

Leakage-Resilient Cryptography

Stefan Dziembowski
University of Rome
La Sapienza



Krzysztof Pietrzak
CWI Amsterdam

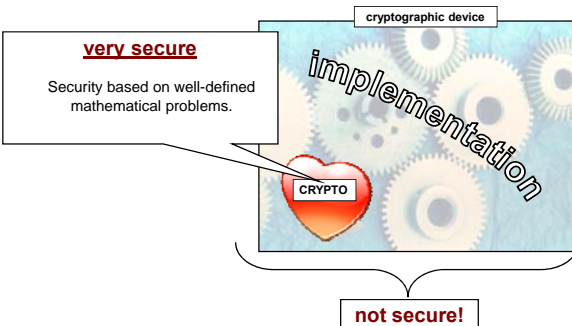


Plan

1. Motivation and introduction
2. Our model
3. Our construction

these slides are available at
www.dziembowski.net/Slides

How to construct secure cryptographic devices?



very secure
Security based on well-defined mathematical problems.

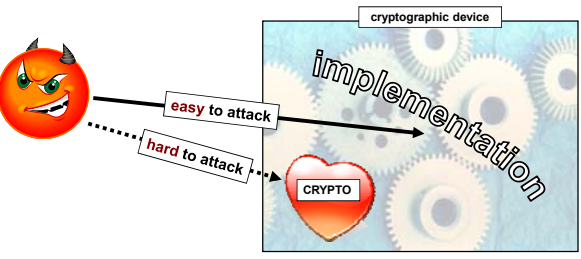
cryptographic device

implementation

CRYPTO

not secure!

The problem



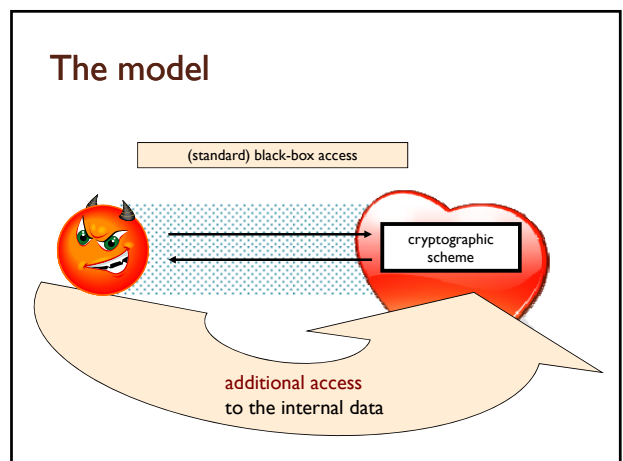
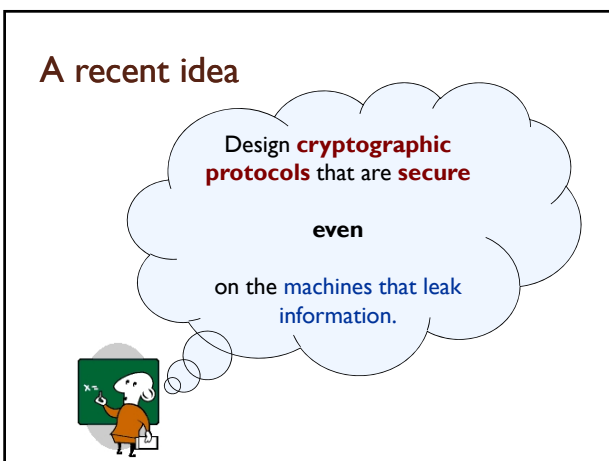
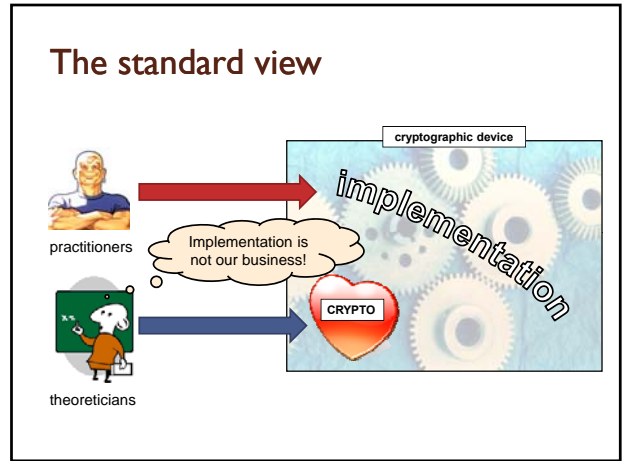
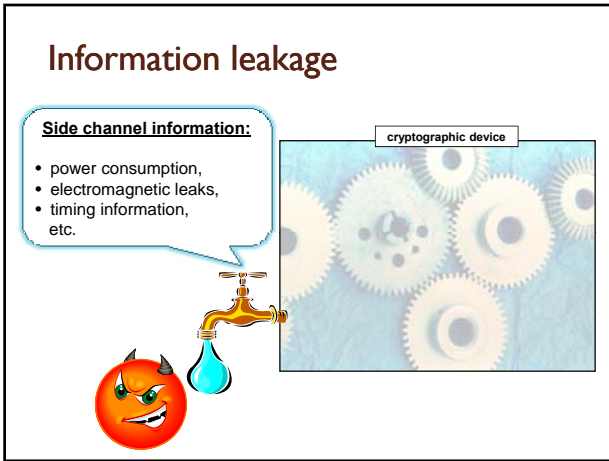
cryptographic device

