Welcome

AARHUS UNIVERSITY

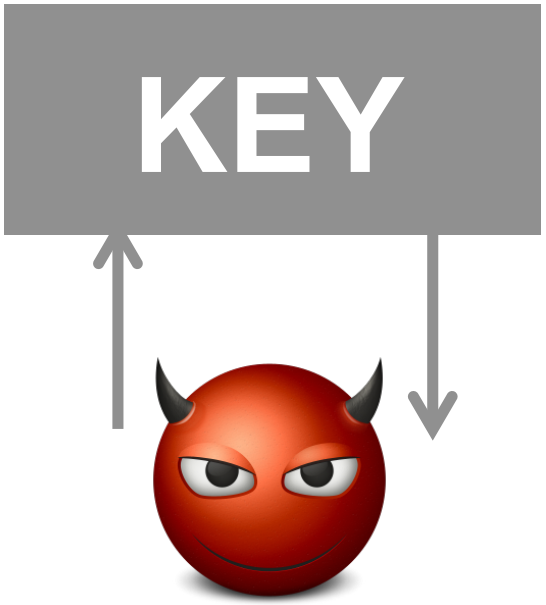
Practical Leakage-Resilient Symmetric Cryptography

Sebastian Faust — **Aarhus University**
Krzysztof Pietrzak — IST Austria
Joachim Schipper — IST Austria

Theory vs. Reality

AARHUS UNIVERSITY

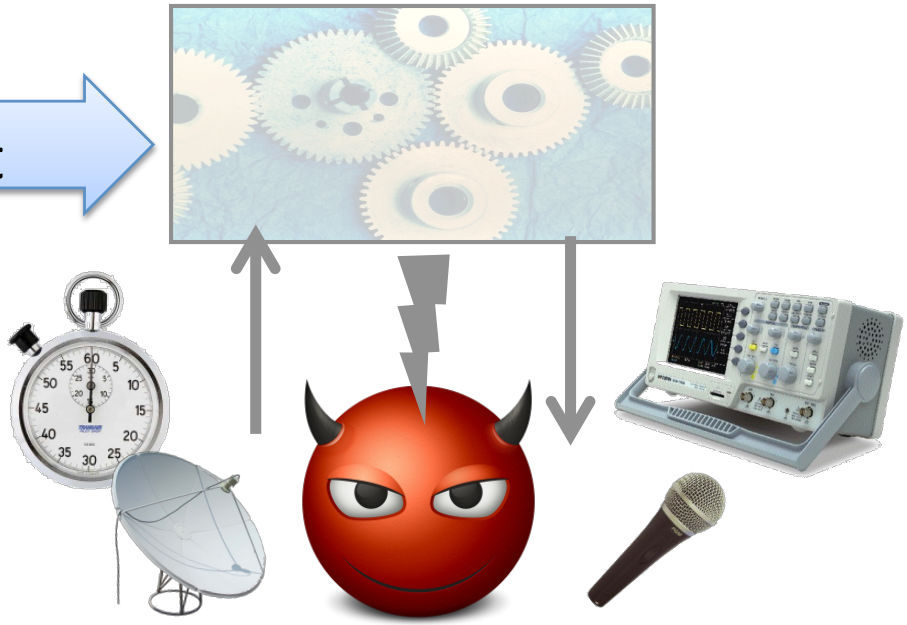
Black-box analysis:



Controls inputs /outputs but internals stay hidden

Goal: Prove **no** adversary can break security in model

Attack Implementation:



Devices are not black-boxes: **leak** about internals

Provable schemes get broken in practice

Leakage Resilient Crypto

AARHUS UNIVERSITY

Include leakage into model

Devise new s n security

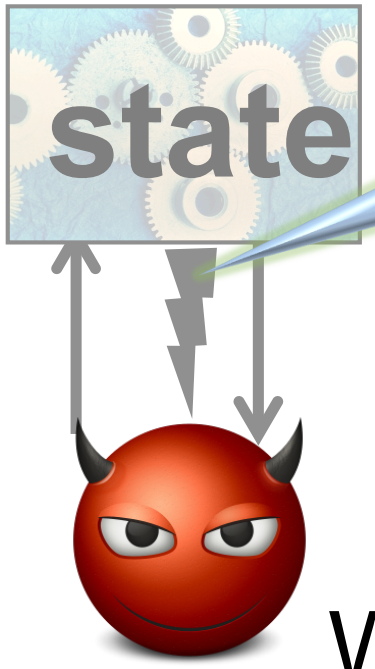
How to model this?

Prove that no attack possible given leakage

Most works: PKE, Sigs, IBE, MPC, ZK, …

This work: symmetric schemes

How to model leakage?



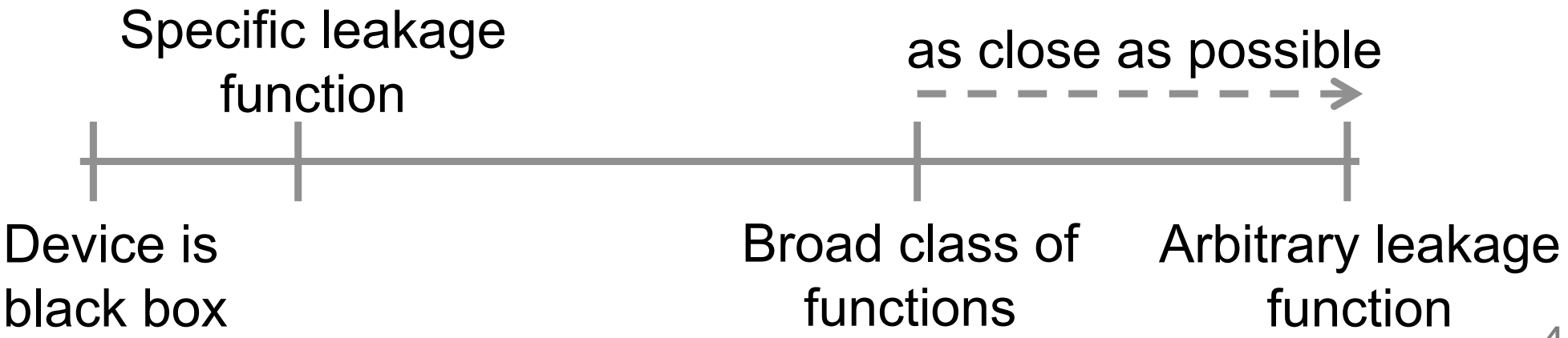
Modeled by leakage function f
Adversary obtains f(state)

Arbitrary function? No!

