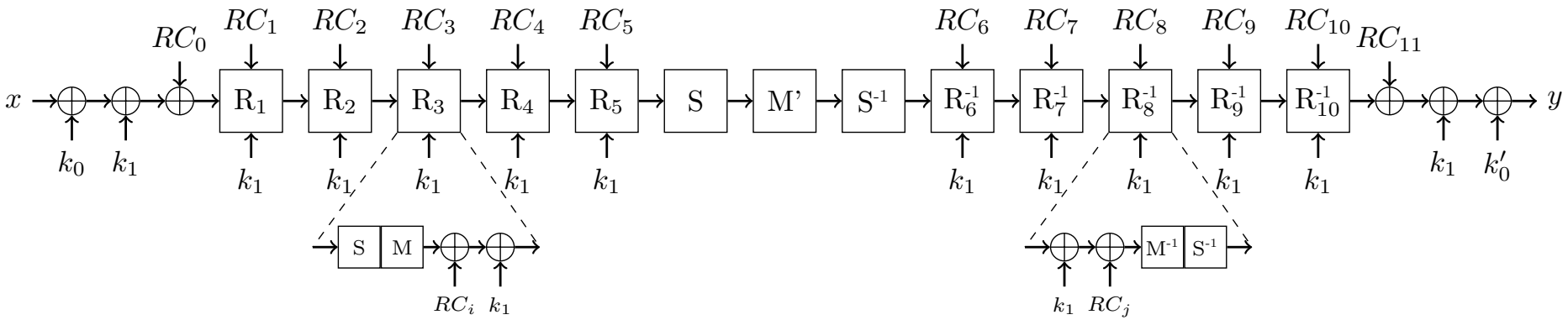
Components of PRINCE



- k_i-add: The 64-bit subkey k_i is XORed to the state.
- S-Layer: All state nibbles are substituted using the 4-bit SBox below.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

Components of PRINCE



- M / M'-Layer: The 64-bit state is multiplied with a 64 $\times$ 64 matrix M resp. M', where $M = SR \circ M'$ and SR shifts row i of the state matrix cyclically to the left by $i - 1$ nibbles.
- RC_i-add: A 64-bit round constant is XORed to the state.

i	RC_i
0 – 2	0000000000000000, 13198a2e03707344, a4093822299f31d0
3 – 5	082efa98ec4e6c89, 452821e638d01377, be5466cf34e90c6c
6 – 8	7ef84f78fd955cb1, 85840851f1ac43aa, c882d32f25323c54
9 – 11	64a51195e0e3610d, d3b5a399ca0c2399, c0ac29b7c97c50dd

Overview

1. Motivation

2. The PRINCE Block Cipher

3. A Multi-Stage Fault Attack on PRINCE

What's a Multi-Stage Fault Attack?

UNIVERSITÄT PASSAU

Outline

- Obtain one (or more) pair(s) of correct and faulty ciphertexts c and c'.
- Use *Differential Fault Analysis* to examine the ciphertext pairs (c, c') and obtain informations about the secret key k.
- Repeat the above scheme if the attacked block cipher uses multiple independent subkeys, i.e. if $k = k_0 \parallel \cdots \parallel k_n$ and the k_i are not "connected" through a key schedule.

Recall: PRINCE uses two independent subkeys.

What's a Multi-Stage Fault Attack?

UNIVERSITÄT PASSAU

Outline

- Obtain one (or more) pair(s) of correct and faulty ciphertexts c and c'.
- Use *Differential Fault Analysis* to examine the ciphertext pairs (c, c') and obtain informations about the secret key k.
- Repeat the above scheme if the attacked block cipher uses multiple independent subkeys, i.e. if $k = k_0 \parallel \cdots \parallel k_n$ and the k_i are not "connected" through a key schedule.

Recall: PRINCE uses two independent subkeys.

What's a Multi-Stage Fault Attack?

