Unifying Leakage Models

From Probing Attacks to Noisy Leakage

Sebastian Faust

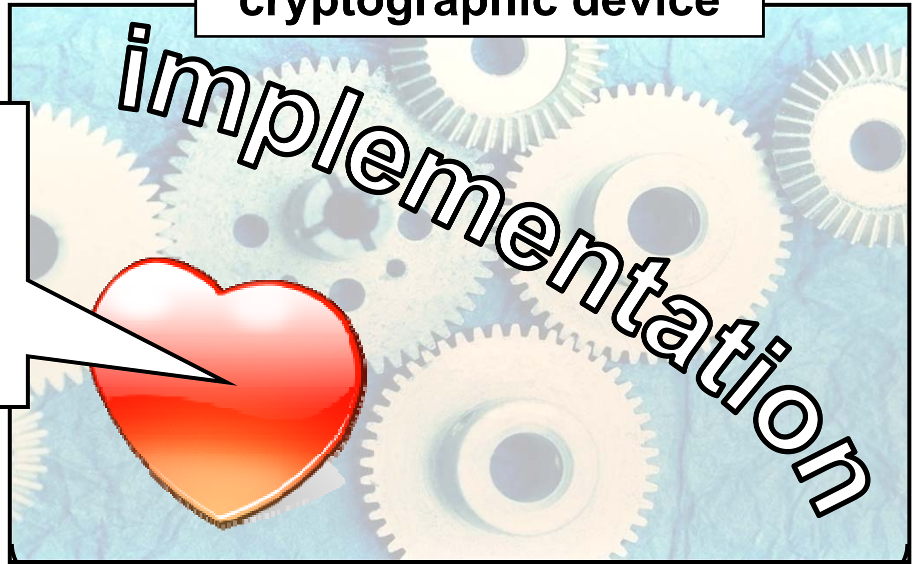
EPFL, Switzerland

Joint work with:

Alexandre Duc, Stefan Dziembowski

Crypto Implementations

cryptographic device



very secure

– well-defined mathematical object

– often proof-driven security analysis

much less secure!

– new attacks possible on crypto implementations:

➔ side-channel leakage devastating for security

Q: Proof-driven security analysis for implementations?

Goal of leakage resilient cryptography

Leakage resilient crypto

Bounded leakage

Probing

Bounded space

Many models

Low-depth circuits

Hard-to-invert leakage

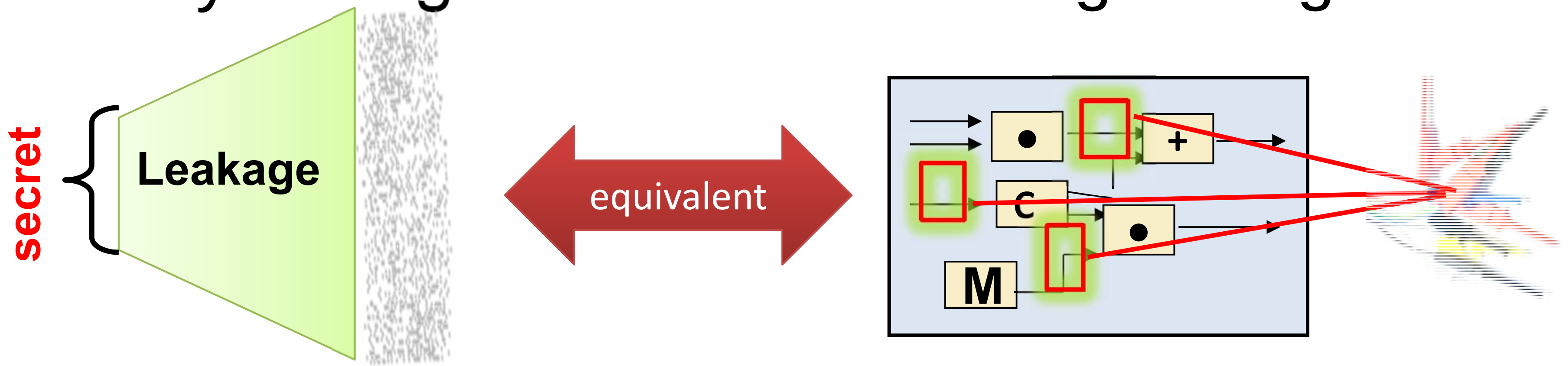
Beautiful theory! <u>Q:</u> Does it match practice?

Models do **not** match with my engineering experience

Leakages are not quantitatively bounded

Our main observation

Noisy leakage | Probing leakage



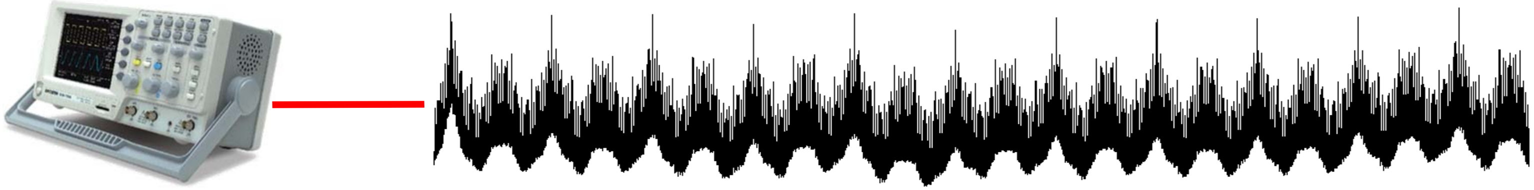
Preferred model in practice

Simpler model than bounded leakage

Improved security analysis of masking countermeasure

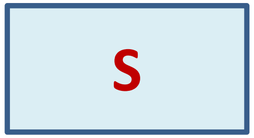
Masking countermeasure

Common countermeasure against power analysis



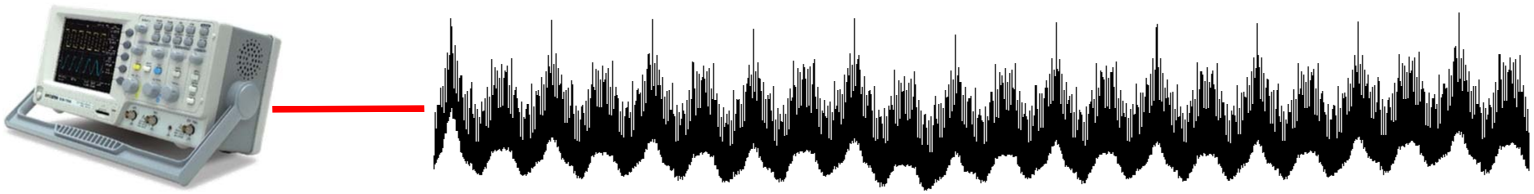
Basic idea: protect sensitive information by randomized encoding