➔ e.g.: f(state) = key means no security

Some restrictions are necessary

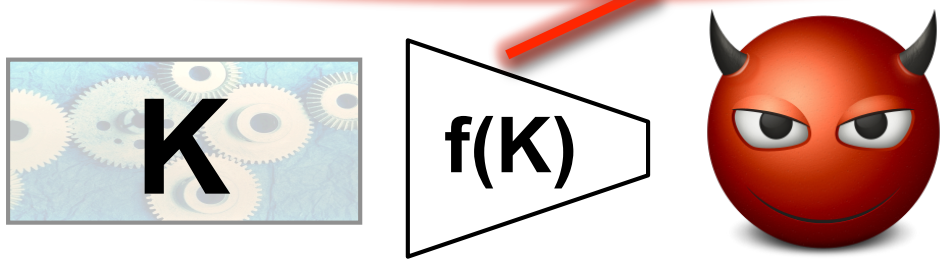
What are minimal restrictions?



Broad class of leakages

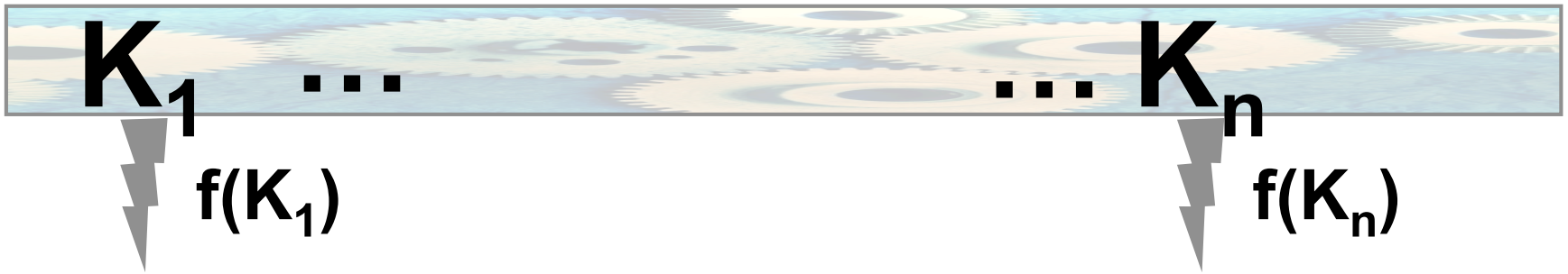
AARHUS UNIVERSITY

All input shrinking functions



Sufficient: Leakage leaves (pseudo)entropy in the key

Continuous leakage: many observations!

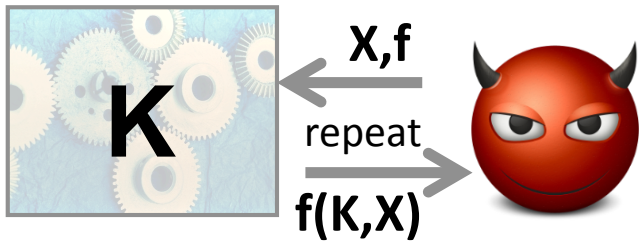


Models attacks that exploit a limited amount of information per observation

More details on model

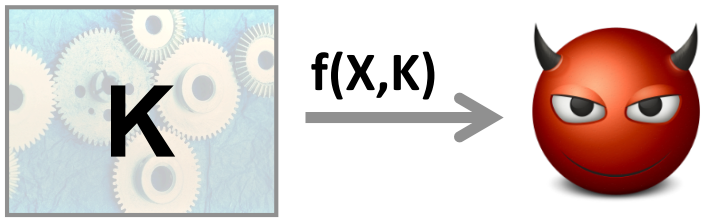
AARHUS UNIVERSITY

Adaptive model [DP08]



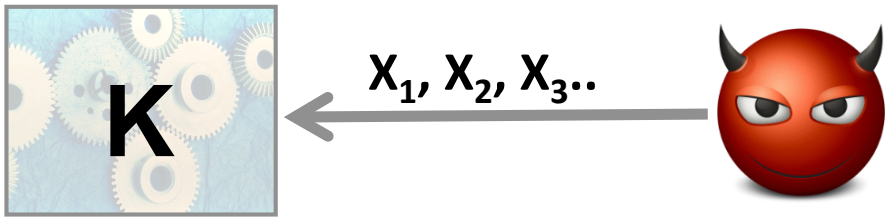
Adaptively chosen leakage function

Fixed leakage model [SPY+]



In practice: leakage function fixed by device!