UNIVERSITÄT PASSAU

Outline

- Obtain one (or more) pair(s) of correct and faulty ciphertexts c and c'.
- Use *Differential Fault Analysis* to examine the ciphertext pairs (c, c') and obtain informations about the secret key k.
- Repeat the above scheme if the attacked block cipher uses multiple independent subkeys, i.e. if $k = k_0 \parallel \cdots \parallel k_n$ and the k_i are not "connected" through a key schedule.

Recall: PRINCE uses two independent subkeys.

What's a Multi-Stage Fault Attack?

UNIVERSITÄT PASSAU

Outline

- Obtain one (or more) pair(s) of correct and faulty ciphertexts c and c'.
- Use *Differential Fault Analysis* to examine the ciphertext pairs (c, c') and obtain informations about the secret key k.
- Repeat the above scheme if the attacked block cipher uses multiple independent subkeys, i.e. if $k = k_0 \parallel \cdots \parallel k_n$ and the k_i are not "connected" through a key schedule.

Recall: PRINCE uses two independent subkeys.

More Questions

- How good does a fault injection need to be controllable by an attacker such that informations about the secret key can be derived?
- In other words: Can we use ciphertexts obtained from arbitrary faults or are there any requirements in order to produce “useful” faulty ciphertexts?
- What exactly is *Differential Fault Analysis* and how does it work in the case of PRINCE?

UNIVERSITÄT PASSAU

More Questions

- How good does a fault injection need to be controllable by an attacker such that informations about the secret key can be derived?
- In other words: Can we use ciphertexts obtained from arbitrary faults or are there any requirements in order to produce “useful” faulty ciphertexts?
- What exactly is *Differential Fault Analysis* and how does it work in the case of PRINCE?

UNIVERSITÄT PASSAU

More Questions

- How good does a fault injection need to be controllable by an attacker such that informations about the secret key can be derived?
- In other words: Can we use ciphertexts obtained from arbitrary faults or are there any requirements in order to produce “useful” faulty ciphertexts?
- What exactly is *Differential Fault Analysis* and how does it work in the case of PRINCE?

Requirements

UNIVERSITÄT PASSAU

Capabilities of an Attacker

- Known Plaintext Attack: We assume that the attacker is able to generate an arbitrary number of plaintext, (faulty) ciphertext triples (m, c, c').
- Kerckhoffs Principle or "The enemy knows the system": The design of the cipher is known to the adversary. (No security by obscurity)

Fault Models

- Temporal Resolution: Fault injection timing is controllable very precisely, i.e. injection after a specific operation of the cipher.
- Spatial Resolution: Injection effects a single nibble (4-bit value) of the whole state. The affected nibble itself is either **known** (model: RKF) or **unknown** (model: RUF).
- Effects: Injection changes the state nibble to a random and unknown 4-bit value.

