PARAMETRIC TROJANS FOR FAULT-BASED ATTACKS ON CRYPTOGRAPHIC HARDWARE

Raghavan Kumar, University of Massachusetts Amherst

Contributions by:

Philipp Jovanovic, University of Passau

Wayne P. Burleson, University of Massachusetts Amherst

Ilia Polian, University of Passau

UMass Amherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Motivation

- **Hardware Trojans**: malicious modifications of circuits by an untrusted (overseas) foundry.
- Here: Trojan insertion techniques by manufacturing process manipulation (“**MAPLE Trojans**”).
- Based on manipulation of V_{in}-V_{out} characteristics.
- Very low likelihood of detection by any means.
- Demonstration of a fault-based attack to a recent cryptosystem made possible by MAPLE Trojans.

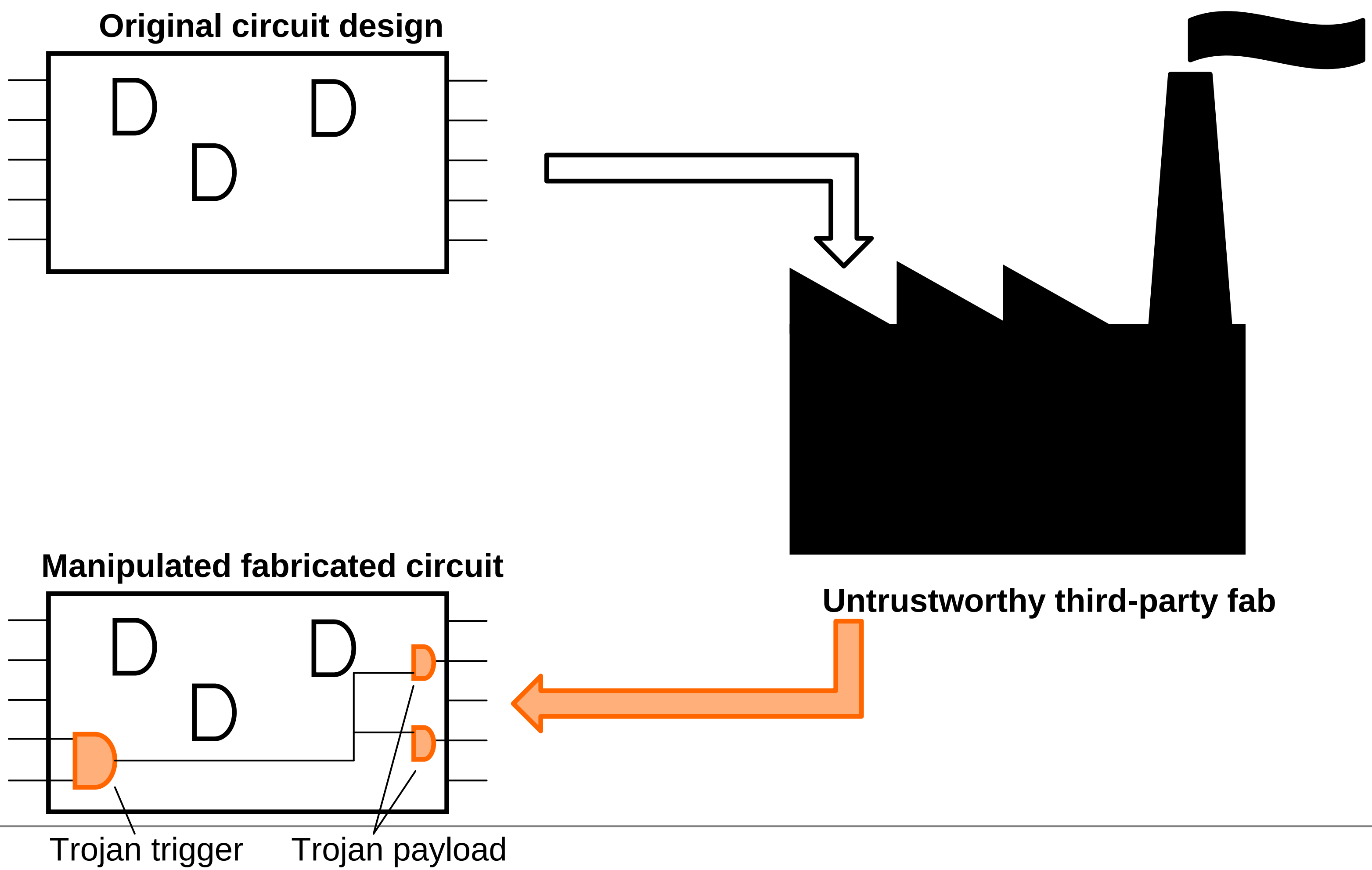
Outline: Questions

- What are Hardware Trojans?
- How do MAPLE Trojans work?
- What are fault-based attacks on ciphers?
- How do MAPLE Trojans facilitate such attacks?
- What countermeasures are effective?

UMassAmherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Hardware Trojans



Hardware Trojans

- Triggering mechanism:
  - Internal (time-based, physical condition)
  - External (by user or by another component)
- Payload:
  - Change functionality
  - Leak information
  - Denial of service
- Detection:
  - Functional testing (like for manufacturing defects)
  - Parametric / side-channel analysis
  - Optical inspection

UMass Amherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Underlying Attack Model

- Most Hardware Trojans, including MAPLE Trojans presented here, require two co-operating attackers.
- Attacker 1: Malicious fab (or individual employees) who plants the Trojan trigger/payload into the circuit.
- Attacker 2: User of the manufactured circuit who knows the triggering condition.
- Attacker 1 and 2 are in general not identical.
- Users of the circuit who are not attackers are interested in detecting the presence of a Trojan.

UMassAmherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Are Hardware Trojans Real?

- Not known with certainty!
- No fully documented, published case.
- Strong indirect indications found.
- Large interest in academia, government / military, industry; significant research funding.
- Many assumptions in literature don’t seem realistic.

- What is for sure: they are an interesting scientific problem with strong relationship to test.

UMassAmherst

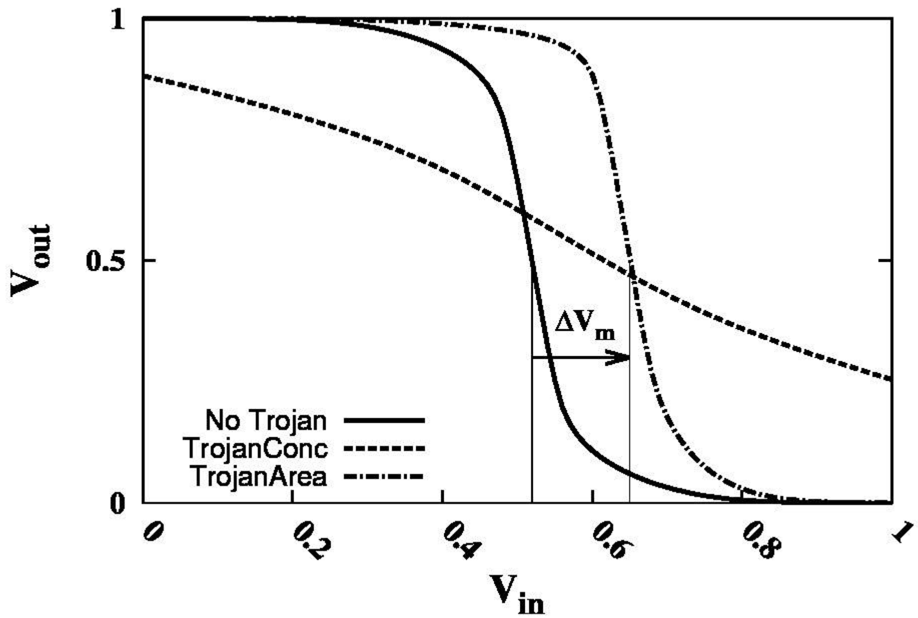
UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Outline: Questions

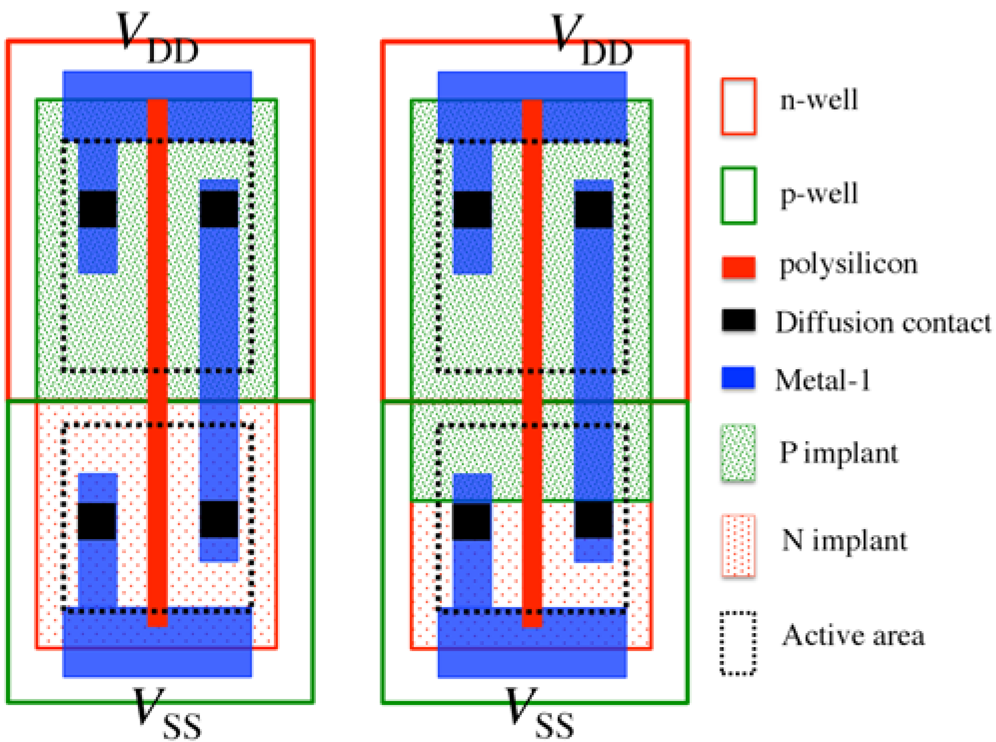
- What are Hardware Trojans?
- **How do MAPLE Trojans work?**
- What are fault-based attacks on ciphers?
- How do MAPLE Trojans facilitate such attacks?
- What countermeasures are effective?