For PRF/PRP: non-adaptive inputs



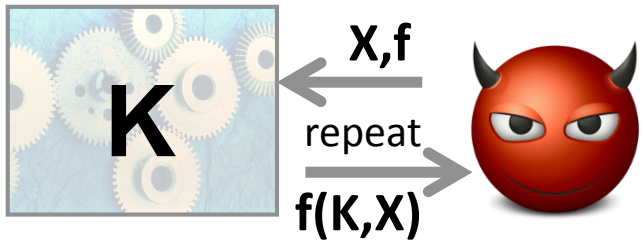
Inputs are fixed from beginning

*Dziembowski, Pietrzak (FOCS'08) *Standaert et al (Hardware Intrinsic Security'10)

More details on model

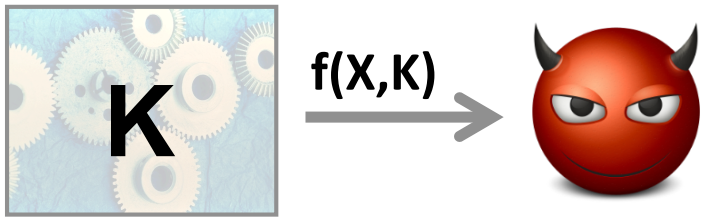
AARHUS UNIVERSITY

Adaptive model [DP08]



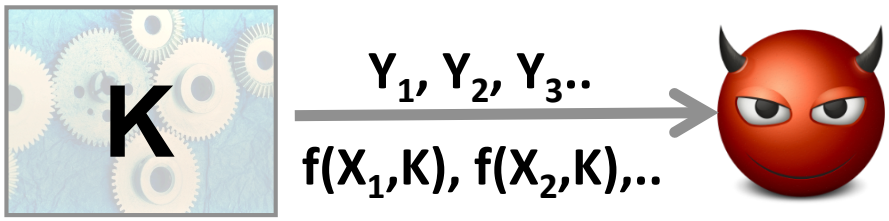
Adaptively chosen leakage function

Fixed leakage model [SPY+]



In practice: leakage function fixed by device!

For PRF/PRP: non-adaptive inputs



Inputs are fixed from beginning

Models SCA that exploit leakage from random inputs

*Dziembowski, Pietrzak (FOCS'08) *Standaert et al (Hardware Intrinsic Security'10)