Additive secret sharing:



Masking countermeasure

Common countermeasure against power analysis



Basic idea: protect sensitive information by randomized encoding

Additive secret sharing („+“ is field addition, e.g., $GF(2^8)$ for AES):

S —Encode→ **C := (C_1,...C_n) random s.t. S = C_1 +...+ C_n**

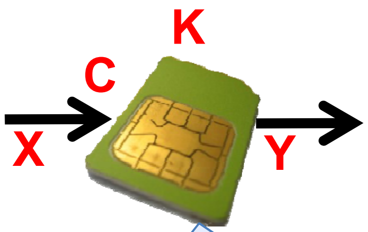
S is hidden if leakage depends on **n-1** shares only

Protects against **n-1** probing attacks

How to extend to computation?

Leakage resilient circuits

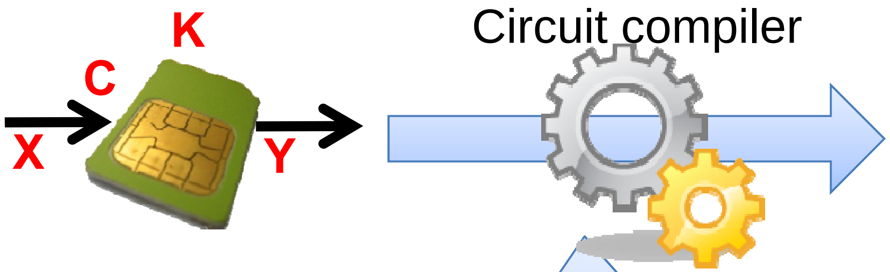
Formalization of masking by Ishai-Sahai-Wagner-03



Arbitrary algoritm with input **X**, output **Y** and state **K** described as a circuit, e.g., AES

Leakage resilient circuits

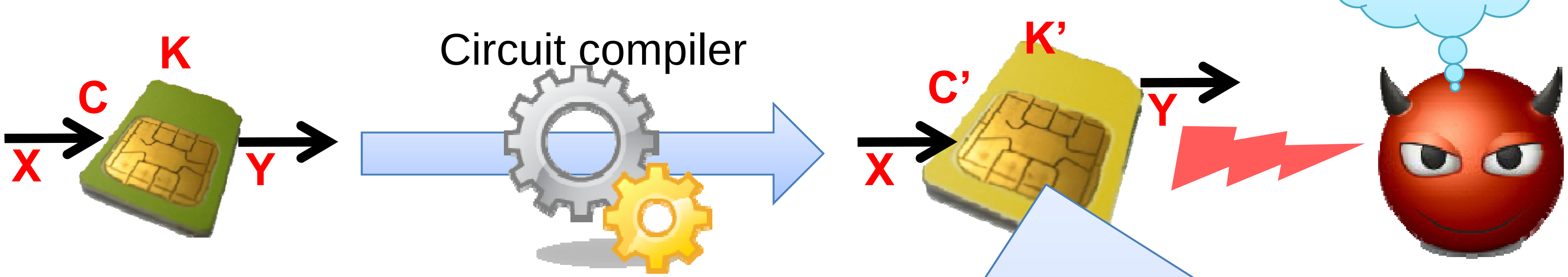
Formalization of masking by Ishai-Sahai-Wagner-03



Run only once at production time (no leakage!)

Leakage resilient circuits

Formalization of masking by Ishai-Sahai-Wa… 3



Output: Description of circuit **C'** with key **K'**

Correctness: **C[K]** and **C'[K']** have same functionality

Additionally: **C'[K'] leakage resilient** for many executions

Security: adversary learns nothing "useful" from leakage

(formalized by simulation-based security)

Leakage models?

ISW secure against probing

Leakage bounded per observation: learn **n-1** wires

Proves soundness of masking scheme

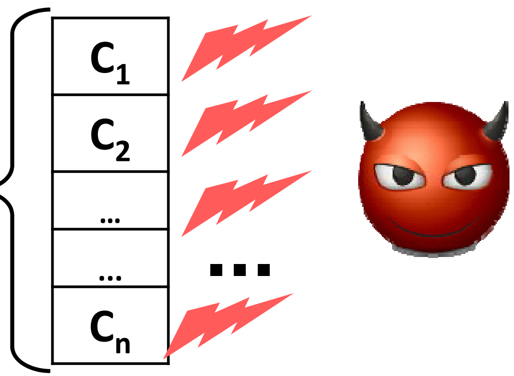
Probing: oblivious of large parts of computation

More realistic: no quantitative bound but noisy leakage

Chari et al. 99:

only encoding

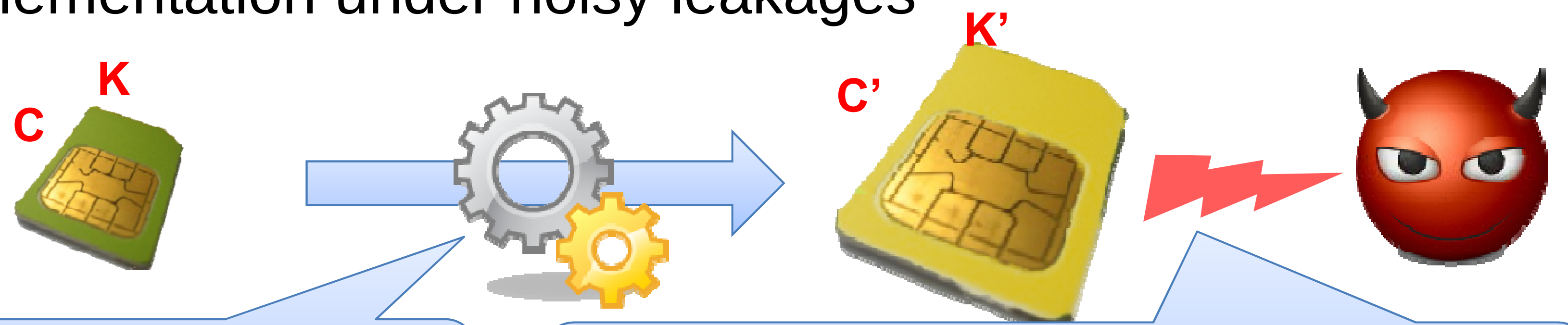
Enc(s)