MAPLE Trojans

- Manipulate the V_{in}-V_{out} characteristic of a logic gate (here: inverter).
- **TrojanArea:** reduce the dopant area within a transistor's active area.
- **TrojanConc:** significantly reduce doping concentration.
- Both techniques can be applied to individual gate instances.



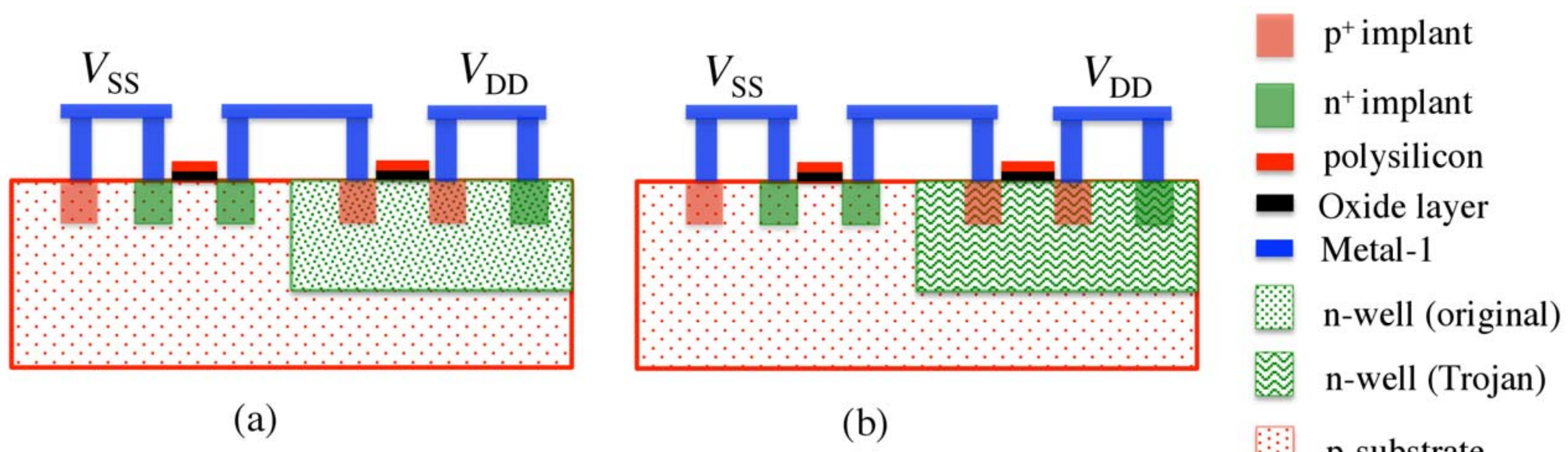
TrojanArea (view from above)



(a) (b)

- Simple modification of mask layout

TrojanConc (cross-sectional view)



- Requires an extra mask and 2 extra process steps

Outline: Questions

- What are Hardware Trojans?
- How do MAPLE Trojans work?
- **What are fault-based attacks on ciphers?**
- How do MAPLE Trojans facilitate such attacks?
- What countermeasures are effective?

Fault-based Attacks

- Cryptographic systems (ciphers) restrict access to **secret information** to authorized persons.
- Traditional **cryptanalysis** obtains secret information without authorization by utilizing mathematical weaknesses of the cipher (“breaking the code”).
- Fault-based attacks target the **hardware implementation** of the cipher.
- Perform encoding / decoding with a **fault injected** into the circuit by a physical disturbance.
- Derive secret information by differential cryptanalysis.

UMass Amherst

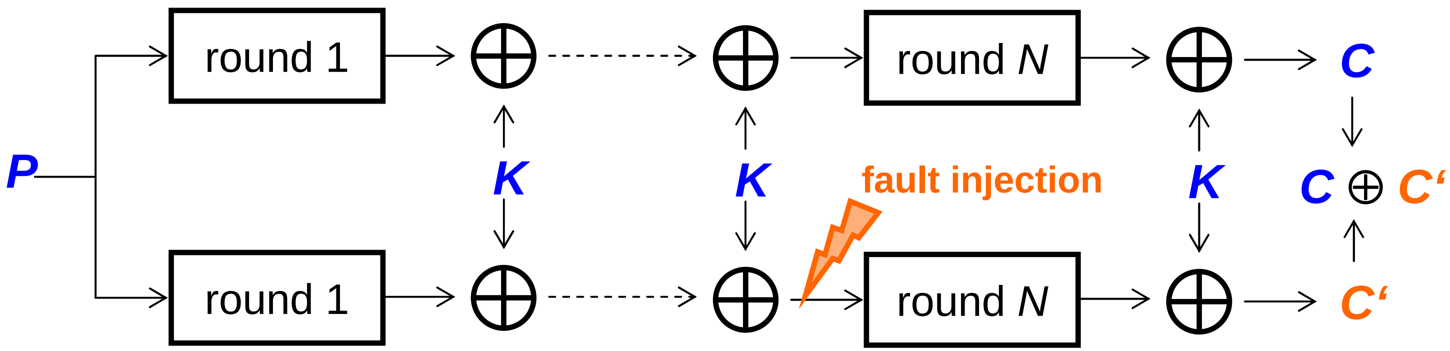
UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Fault-based Attacks: Fault Injection