Rest of this talk

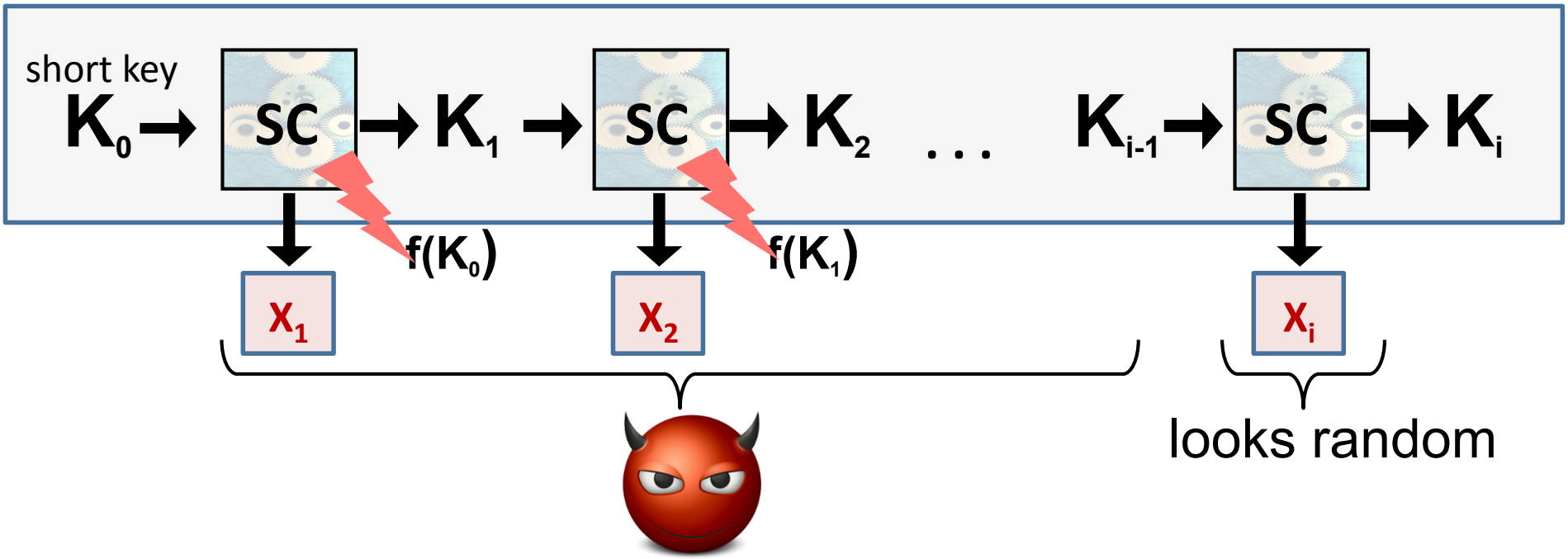
AARHUS UNIVERSITY

Design principles for symmetric crypto…

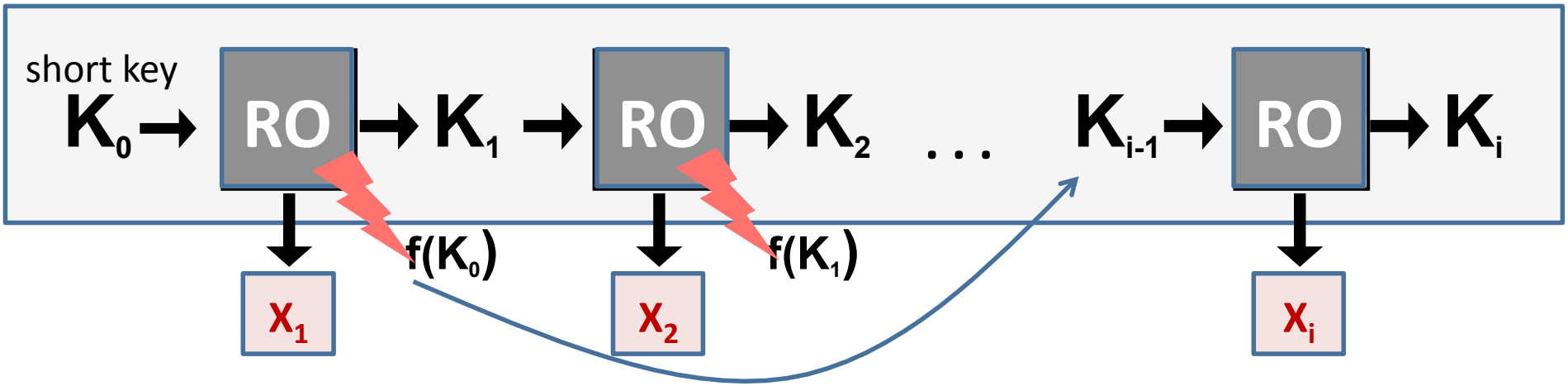
…leakage resilient stream ciphers

…non-adaptive LR PRF and PRP

Stream ciphers



Stream ciphers



Can such a construction be secure? No!
Pre-computation attack: leaks about future keys

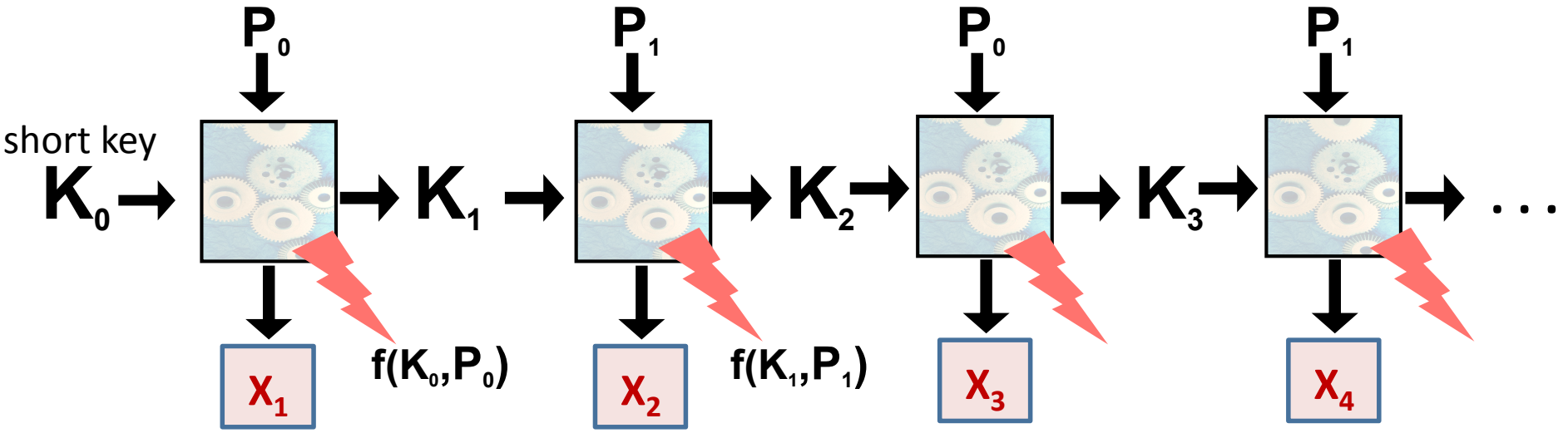
Security proof in the RO model* –