Requirements

UNIVERSITÄT PASSAU

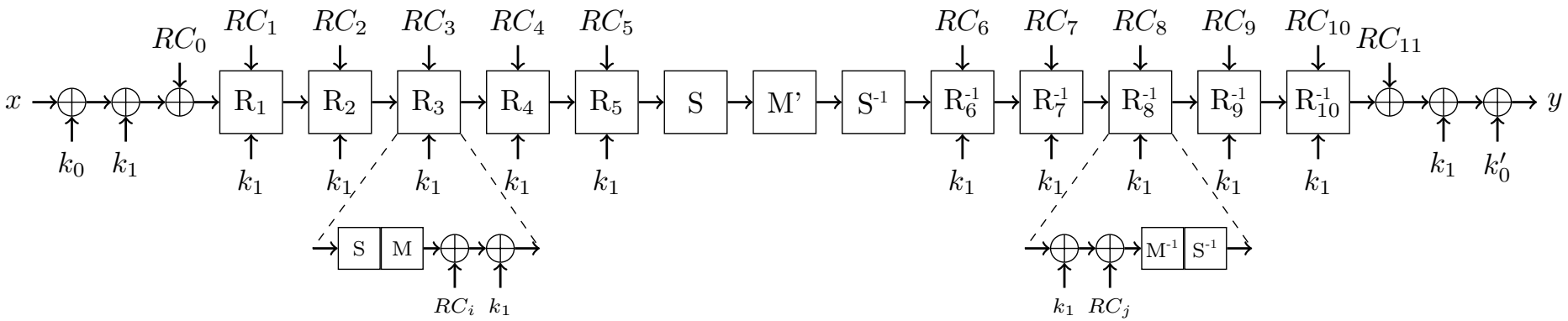
Capabilities of an Attacker

- Known Plaintext Attack: We assume that the attacker is able to generate an arbitrary number of plaintext, (faulty) ciphertext triples (m, c, c').
- Kerckhoffs Principle or "The enemy knows the system": The design of the cipher is known to the adversary. (No security by obscurity)

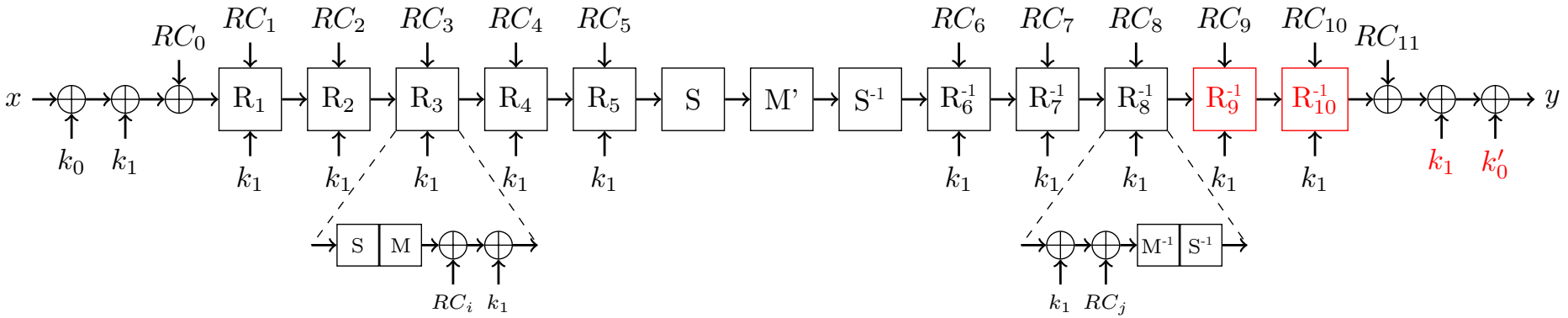
Fault Models

- Temporal Resolution: Fault injection timing is controllable very precisely, i.e. injection after a specific operation of the cipher.
- Spatial Resolution: Injection effects a single nibble (4-bit value) of the whole state. The affected nibble itself is either **known** (model: RKF) or **unknown** (model: RUF).
- Effects: Injection changes the state nibble to a random and unknown 4-bit value.

Where to Inject Faults?