- A variety of techniques:
  - Vary the supply voltage (generate a spike).
  - Vary the clock frequency (generate a glitch).
  - Overheat the device.
  - Expose to intense light (laser).
- State-of-the-art attacks require very accurate fault injection (time and location).
- **Use Trojan-infected gates for precise fault inj.**

Source: www.riscure.com

Fault-based Attacks: Post-processing

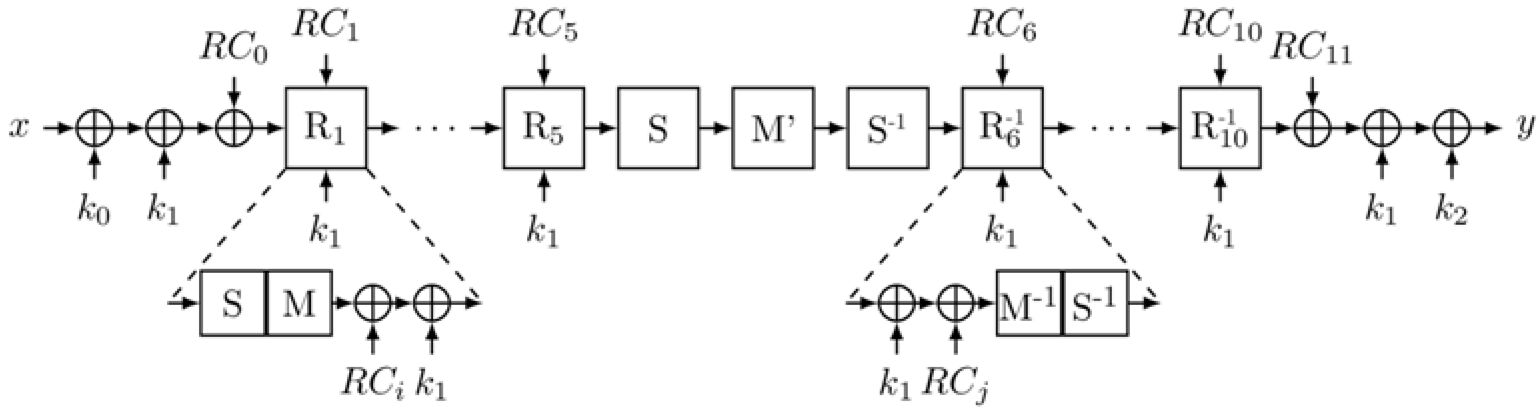


- A cipher E encrypts plaintext P into ciphertext C using secret key K. Solving $C = E(P, K)$ breaks the cipher but is (should be) mathematically infeasible.
- Repeated encryption with fault injection f yields a fault-affected ciphertext C' with $C' = E_f(P, K)$.
- This information can assist in solving $C = E(P, K)$.

UMassAmherst

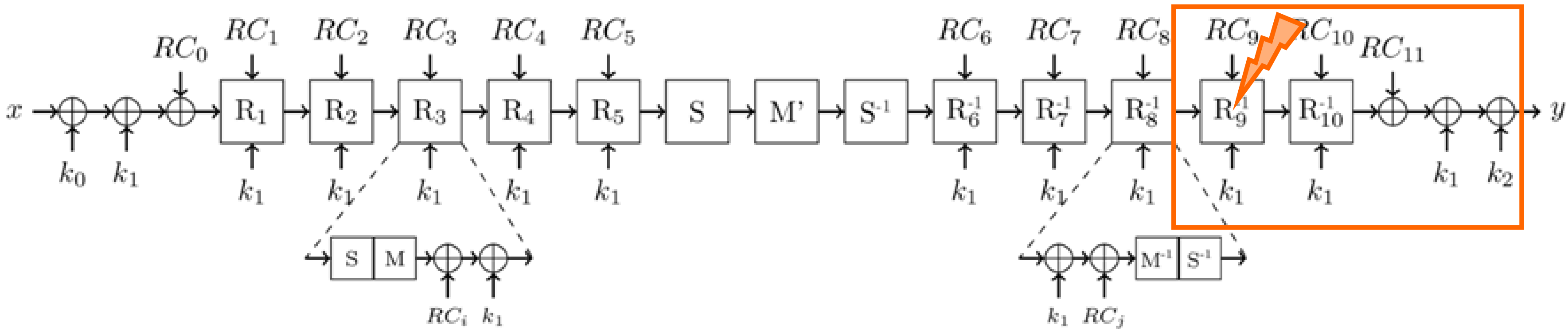
UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Case Study: Lightweight Block Cipher PRINCE