Leakage independent of implementation

Constructions without RO assumption –**
Complicated scheme needed solely for security proof

Simpler constructions without RO?

*Yu, Standaert, Pereira, Yung (CCS'10) **Dziembowski, Pietrzak (FOCS'08), Pietrzak (Eurocrypt'09)

Towards natural schemes*



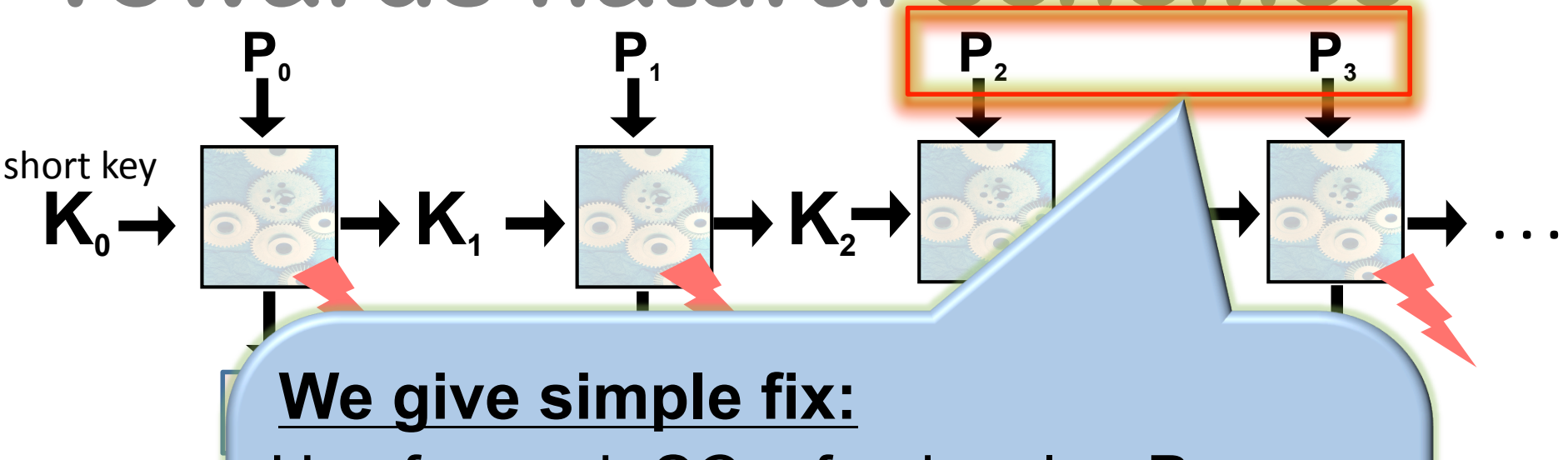
Additional public inputs –
Each execution takes additional input P_0 or P_1