implementation

CRYPTO

easy to attack

hard to attack



Some prior work



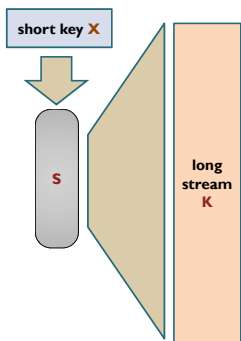
- S. Chari, C. S. Jutla, J.R. Rao, P. Rohatgi. **Towards Sound Approaches to Counteract Power-Analy: Attacks.** CRYPTO 1999
- Y. Ishai, A. Sahai, and D.Wagner. **Private Circuits: Securing Hardware against Probing Attacks.** CRYPTO 2003
- S. Micali and L. Reyzin. **Physically Observable Cryptography (Extended Abstract).** TCC 2004
- R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. **Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering.** TCC 2004.
- C. Petit, F.-X. Standaert, O. Pereira, T.G. Malkin, M. Yung. **A Block Cipher Based PRNG Secure Against Side-Channel Key Recovery.** ASIACCS 2008
- a sequence of papers by F.-X. Standaert, T.G. Malkin, M. Yung, and others, available at the [web-page](#) of F.-X. Standaert.

Our contribution

We construct a
stream cipher
that is secure against a
very large and well-defined class of leakages.

Our construction is in the standard model
(i.e. **without the random oracles**).

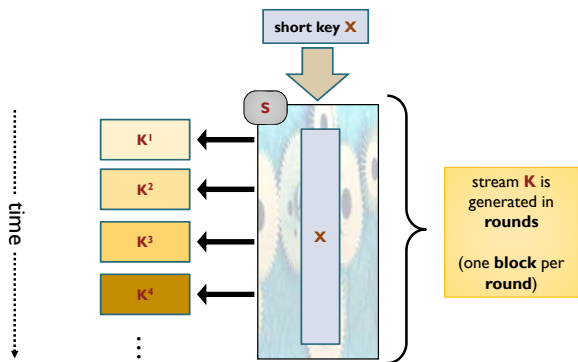
stream ciphers \approx pseudorandom generators