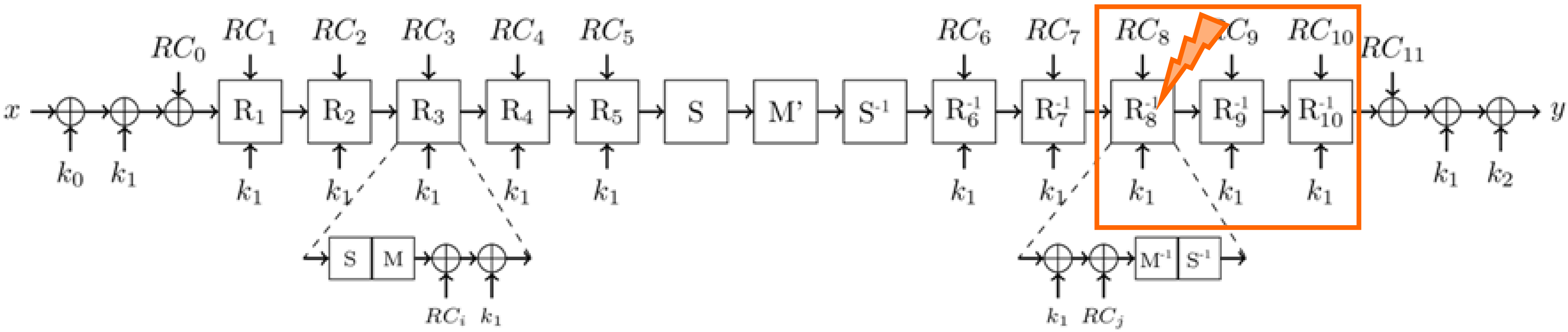
- 2×64 bit key $k = k_0 \,||\, k_1$
  - Key expansion into 192 bits: $k_2 := (k_0 \ggg 1) \oplus (k_0 \gg 63)$.
- 10 rounds with 4 operations
  - Nonlinear SBox S; multiplication with matrix M; addition of round constant RC_j; subkey addition k_i.

Fault-based Cryptanalysis of PRINCE



- **Stage 0:** inject fault in round 9, derive a "small" set of candidates (~ 2^{13}) for expression **($k_1 \oplus k_2$)**.

Fault-based Cryptanalysis of PRINCE



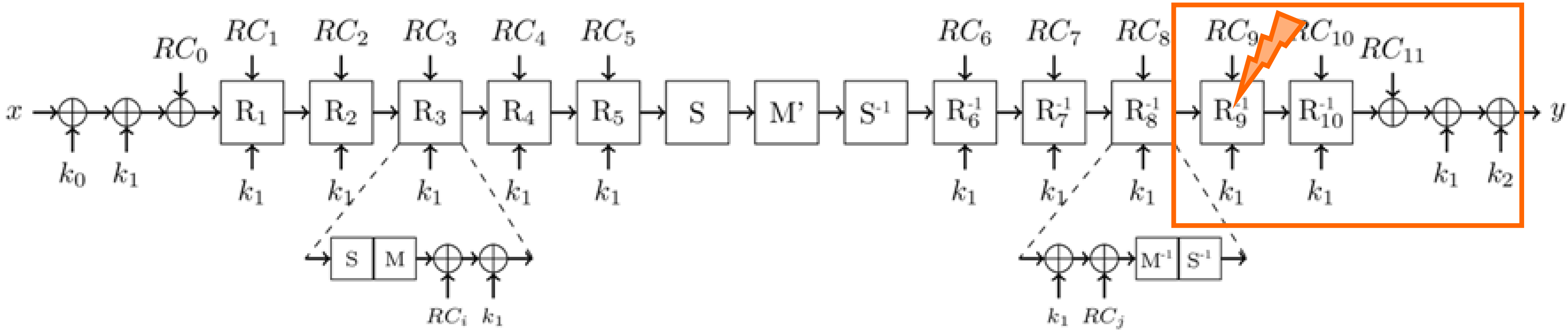
- **Stage 0:** inject fault in round 9, derive a "small" set of candidates (~ 2^{13}) for expression $(k_1 \oplus k_2)$.
- **Stage 1:** for each candidate from stage 1 compute value after round 10; inject fault in round 8; derive a "small" set of candidates (~ 2^{16}) for k_1.

Requirements on Fault Injection

- The state of PRINCE is organized in 4-bit “nibbles”.
- Stage-0 faults must be **restricted to one nibble** in round 9.
  - No faults may be simultaneously present in other nibbles or in other rounds, otherwise post-processing won’t work.
- Stage-1 faults: restricted to one nibble in round 8.

- We call faults according to this requirement **exploitable** for stage 0 / stage 1.