Leakage is
c_i + Gaussian noise

PR13: Circuit security

Prouff-Rivain, Eurocrypt 13: Prove security of a masked implementation under noisy leakages



Compiler of ISW03 with leak-free gates

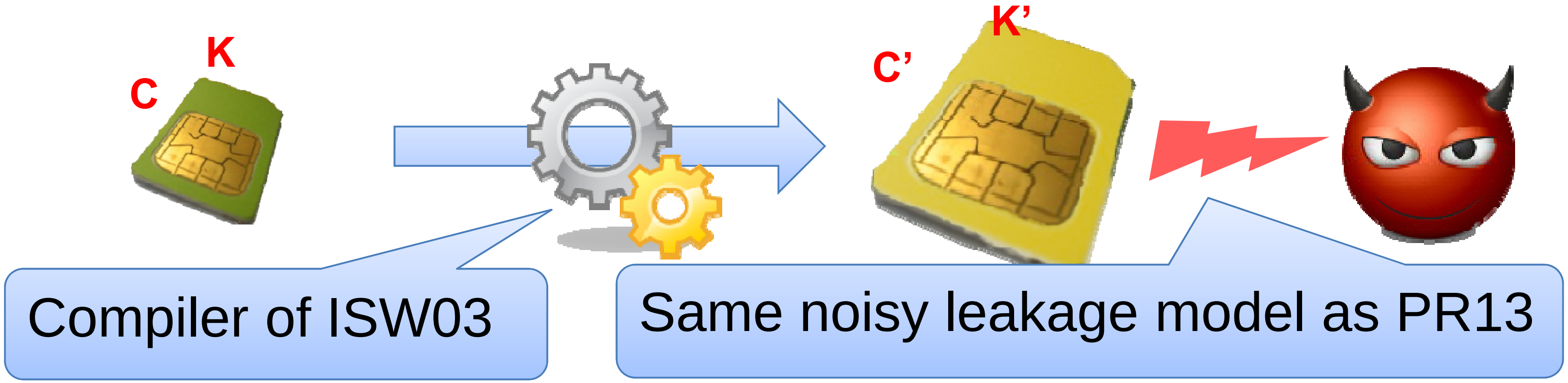
Adversary obtains noisy version of each wire: $N(w_i)$

No quantitative bound on amount of leakage

Drawbacks of the analysis:

- Leak-free gates: no leakage from refreshing
- Security argument only for random-message attack
- Technical proof

Our Results



ISW03 is secure against noisy leakages

- No leak-free gates
- Full simulation-based security analysis
- Unifying leakage models:
  n-probing security ➔ security against noisy leakage

Useful tool: proofs in n-probing model much simpler than proofs in noisy model

Rest of this talk

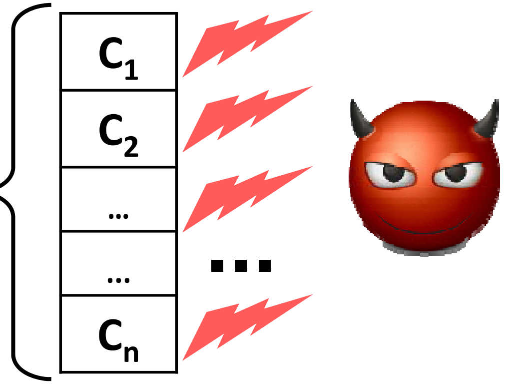
1. The noise model in detail

2. Proof outline

Noise model of PR13

s in F

Enc(s)



Any **p**-Noisy function **N**

➔ adversary learns **N**(**C_i**)

e.g. **N(C$_i$)**: compute Hamming weight and add Gaussian noise

Prouff-Rivain 13: rather complicated definition

We propose simpler equivalent definition:

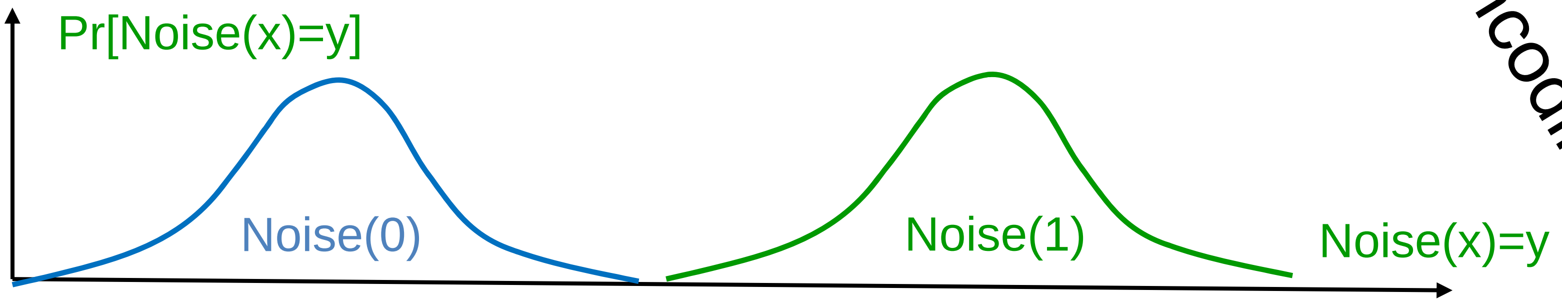
uniform

X_0, X_1 → 1. b ← {0,1}   2. N(X_b)