Fixed leakage function –
Otherwise pre-computation attack possibe

Looks promising –
Unfortunately, we don't know how to prove it

*Yu, Standaert, Pereira, Yung (CCS'10)

Towards natural schemes*



Additi

Each e

Fixed

Otherv

Looks

Unfortun

We give simple fix:

Use for each SC a fresh value P_i

➔ Not very useful in practice!

Will be useful to construct "practical" leakage resilient PRFs

Upcoming: Yu and Standaert show security in minicrypt.

*Yu, Standaert, Pereira, Yung (CCS'10)

Rest of this talk

AARHUS UNIVERSITY

Design principles for symmetric crypto….

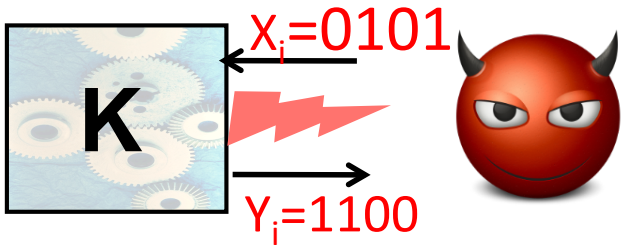
…leakage resilient stream ciphers

…non-adaptive LR PRF and PRP

Leakage Resilient PRFs

AARHUS UNIVERSITY

Idea: Looks as random function even given leakage



For new X output Y looks random even given previous leakages

Standard PRFs build with GGM tree –
MR04*: "GGM tree useful against leakage attacks"

Previous constructions –

- Simple GGM with RO**: strong assumption!
- Tailored GGM***: complicated construction!

Simpler constructions without RO assumption?

*Micali, Reyzin (TCC'04) **Micali, Reyzin (TCC'04) *** Dodis, Pietrzak (Crypto'10)

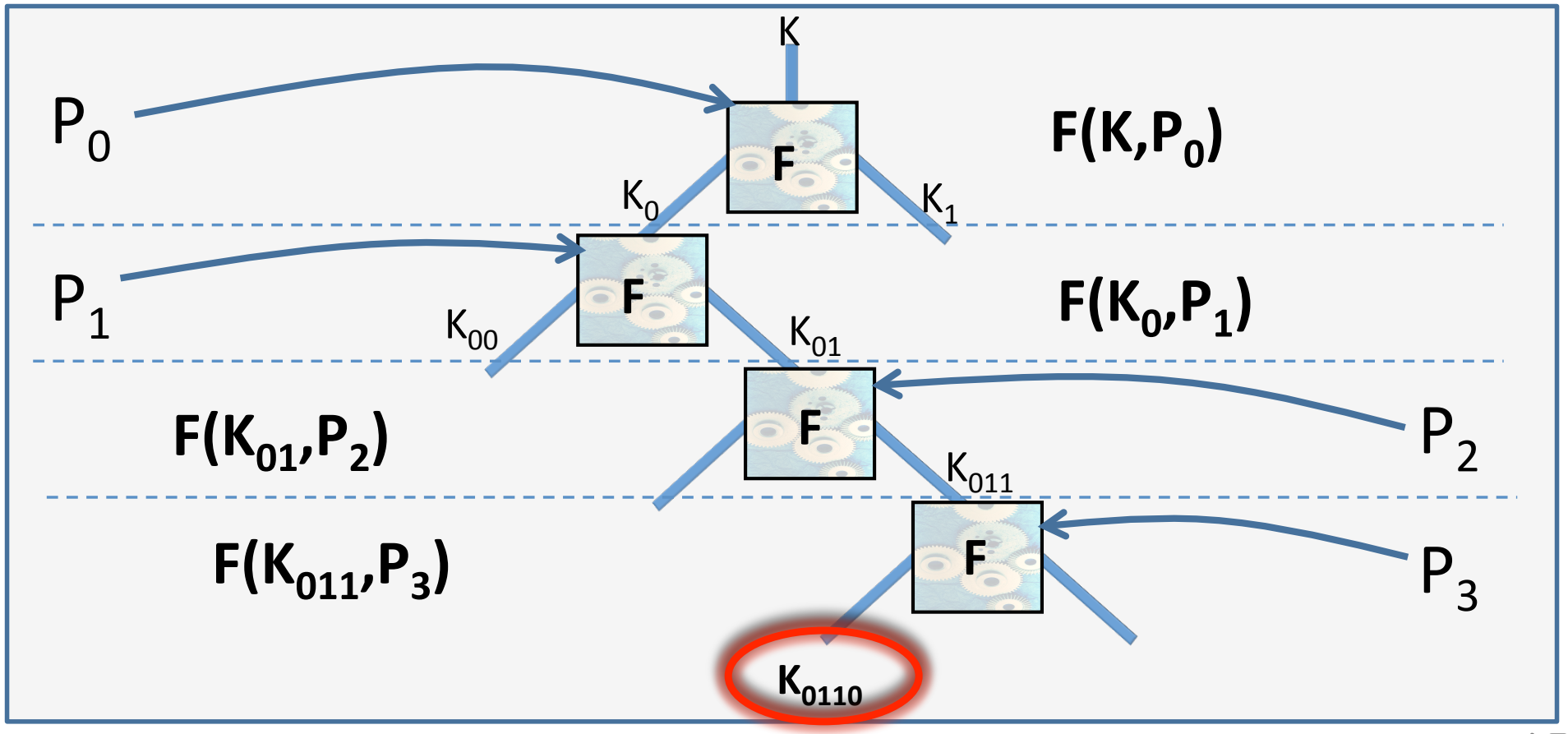
Our construction

AARHUS UNIVERSITY

Instantiate GGM with our simple SC

Use random public value P_i for each level of tree

State: K, P_0,… P_3 Input: 0110 Output: G(K,0110)