Cryptanalysis Details (Stage 0)



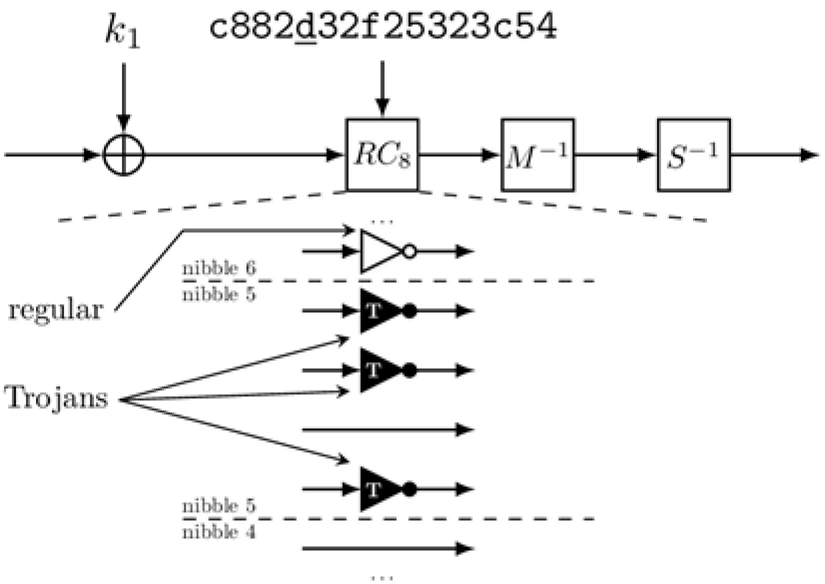
- Forward-propagate fault effect (Boolean difference) from round 9 to SBox in round 10.
- Backward-propagate the fault-free and the faulty ciphertext observed at the outputs to same location.
- Construct equations, use them for excluding key candidates (filtering).

Outline: Questions

- What are Hardware Trojans?
- How do MAPLE Trojans work?
- What are fault-based attacks on ciphers?
- **How do MAPLE Trojans facilitate such attacks?**
- What countermeasures are effective?

Fault Injection by MAPLE Trojan

- Manipulate some gates to make them “weaker”.
  - Under nominal Vdd, the circuit will work normally.
  - Under slightly reduced (~ 10%) Vdd, the manipulated gates will fail first (with certain probability).



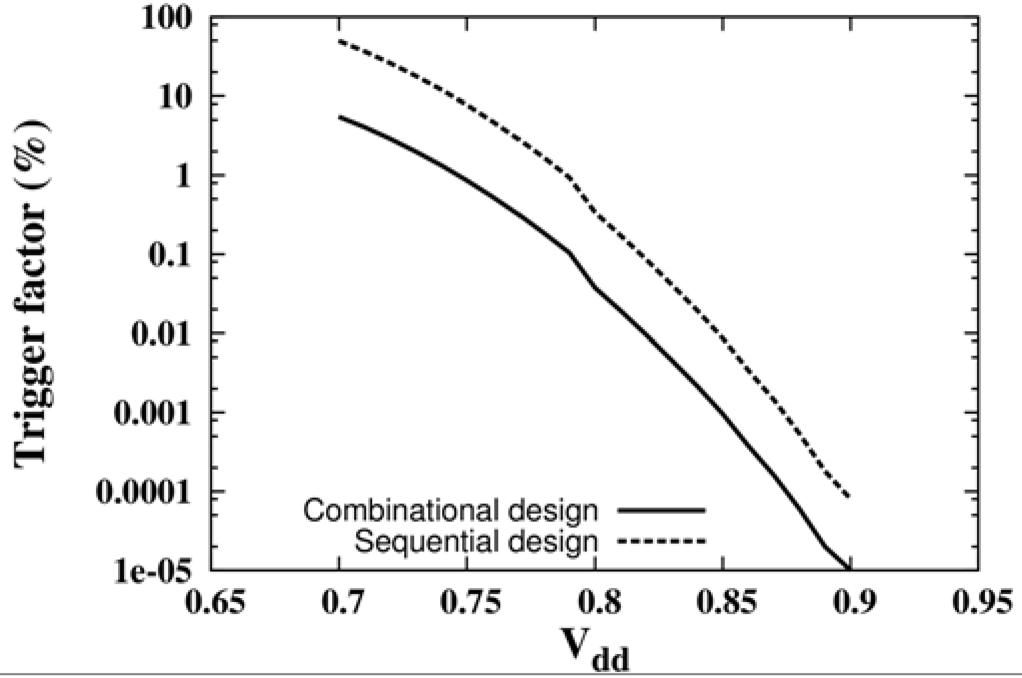
- Select gates such as to inject **exploitable faults.**
  - Example: 3 inverters belonging to the same state nibble in round constant addition.