N(X_0), X_0, X_1   **≈**   **N(X_1), X_0, X_1**

N is **p**-noisy if statistical distance **< p**

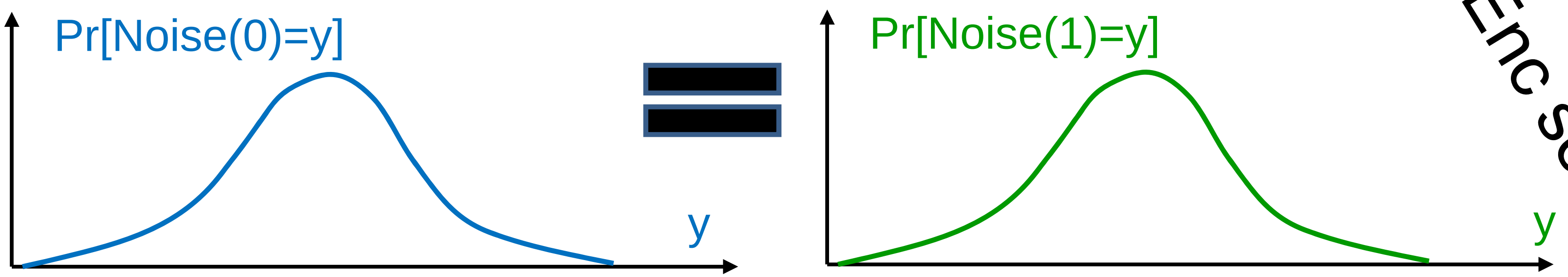
Some examples (F = GF(2))



Encoding insecure

No noise **p ≈ 1**: very informative leakage

➔ Adv. learns **Noise(C_i)**: full knowledge about secret **s**



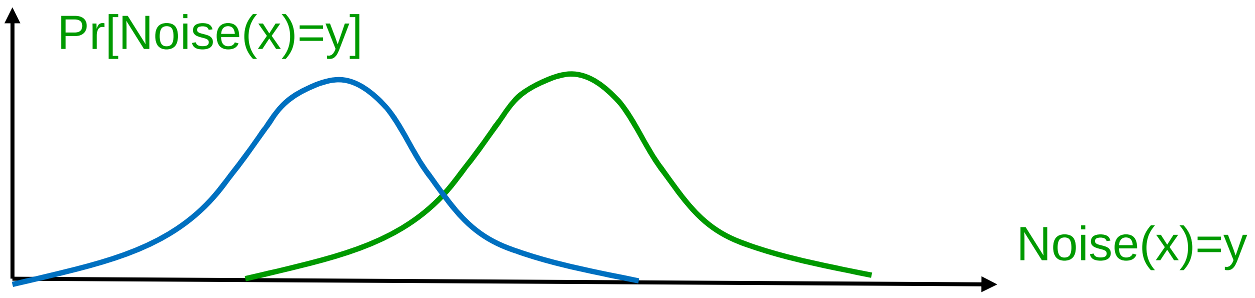
Enc secure

High noise **p = 0**: non-informative leakage

➔ Adv. learns **Noise(C_i)**: no knowledge about **s**

Some examples (F = GF(2))

Interesting case: „some noise“



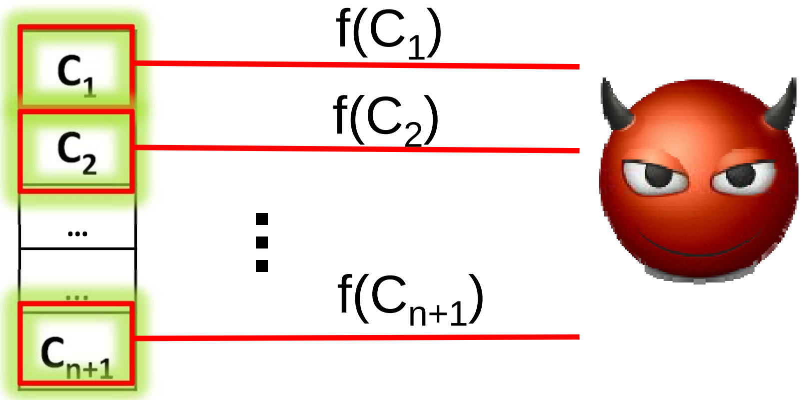
1. Simpler noise model: random probing

2. Random probing = noisy leakages

➔ Simulate noisy leakage with random probing

Proof outline

Random probing model (ISW03)



$f(C_i) = C_i$ with prob. q;
otherwise $f(C_i) =$ „?“

Adv. learns **S** only if „lucky“ in each random probe

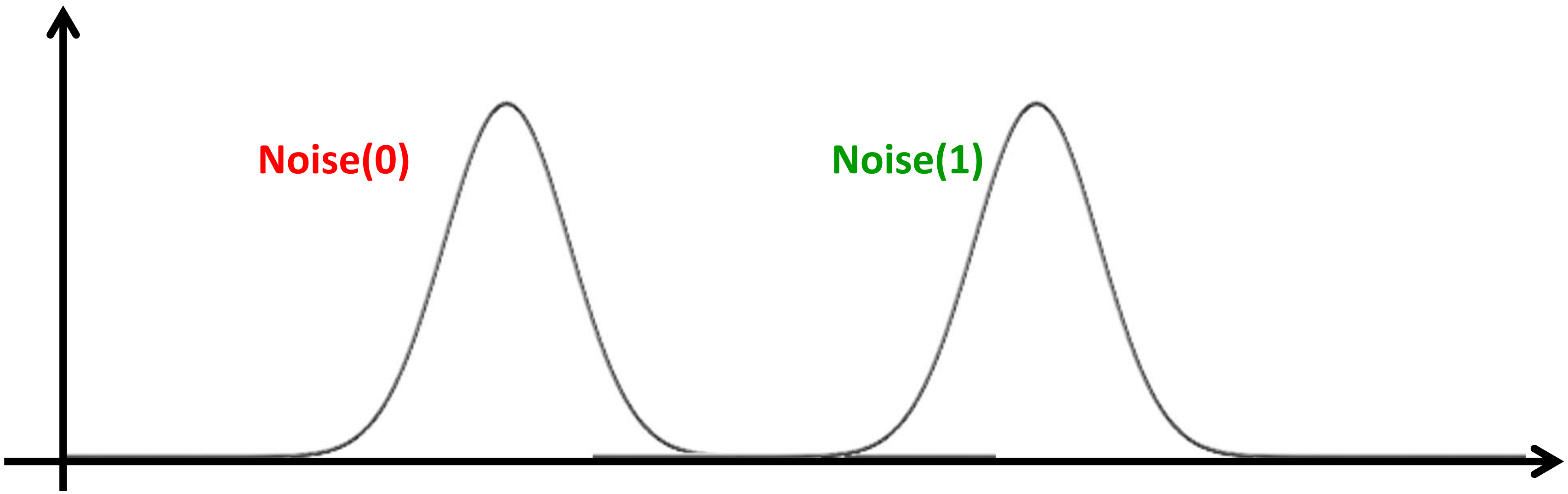
➡ Encoding secure in random probing model for constant q