Our results

AARHUS UNIVERSITY

Theorem 1: **Leakage resilient** PRF when inputs are chosen **non-adaptively** for all leakage queries

We don't know if scheme secure with adaptive inputs

Theorem 2: 3-round Feistel with LR PRF is LR PRP when inputs are chosen non-adaptively

Complements DP-10*: Feistel with log-number of rounds is never LR PRP with adaptive inputs

Implementation from AES: 48k public randomness
Yu-Standaert: 128bit public randomness

* Dodis, Pietrzak (Crypto'10)

Take home message

AARHUS UNIVERSITY

Theory and practice shall work together to achieve better real-world security

What is the “right” model?

Which “practical” ideas can be backed in theory?

Which “theory” ideas are practical?

Thank you

Our construction

AARHUS UNIVERSITY

Inst… SC

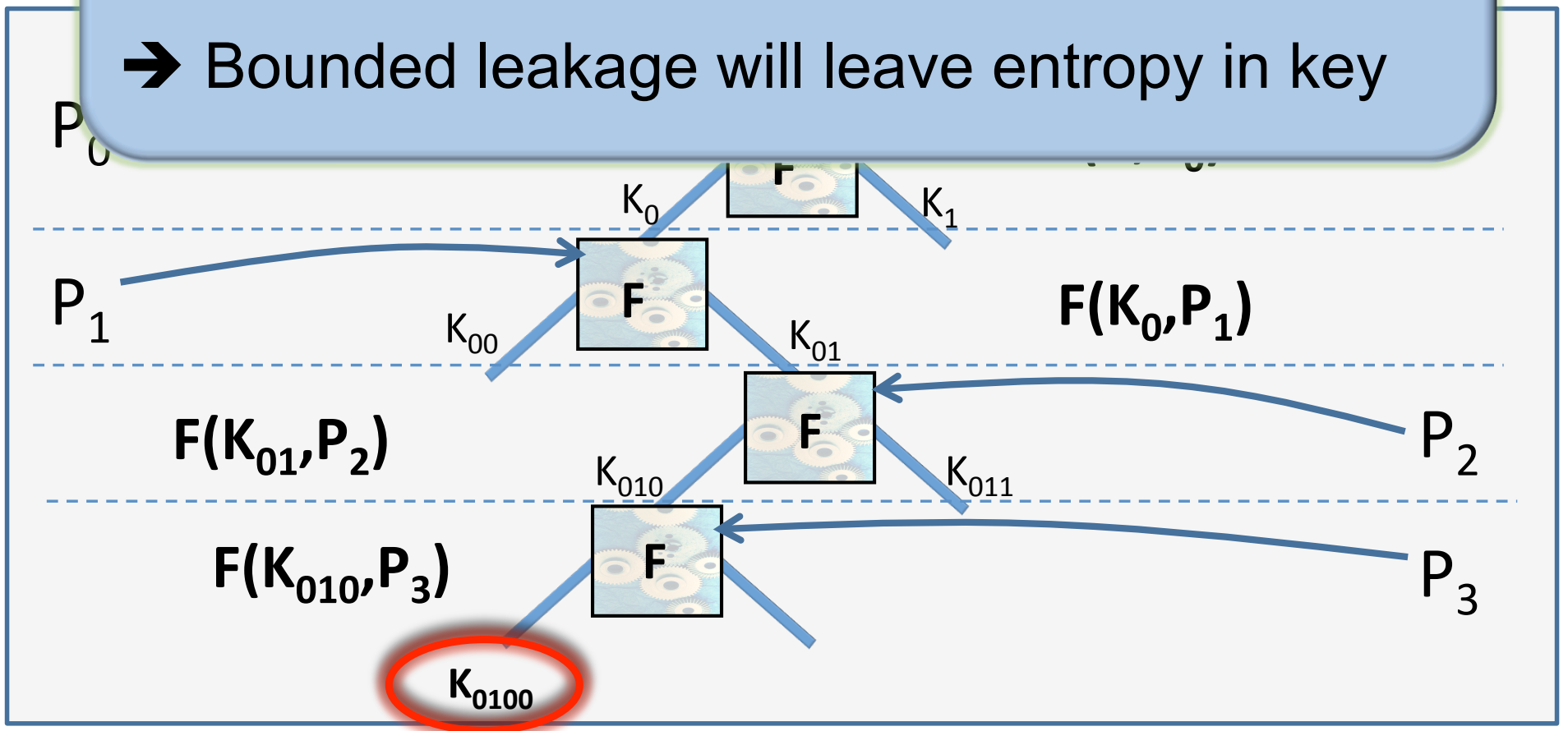
U…

St…

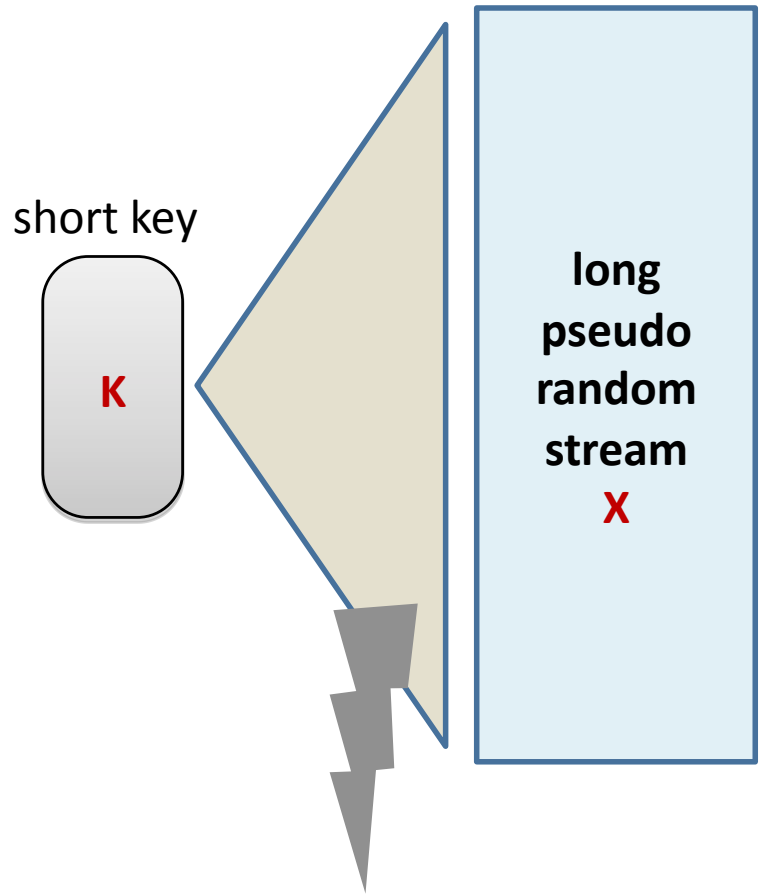
Main idea of proof:

Each key leaks only twice

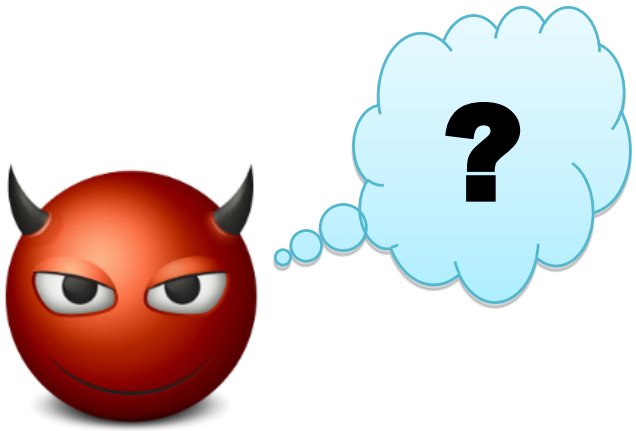
➔ Bounded leakage will leave entropy in key