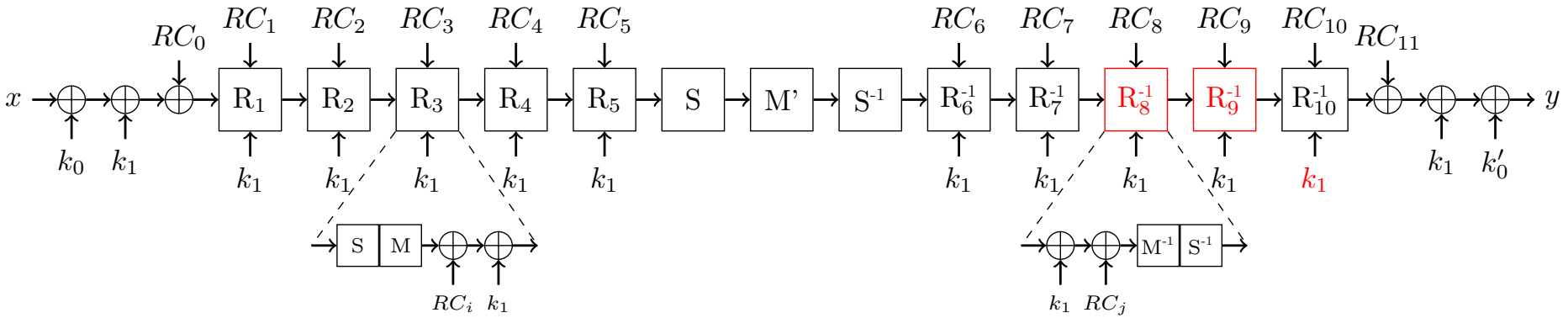


Attack Stage 0



- Inject fault in R_9^{-1}.
- Analyse fault propagation and ciphertext pairs.
- Obtain informations on $k_0' \oplus k_1$.

Attack Stage 1



- Inject fault in R_8^{-1}.
- Analyse fault propagation and ciphertext pairs.
- Obtain informations on k_1.

Definition

Let $a = b_0 \parallel b_1 \parallel b_2 \parallel b_3$ be a 4-bit value and let $j \in \{0, \ldots, 3\}$. Then we define the map:

$$\varphi : \mathbb{B}^4 \times \{0, \ldots, 3\} \longrightarrow \mathbb{B}^4, (a, j) \longmapsto \varphi_j(a)$$

where $\varphi_j(a)$ is equal to a but with the j-th bit b_j set to 0.

Bit Pattern

j	$\varphi_j(a)$			
0	0	b_1	b_2	b_3
1	b_0	0	b_2	b_3
2	b_0	b_1	0	b_3
3	b_0	b_1	b_2	0

UNIVERSITÄT PASSAU

Definition

Let $a = b_0 \parallel b_1 \parallel b_2 \parallel b_3$ be a 4-bit value and let $j \in \{0, \ldots, 3\}$. Then we define the map:

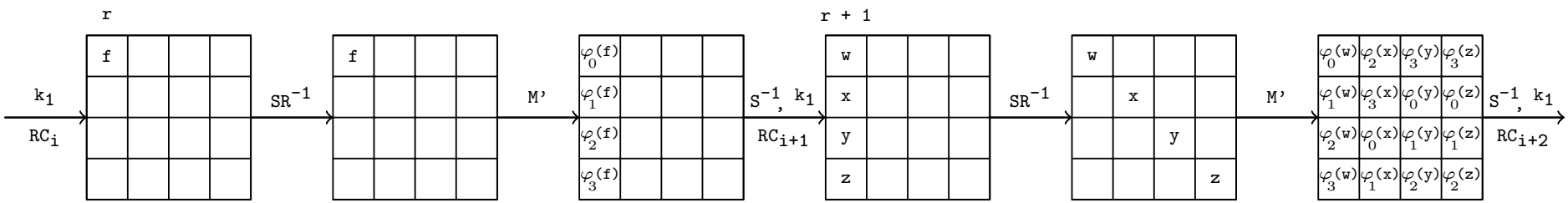
$$\varphi : \mathbb{B}^4 \times \{0, \ldots, 3\} \longrightarrow \mathbb{B}^4, (a, j) \longmapsto \varphi_j(a)$$

where $\varphi_j(a)$ is equal to a but with the j-th bit b_j set to 0.