Proof outline

Random probing ➔ noisy leakage

For any **x** and **Noise(.)** there exists **Noise‘(.)** such that **Noise(x) = Noise‘(f(x))**

First extreme case: “no noise” (**p≈1**)



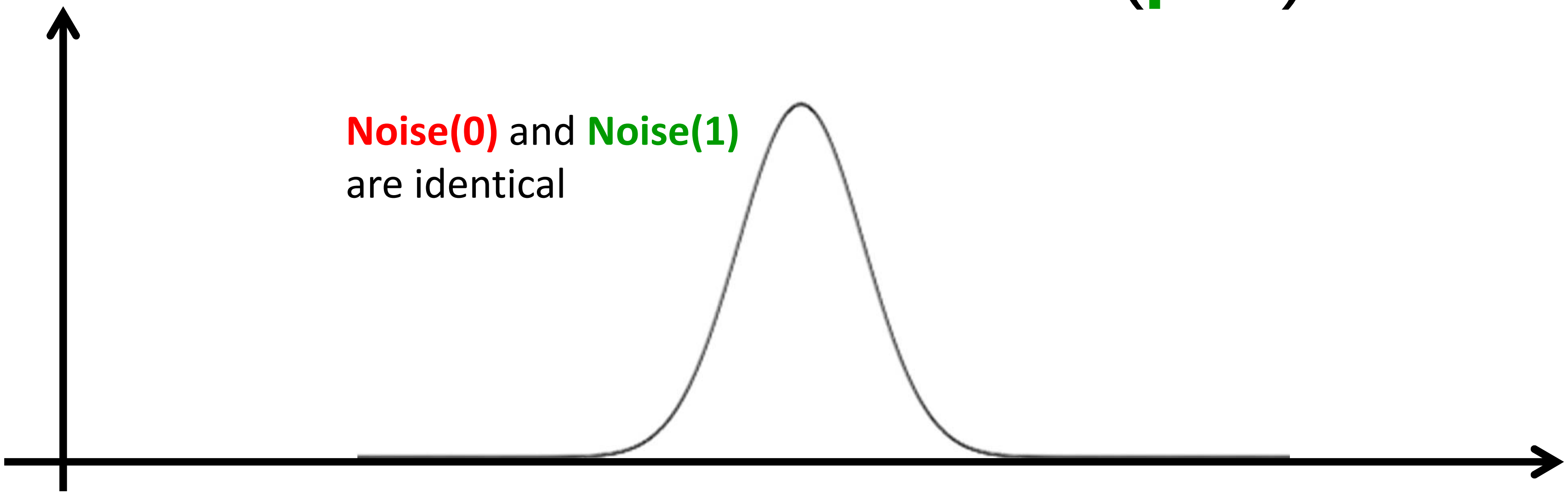
No way to “simulate” this noise except with random probing where **q=1** (i.e., reveals everything).

Proof outline

Random probing ➔ noisy leakage

For any **x** and **Noise(.)** there exists **Noise‘(.)** such that **Noise(x)** = **Noise‘(f(x))**

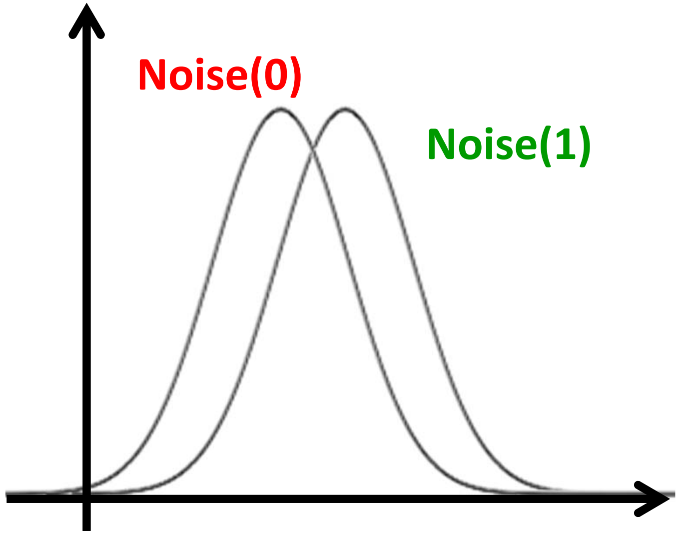
Second extreme case: “full noise” (**p=0**)

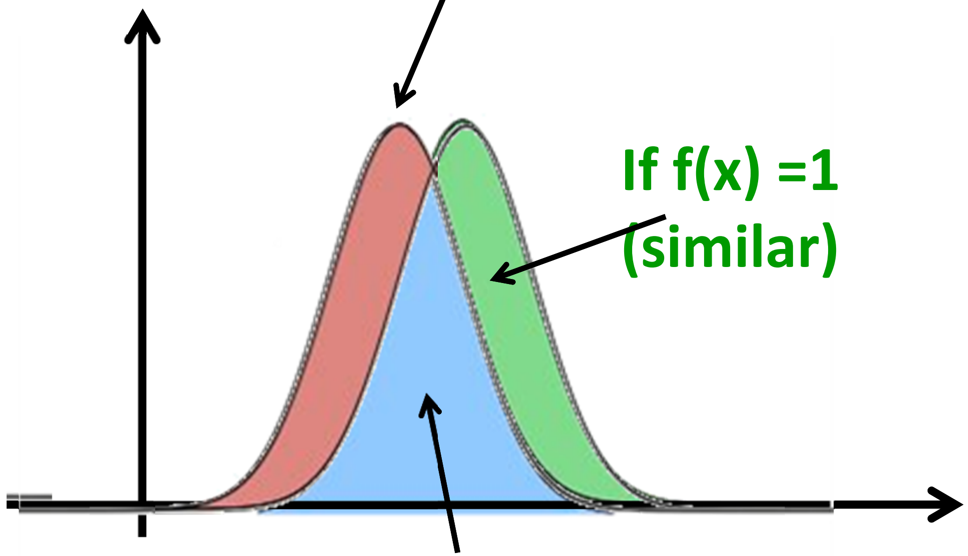


Set **Noise‘** = **Noise**: Simulation is possible without even probing: **q = 0** (i.e., reveals nothing)