UMass Amherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

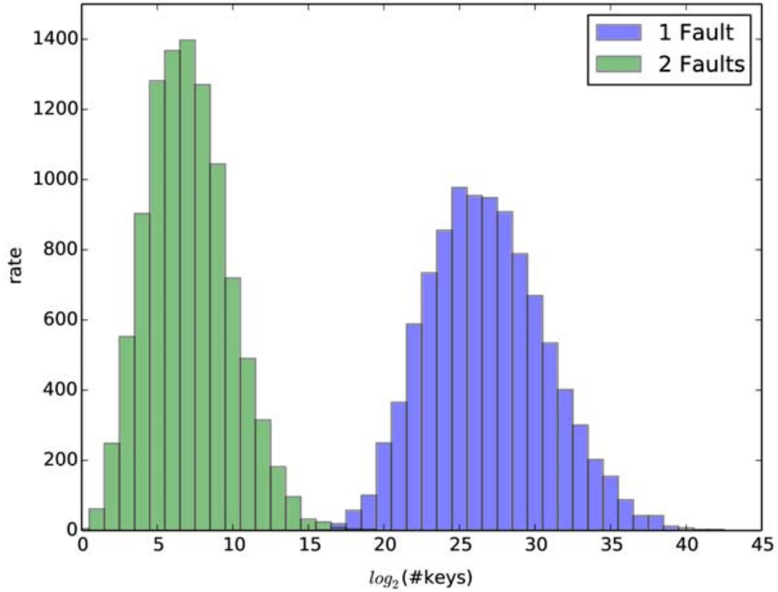
Probability of Exploitable Faults

- Faults in one nibble in either round 8 or 9.
- TrojanConc (similar results for TrojanArea).
- ~10^{-5} for 10% Vdd reduction.

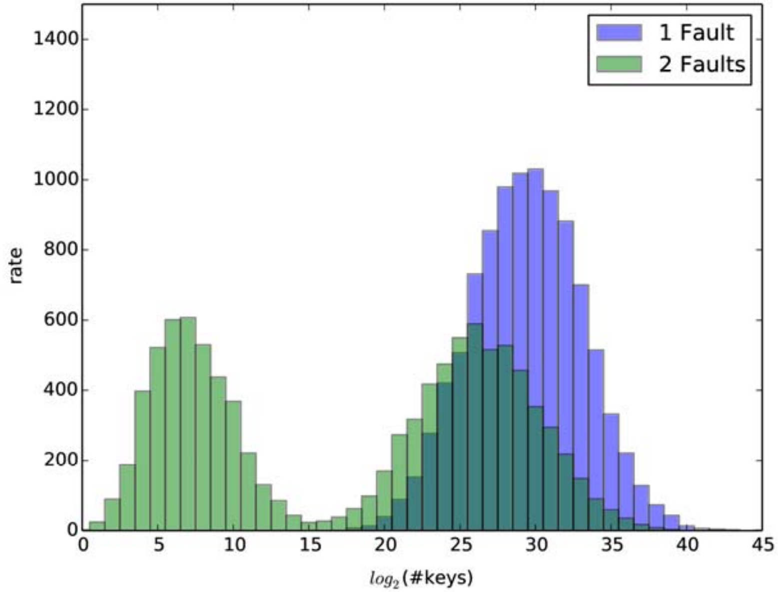


Results

- 10,000 executions of the attack with random plaintext.
- 4–5 fault injections sufficient for key reconstruction.



stage 0



stage 1

Outline: Questions

- What are Hardware Trojans?
- How do MAPLE Trojans work?
- What are fault-based attacks on ciphers?
- How do MAPLE Trojans facilitate such attacks?
- **What countermeasures are effective?**

Detection of MAPLE Trojans

- Functional testing
  - No fault effect under nominal Vdd.
  - Too low probability of activation for slightly reduced Vdd.
  - Not distinguishable from random fails under low Vdd.
- Side-channel analysis
  - Only very few gates affected; impact minimal compared with circuit-global variability.
- Visual inspection
  - No layout modification; changes in doping concentration or dopant area are nearly impossible to see.

Other Countermeasures

- On-chip voltage detectors
  - Very moderate Vdd reduction to values that are routinely observed in regular operation due to power-supply noise.
- Limiting the number of encryptions
  - Effective but does not tell whether circuit is manipulated.
- Frequent key exchange
  - If a key is determined, only data protected by that key (before exchange) is compromised.
  - Key distribution may not work if the attacker has physical access to the chip.

UMass Amherst

UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Conclusions

- New, extremely stealthy Trojans.
- Based on manufacturing process manipulation.
- Alter electrical characteristics of selected gates.
- Application to fault-based analysis shows feasibility (4-5 exploitable faults required for key recovery, 10,000 fault injections per exploitable fault).

- Future work: silicon experiments (with ETH Zurich), better understanding of countermeasures.