Bit Pattern

j	$\varphi_j(a)$			
0	0	b_1	b_2	b_3
1	b_0	0	b_2	b_3
2	b_0	b_1	0	b_3
3	b_0	b_1	b_2	0

Fault Propagation over 2 Rounds



Fault Equations

Let v_i, v_i', k_i and q_i be variables. We substitute the nibbles of correct and faulty ciphertexts (intermediate states) for v_i and v_i', key nibbles for k_i and round constant nibbles for q_i.

$$E_i : SBox(v_i \oplus k_i \oplus q_i) \oplus SBox(v_i' \oplus k_i \oplus q_i) = \begin{cases} \varphi_{j_i}(w), & i \in \{0, \ldots, 3\} \\ \varphi_{j_i}(x), & i \in \{4, \ldots, 7\} \\ \varphi_{j_i}(y), & i \in \{8, \ldots, 11\} \\ \varphi_{j_i}(z), & i \in \{12, \ldots, 15\} \end{cases}$$

$$(j_i)_{i=0,\ldots,15} = \begin{cases} (0,1,2,3,2,3,0,1,3,0,1,2,3,0,1,2), & l \in \{0,7,10,13\} \\ (3,0,1,2,1,2,3,0,2,3,0,1,2,3,0,1), & l \in \{1,4,11,14\} \\ (2,3,0,1,0,1,2,3,1,2,3,0,1,2,3,0), & l \in \{2,5,8,15\} \\ (1,2,3,0,3,0,1,2,0,1,2,3,0,1,2,3), & l \in \{3,6,8,12\} \end{cases}$$

Definition

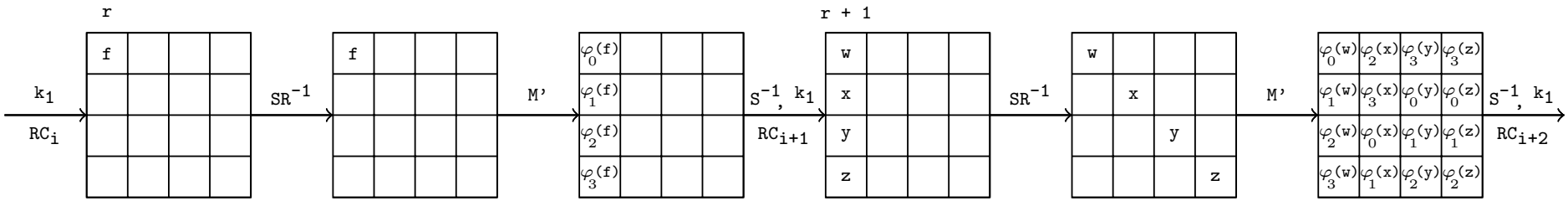
For every fault eqaution E_i we introduce a *key nibble candidate set* S_i with

$$S_i = \{(t, u) \mid t, u \in \mathbb{B}^4\}$$

for $i \in \{0, \ldots, 15\}$. Furthermore let $S = (S_i)_{i=0,\ldots,15}$.

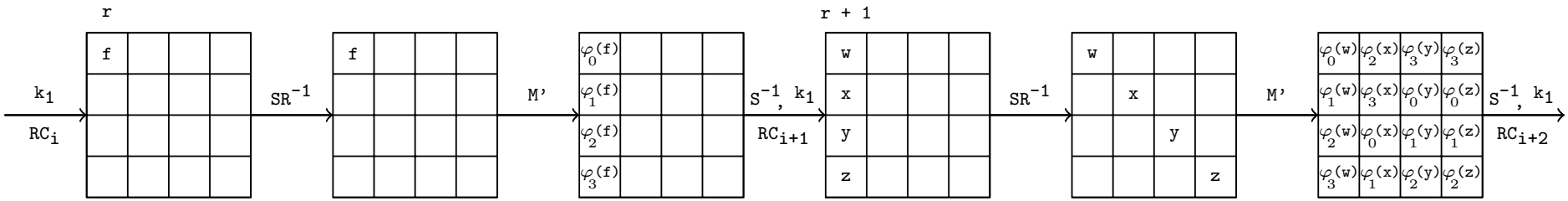
DFA Algorithm:

Input: (c, c') (intermediate state (v, v') for the 2nd stage)
Output: Set S containing candidates for $k_0' \oplus k_1$ (or k_1 for the 2nd stage)
return `outer_filtering(inner_filtering(evaluation(`c, c'`)))`