Proof outline

Involved case: “some noise”





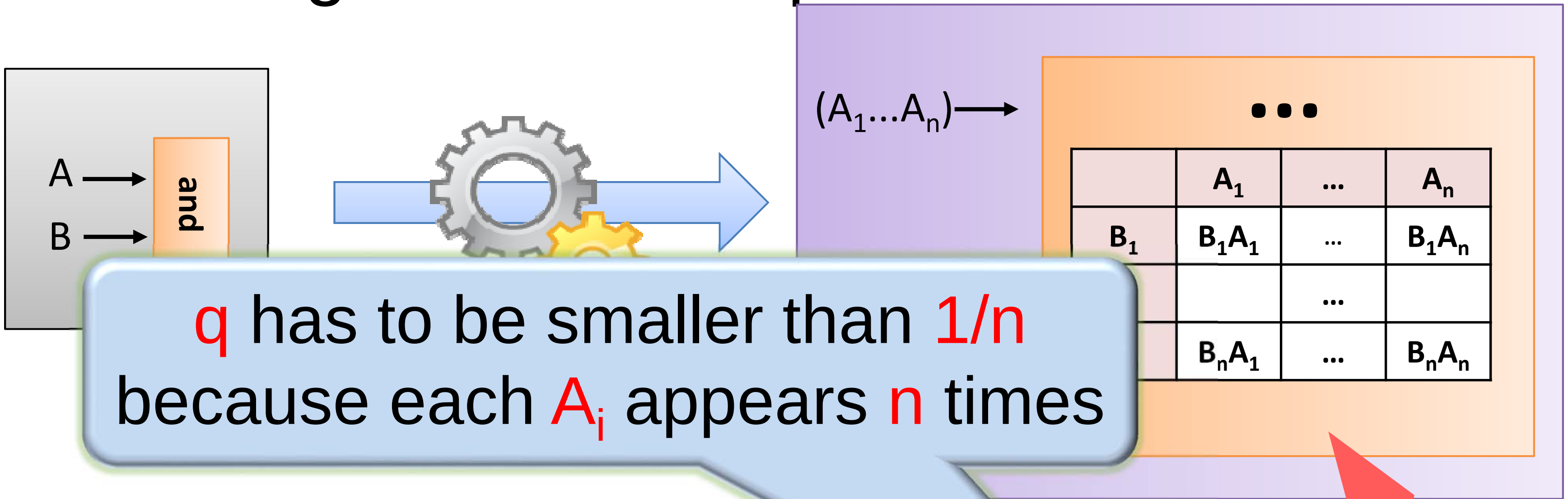
One can “simulate” **Noise(.)** with random probing when probability **q** is exactly **Δ(Noise(0),Noise(1))**

(proof in the paper also for larger fields)

Last step: extend to masked computation

Proof outline

Extending to masked operation

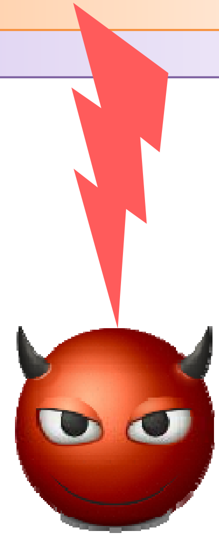


q has to be smaller than 1/n because each A_i appears n times

Given simulator Noise‘ it is sufficient to prove security in random probing model

learns each value with probability q

➔ at least one share B_i, A_j is not learnt



ISW03 is secure in noisy leakage model

Conclusion

ISW03 secure in practically motivated model:

- No leak free gates
- Full simulation-based security
- Usefull tool: probing ➔ security against noisy

Main drawback: requires high nois rate **p=1/n|F|**

Upcoming work: improve bounds (soon to appear)!

Open problems:

Eliminate independence

Practical estimation of noise parameter

Thank you!