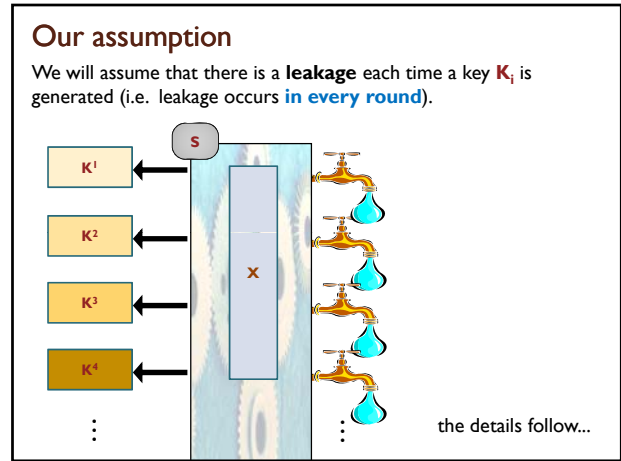
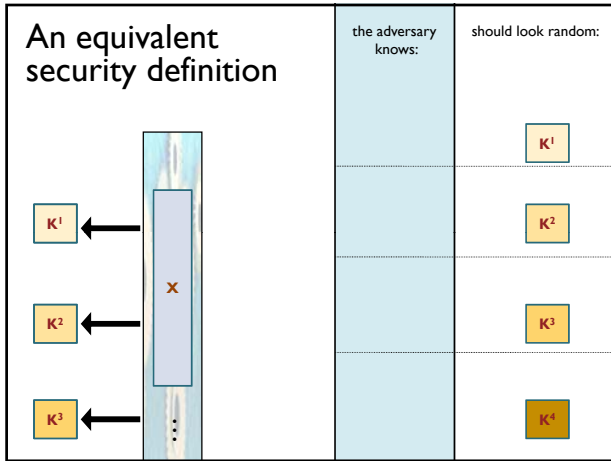


a computationally bounded adversary should not be able to distinguish **K** from random



How do the stream ciphers work in practice?





Leakage-resilient stream cipher - the model

Examples of the “leakage functions” from the literature:

- Y. Ishai, A. Sahai, and D. Wagner. Private Circuits: **Securing Hardware against Probing Attacks.**

The adversary can learn the **value of some wires** of a circuit that computes the cryptographic scheme.
- another example (a “**Hamming attack**”):

The adversary can learn the **sum of the secret bits.**

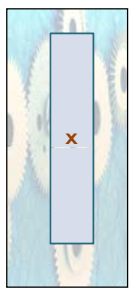
We consider a very general class of leakages



In every i th round the adversary chooses a poly-time computable "bounded-output function"

$$f : \{0,1\}^n \rightarrow \{0,1\}^m \text{ for } m < n$$

and learns $f(X)$



We say that the adversary "retrieved m bits" (in a given round).

How much leakage can we tolerate?

In our construction the total number of retrieved bits will be **larger than** the length of the secret key X

(but in every **round** the **number of retrieved bits** will be much less than $|X|$)

this will be a parameter

How can we achieve it? by **key evolution!**

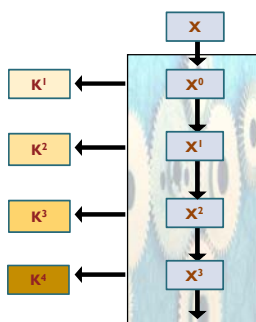
Key evolution

In each round the secret key X gets refreshed.

Assumptions:

key evolution has to be **deterministic**
(no refreshing with external randomness)

also the refreshing procedure may cause leakage



How to define security?

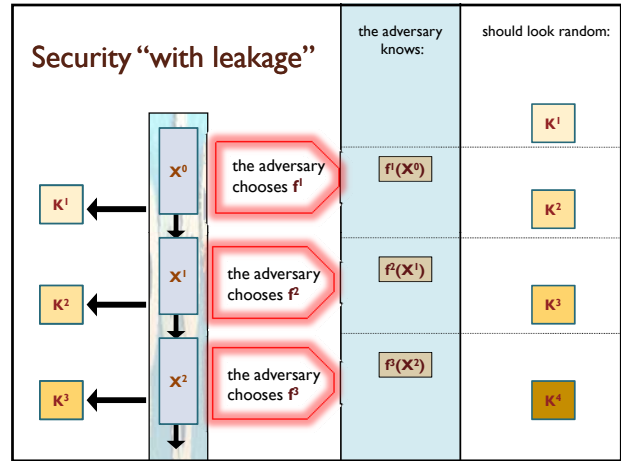
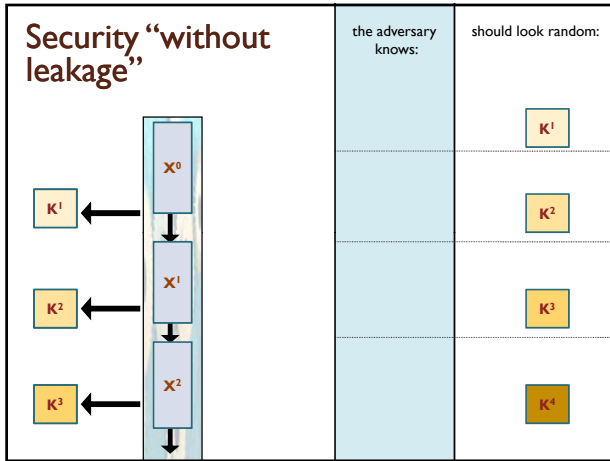
Is "indistinguishability" possible?