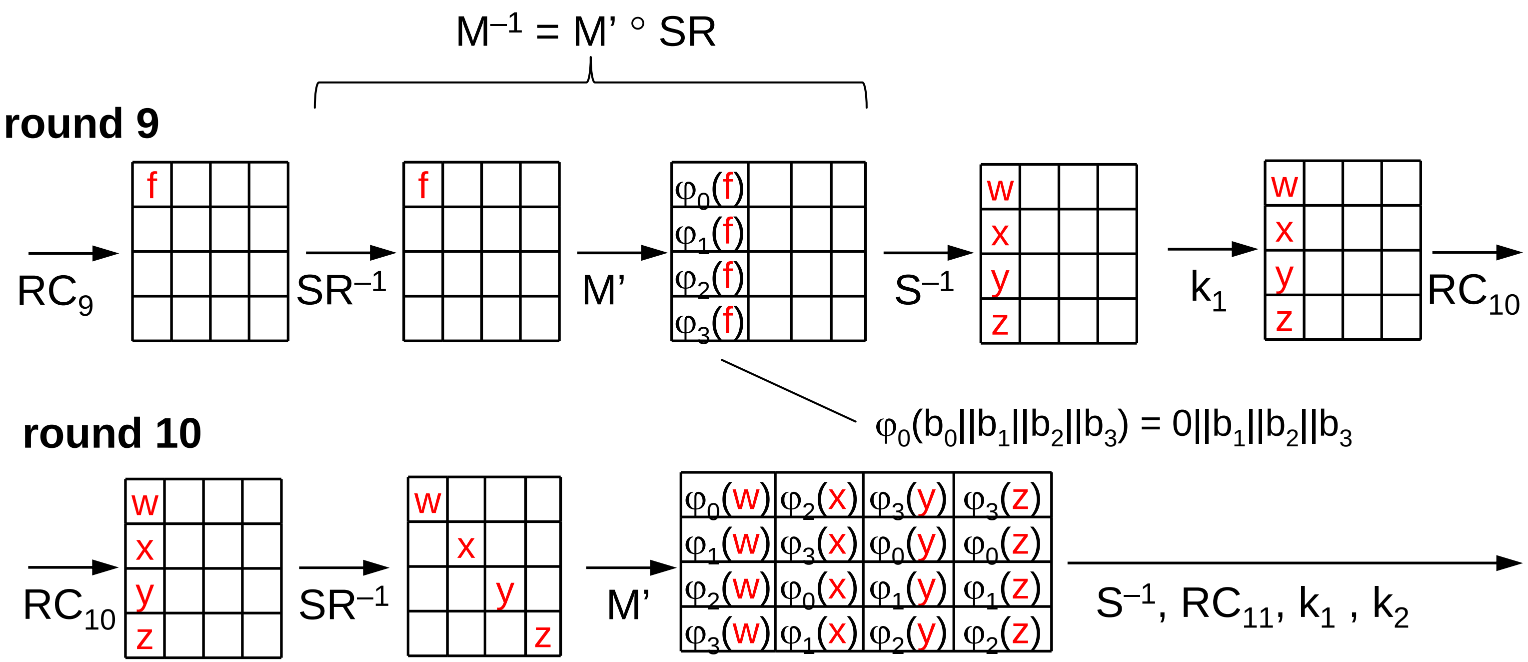
BACKUP SLIDES

UMass Amherst

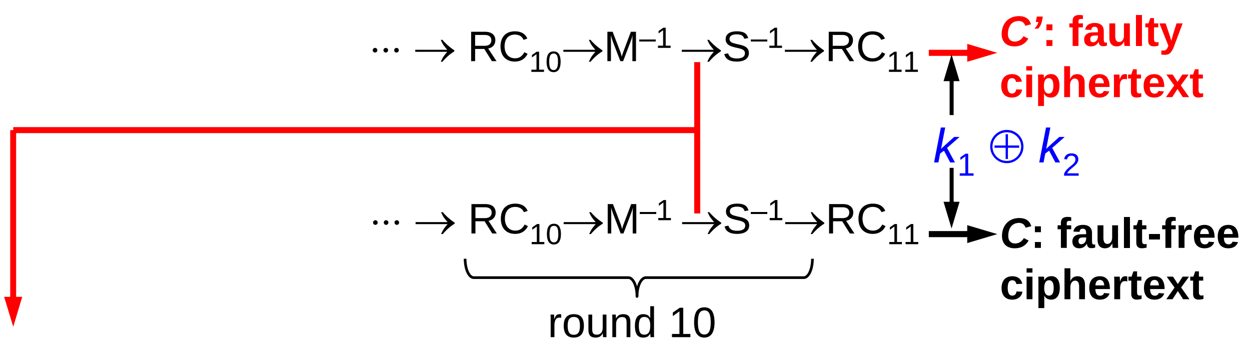
UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Forward-propagation

- Effect propagation of fault f in nibble 0.



Backward-propagation and Filtering



$$\begin{array}{|c|c|c|c|}\hline \varphi_0(w) & \varphi_2(x) & \varphi_3(y) & \varphi_3(z) \\ \hline \varphi_1(w) & \varphi_3(x) & \varphi_0(y) & \varphi_0(z) \\ \hline \varphi_2(w) & \varphi_0(x) & \varphi_1(y) & \varphi_1(z) \\ \hline \varphi_3(w) & \varphi_1(x) & \varphi_2(y) & \varphi_2(z) \\ \hline \end{array} = S(C' \oplus (k_1 \oplus k_2) \oplus RC_{11}) \oplus S(C \oplus (k_1 \oplus k_2) \oplus RC_{11})$$

- System of equations over GF(16) with indeterminates k_1, … k_{16} (secret key), w, x, y, z.
- Exclude key candidates that violate these equations.