Proof outline

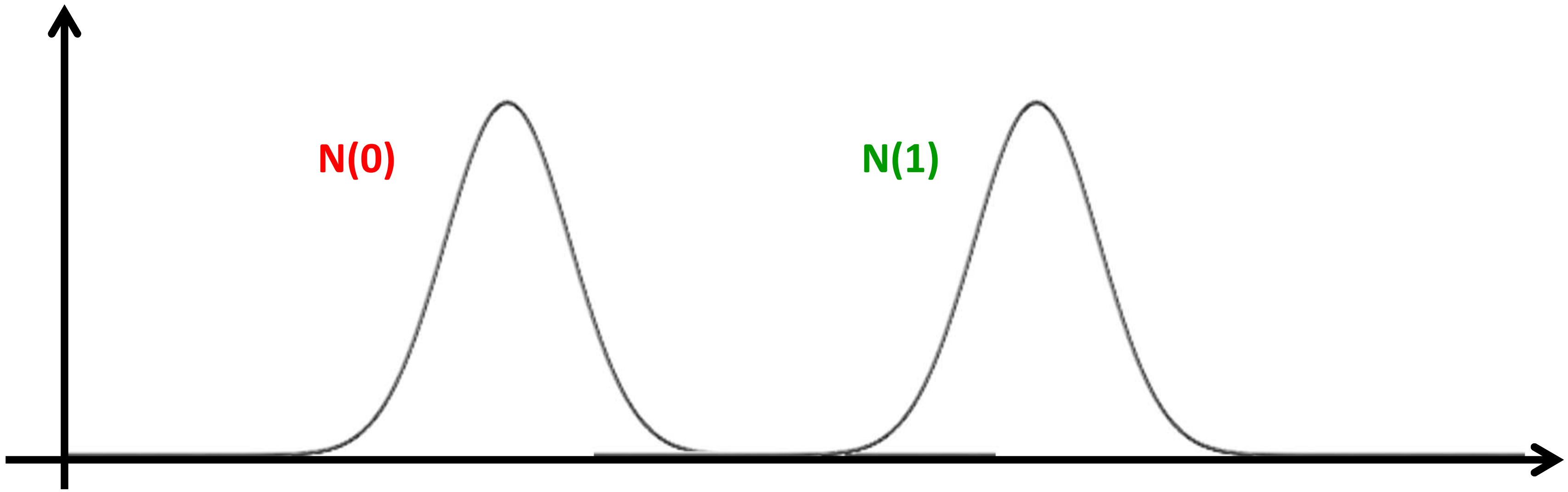
1. Simpler noise model: random probing

2. ISW03 secure in random probing

3. Random probing = noisy leakages

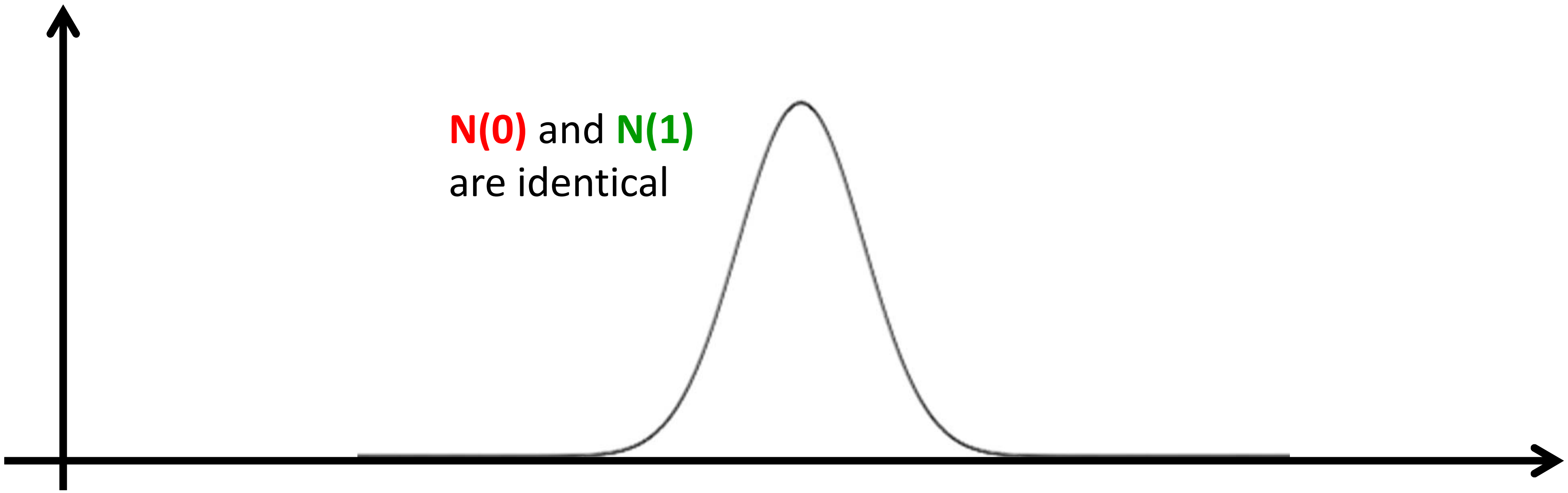
ISW03 secure against noisy leakages

First extreme case: “no noise”



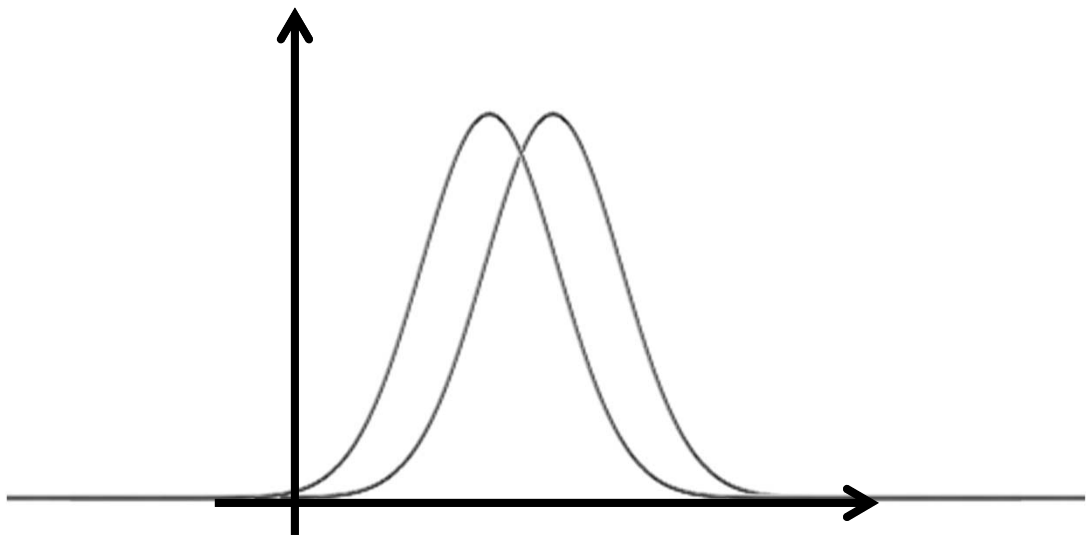
No way to “simulate” this noise except of probing B with probability 1.

Second extreme case: “full noise”



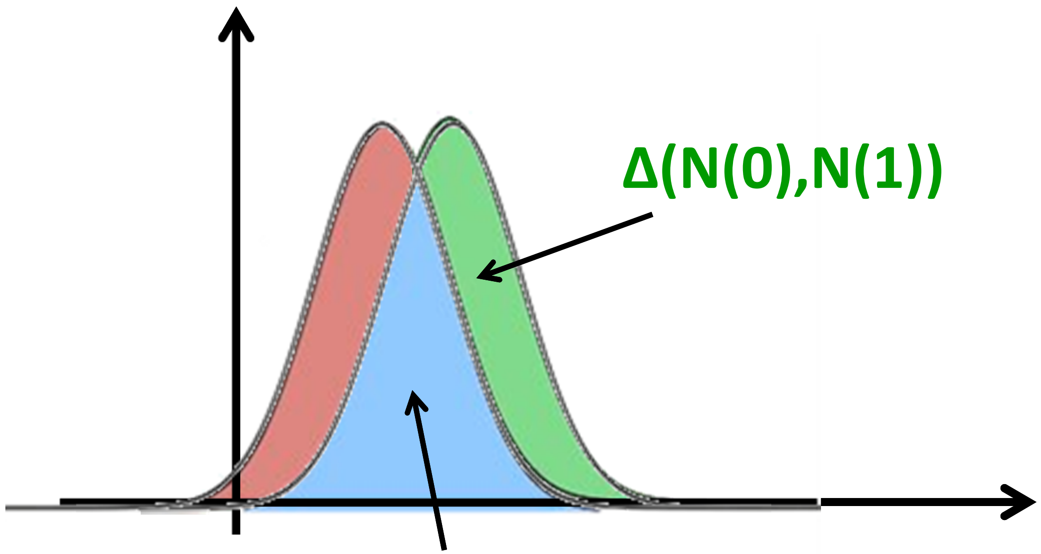
One can “simulate” this noise without ever probing B.