Stream ciphers



Pseudorandomness: no efficient (PPT) adversary can distinguish **X** from random

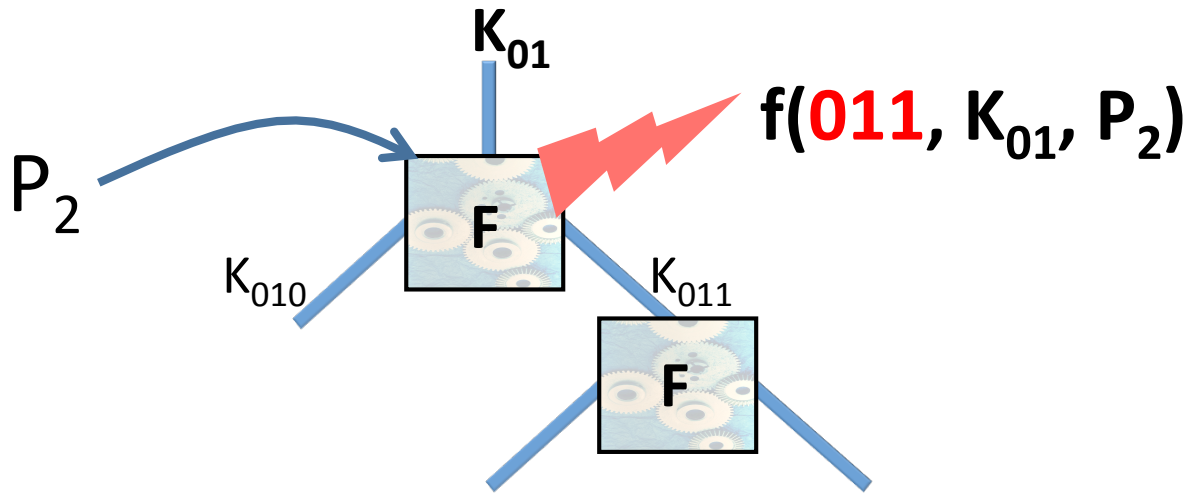


Leakage resilient stream cipher –
Output looks random even given leakage

Our construction

AARHUS UNIVERSITY

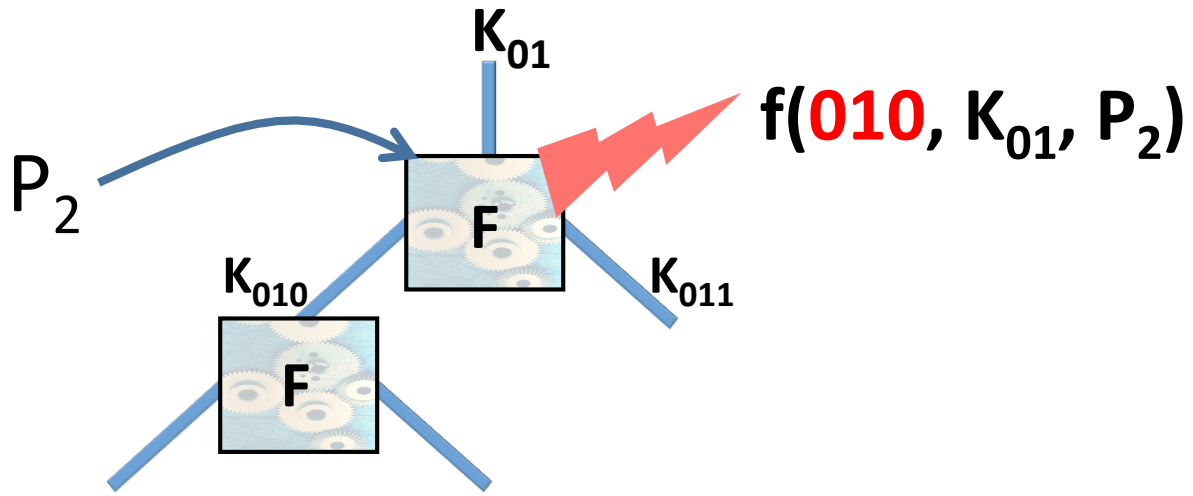
Main observation for proof: 2-limited data complexity



Our construction

AARHUS UNIVERSITY

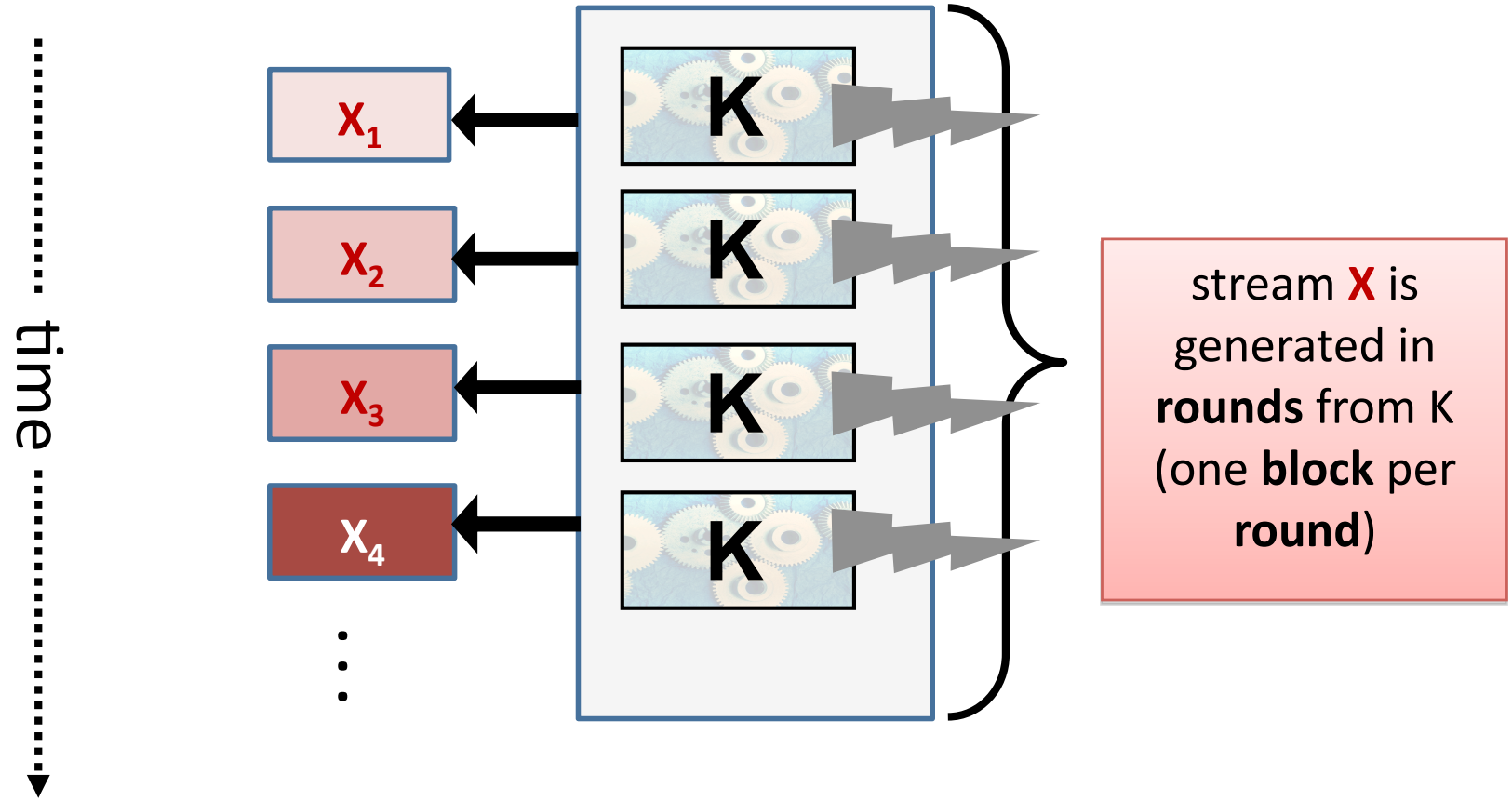
Main observation for proof: 2-limited data complexity



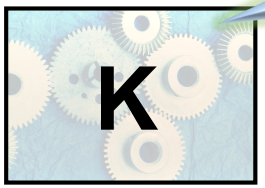
Adversary gets only 2 bounded leakages –
Leakages leave enough "entropy" in each key

Does this intuition suffice for the proof?

Stream ciphers in practice



In this talk we don't describe concrete algorithms



Think of it as an execution of AES