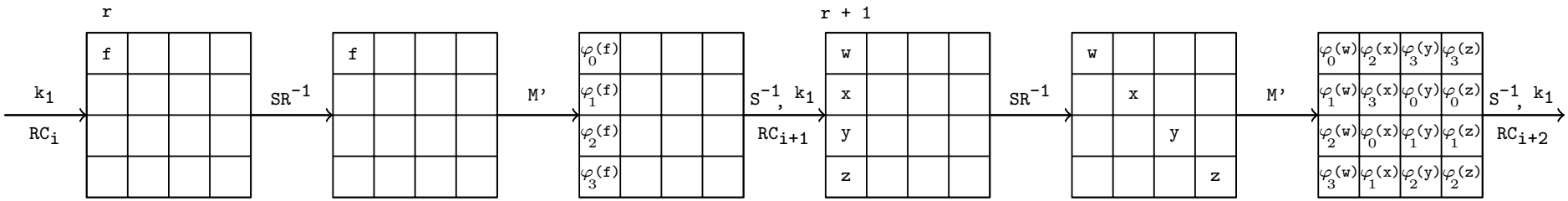
Overview on the Single Steps

- `evaluation`: Compute $E_i(u) = t$ for all $u \in \mathbb{B}^4$ and save the result (u, t) to the set S_i.
- `inner_filtering`: Discard all tuples (u, t) from S_i where t doesn't match the pattern φ_{j_i} associated with E_i.
- `outer_filtering`: Exploit the fact that the elements of the sets $S_{4 \cdot m}, \ldots, S_{4 \cdot m+3}$ are derived from a commen preimage to discard even more invalid tuples (u, t).



Overview on the Single Steps

- `evaluation`: Compute $E_i(u) = t$ for all $u \in \mathbb{B}^4$ and save the result (u, t) to the set S_i.
- `inner_filtering`: Discard all tuples (u, t) from S_i where t doesn't match the pattern φ_{j_i} associated with E_i.
- `outer_filtering`: Exploit the fact that the elements of the sets $S_{4 \cdot m}, \ldots, S_{4 \cdot m+3}$ are derived from a commen preimage to discard even more invalid tuples (u, t).



Overview on the Single Steps

- `evaluation`: Compute $E_i(u) = t$ for all $u \in \mathbb{B}^4$ and save the result (u, t) to the set S_i.
- `inner_filtering`: Discard all tuples (u, t) from S_i where t doesn't match the pattern φ_{j_i} associated with E_i.
- `outer_filtering`: Exploit the fact that the elements of the sets $S_{4 \cdot m}, \ldots, S_{4 \cdot m+3}$ are derived from a commen preimage to discard even more invalid tuples (u, t).

UNIVERSITÄT PASSAU

Example

Assume we have the following setup:

$$k = \texttt{01234567 89ABCDEF 01234567 89ABCDEF}$$
$$m = \texttt{01234567 89ABCDEF}$$
$$c = \texttt{0A72342A 02193229}$$
$$c' = \texttt{21A19DCD 25D7433C}$$

The faulty ciphertext c' was obtained by injecting the error value $e = \texttt{0xC}$ into nibble s_0 of the state at the beginning of round R_9^{-1}.

A Fault Attack on PRINCE

As a reminder we list again the possible index pattern below.

j	$\varphi_j(a)$			
0	0	b_1	b_2	b_3
1	b_0	0	b_2	b_3
2	b_0	b_1	0	b_3
3	b_0	b_1	b_2	0

Table: Distribution of key nibbles after `evaluation` (1st column) ...

S_i	$\sum$	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\#S_0$	16	2	0	4	0	0	0	2	2	0	0	0	0	4	2	0
$\#S_1$	16	2	0	0	4	0	0	2	0	0	0	2	2	0	2	2
$\#S_2$	16	2	0	0	0	0	0	2	0	4	2	0	0	2	2	2
$\#S_3$	16	0	0	0	0	2	2	0	2	2	2	2	2	0	0	2

Apply the same technique to the other sets $S_4, \ldots, S_{15}$. As a result there remain only $2^{20} = 1.048.576$ from the initial 2^{64} candidates for $k_0' \oplus k_1$.