Problem

If the adversary can "retrieve" just one bit of K_i then he can distinguish it from random...

Solution

Indistinguishability will concern the "future" keys K_i



Key evolution – a problem

Recall that:

1. the key evolution is deterministic
2. the "leakage function f_i " can be any poly-time function.

Therefore:

the function f_i can always compute the "future" keys

What to do?

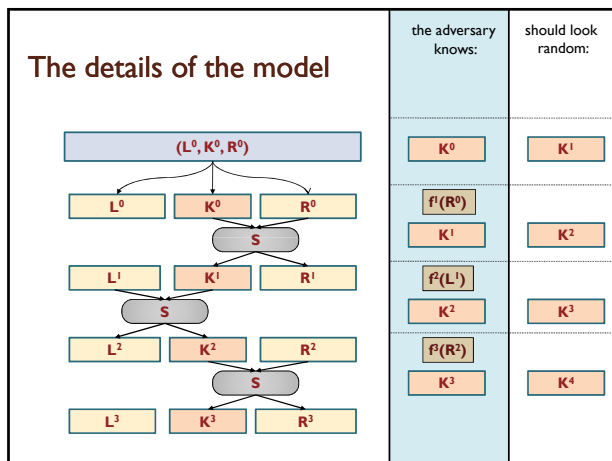
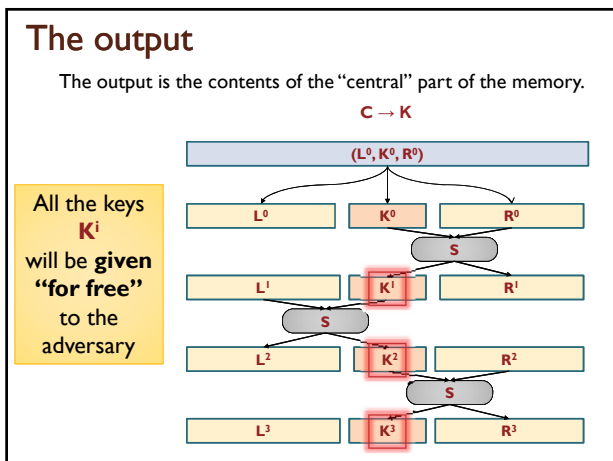
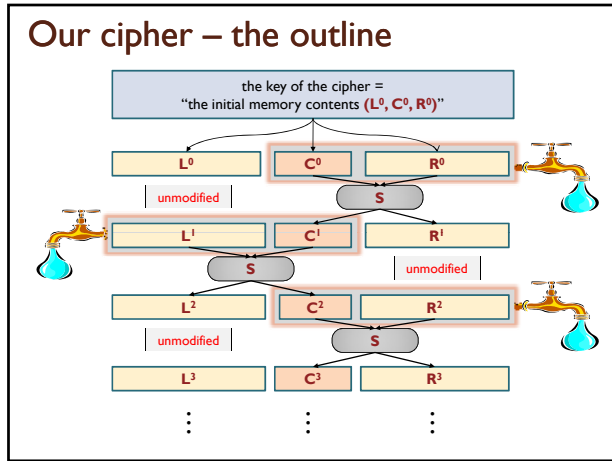
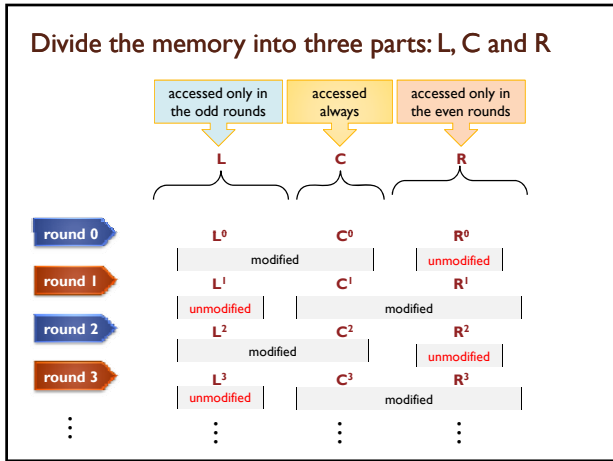
We use the principle introduced in:

S. Micali and L. Reyzin.
Physically Observable Cryptography.
 TCC 2004

"only computation leaks information"

in other words:

"untouched memory cells do not leak information"



Leakage-resilient stream cipher - the construction

How to construct such a cipher?

Idea

Use the **randomness extractors**.

A function

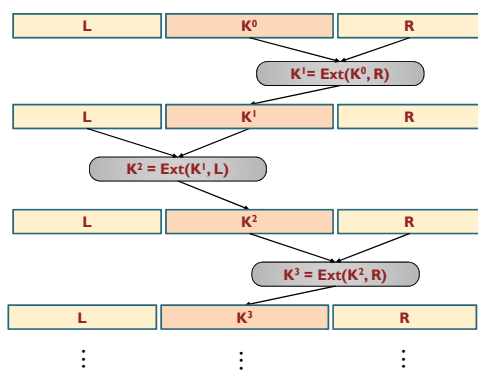
$$\text{Ext} : \{0,1\}^k \times \{0,1\}^r \rightarrow \{0,1\}^m$$

is an (ϵ, n) -**randomness extractor** if for

- a uniformly random \mathbf{K} , and
 - every \mathbf{X} with min-entropy n
- we have that

$$(\text{Ext}(\mathbf{K}, \mathbf{X}), \mathbf{K}) \text{ is } \epsilon \text{ -- close to uniform.}$$

Alternating extraction [DP, FOCS07]



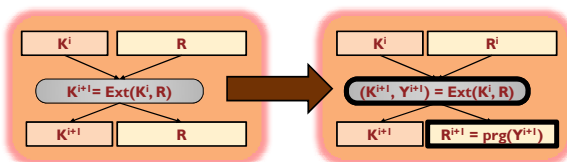
A fact from [DP07]

Even if
a constant fraction of \mathbf{L} and \mathbf{R} leaks
the keys $\mathbf{K}_1, \mathbf{K}_2, \dots$
look “almost uniform”

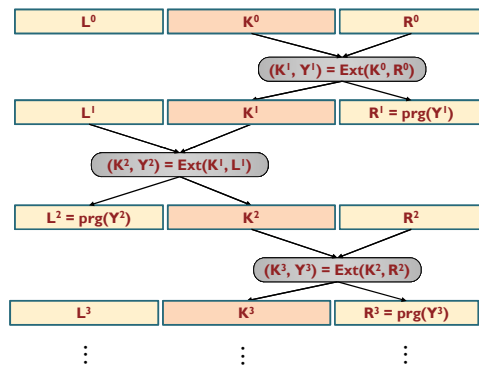
Idea: “add key evolution to [DP07]”

What to do?

Use a pseudorandom generator (**prg**) in the following way:



Our scheme



Our results (1/2)

assume the existence of pseudorandom generators

then

the cipher constructed on the previous slides is secure against the adversary that in every round retrieves:

$$\lambda = \omega(\log(\text{length of the key})) \text{ bits}$$

this covers many real-life attacks (e.g. the “Hamming attack”)

35

Our results (2/2)

assume the existence of pseudorandom generators secure against exponential-size circuits

then

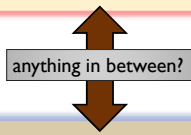
the cipher constructed on the previous slides is secure against the adversary that in every round retrieves:

$$\lambda = \Theta(\text{length of the key}) \text{ bits}$$

36

An open problem

Y. Ishai, A. Sahai, and D. Wagner:
Private Circuits: Securing Hardware against Probing Attacks.
CRYPTO 2003
generic construction, **weaker** model



This paper:
specific construction, **stronger** model

Thank you for your attention!