A Fault Attack on PRINCE

As a reminder we list again the possible index pattern below.

j	$\varphi_j(a)$			
0	0	b_1	b_2	b_3
1	b_0	0	b_2	b_3
2	b_0	b_1	0	b_3
3	b_0	b_1	b_2	0

Table: ... after `inner_filtering` ...

S_i	$\sum$	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\#S_0$	8	2	0	4	0	0	0	2	0	0	0	0	0	0	0	0
$\#S_1$	4	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0
$\#S_2$	8	2	0	0	0	0	0	0	0	4	0	0	0	2	0	0
$\#S_3$	8	0	0	0	0	0	2	0	2	0	2	0	2	0	0	0

Apply the same technique to the other sets $S_4, \ldots, S_{15}$. As a result there remain only $2^{20} = 1.048.576$ from the initial 2^{64} candidates for $k_0' \oplus k_1$.

A Fault Attack on PRINCE

As a reminder we list again the possible index pattern below.

j	$\varphi_j(a)$			
0	0	b_1	b_2	b_3
1	b_0	0	b_2	b_3
2	b_0	b_1	0	b_3
3	b_0	b_1	b_2	0

Table: ... and after `outer_filtering`.

S_i	$\sum$	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\#S_0$	4	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
$\#S_1$	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
$\#S_2$	4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
$\#S_3$	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0

Apply the same technique to the other sets $S_4, \ldots, S_{15}$. As a result there remain only $2^{20} = 1.048.576$ from the initial 2^{64} candidates for $k_0' \oplus k_1$.

A Fault Attack on PRINCE

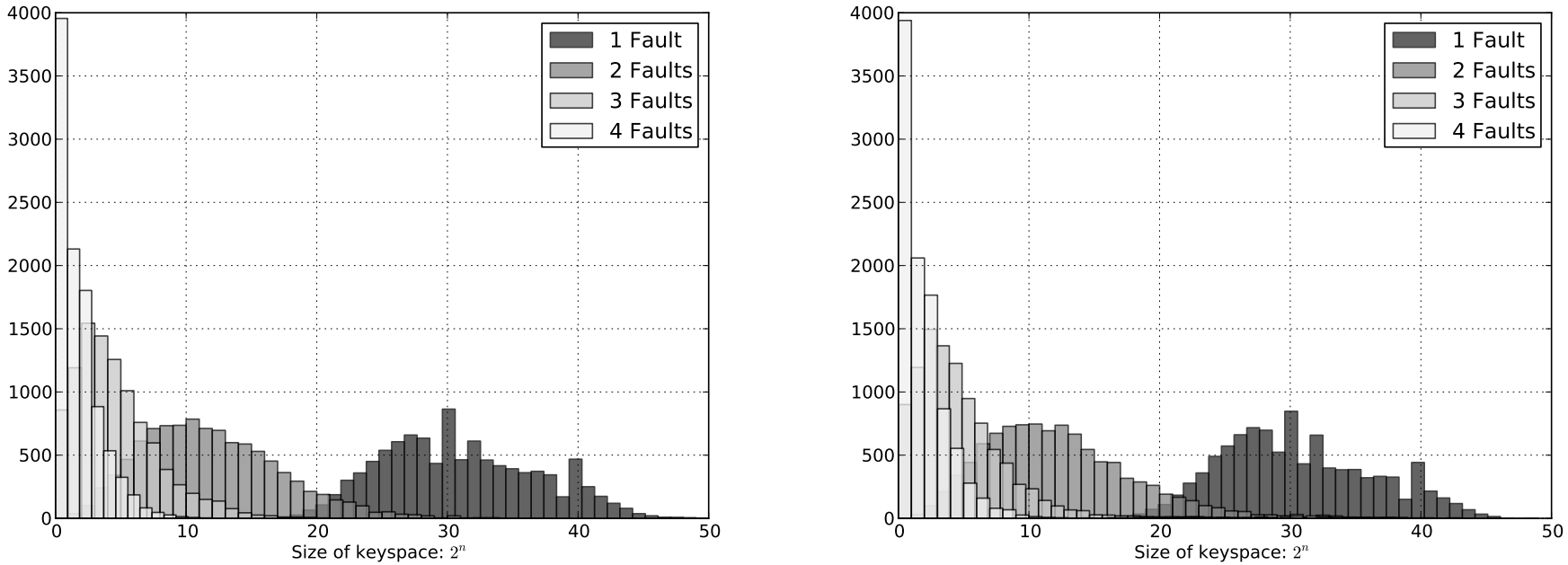


Figure: Experimental results for stage 0 (left) and stage 1 (right). The data was obtained through 10.000 runs of the attack using fault model RUF.

A Fault Attack on PRINCE

Table: Statistics for $k_0' \oplus k_1$ and k_1 candidates after stage 0 and 1.

	stage 0				stage 1			
# keys / # faults	1	2	3	4	1	2	3	4
min	$2^{17.00}$	1	1	1	$2^{16.00}$	1	1	1
max	$2^{50.00}$	$2^{38.00}$	$2^{24.00}$	$2^{12.00}$	$2^{49.00}$	$2^{44.00}$	$2^{40.00}$	$2^{43.00}$
avg	$2^{30.89}$	$2^{11.44}$	$2^{4.12}$	$2^{1.47}$	$2^{30.41}$	$2^{11.64}$	$2^{4.44}$	$2^{1.82}$
median	$2^{34.50}$	$2^{19.50}$	$2^{12.50}$	$2^{7.00}$	$2^{33.50}$	$2^{21.50}$	$2^{21.00}$	$2^{21.00}$

Summary: In order to reconstruct the complete 128-bit key $k_0 \parallel k_1$ it is sufficient to inject approximately 3 – 4 faults.

Outlook

Q: Can we apply Multi-Stage Fault Attacks to other ciphers?

A: Yes, indeed we can!

- We constructed an algorithm that can be used to analyse (SPN) block ciphers having independent subkeys using Multi-Stage Fault Attacks.
- Showed applications to PRINCE (this talk) and LED-128.
- To appear soon. (hopefully :-)

Outlook

Q: Can we apply Multi-Stage Fault Attacks to other ciphers?

A: Yes, indeed we can!

- We constructed an algorithm that can be used to analyse (SPN) block ciphers having independent subkeys using Multi-Stage Fault Attacks.
- Showed applications to PRINCE (this talk) and LED-128.
- To appear soon. (hopefully :-)

Outlook

Q: Can we apply Multi-Stage Fault Attacks to other ciphers?

A: Yes, indeed we can!

- We constructed an algorithm that can be used to analyse (SPN) block ciphers having independent subkeys using Multi-Stage Fault Attacks.
- Showed applications to PRINCE (this talk) and LED-128.
- To appear soon. (hopefully :-)

Outlook

Q: Can we apply Multi-Stage Fault Attacks to other ciphers?

A: Yes, indeed we can!

- We constructed an algorithm that can be used to analyse (SPN) block ciphers having independent subkeys using Multi-Stage Fault Attacks.
- Showed applications to PRINCE (this talk) and LED-128.
- To appear soon. (hopefully :-)

Outlook

Q: Can we apply Multi-Stage Fault Attacks to other ciphers?

A: Yes, indeed we can!

- We constructed an algorithm that can be used to analyse (SPN) block ciphers having independent subkeys using Multi-Stage Fault Attacks.
- Showed applications to PRINCE (this talk) and LED-128.
- To appear soon. (hopefully :-)

Fin

Thank you for your attention!