General case:

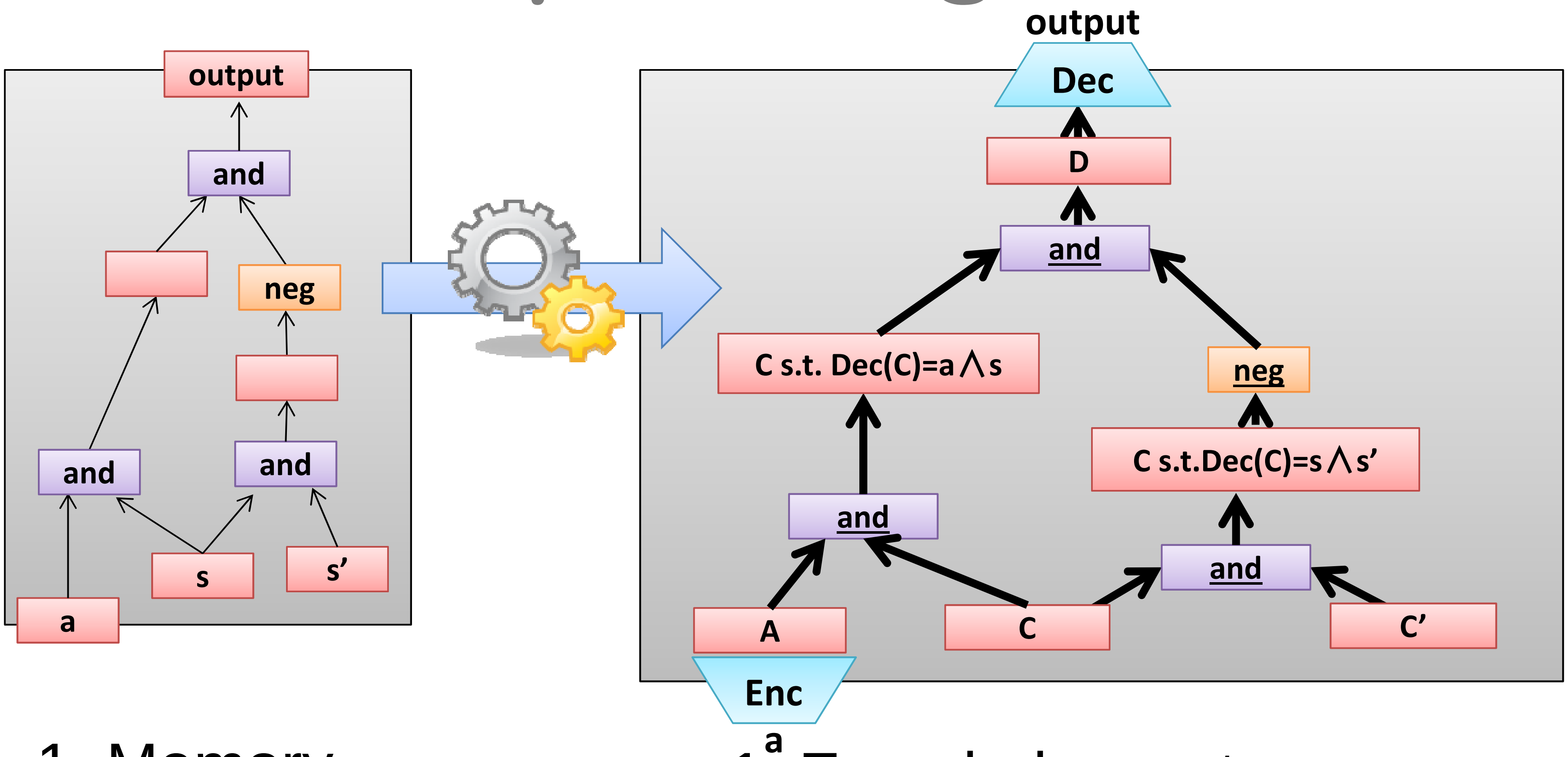


Our observation:

One can "simulate" this noise with probing **B** with probability exactly **Δ(N(0),N(1))**



ISW Compiler: High level



1. Memory
2. Wires
3. Gates

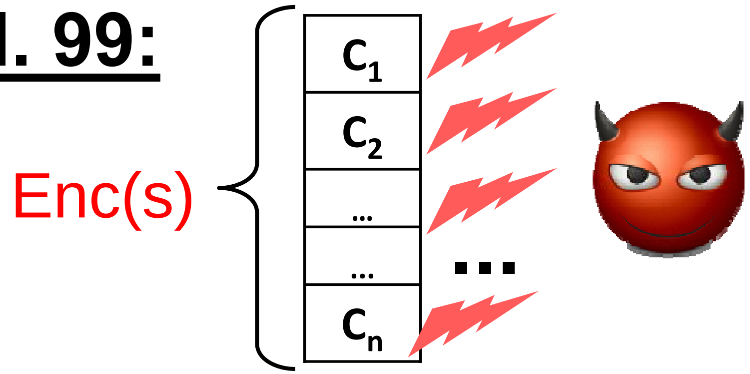
1. Encoded secrets
2. Encoded wires
3. Gadgets compute with encodings

Noisy leakages

Probing model: at least one share is not revealed

More realistic: no quantitative bound but noisy leakage

Chari et al. 99:

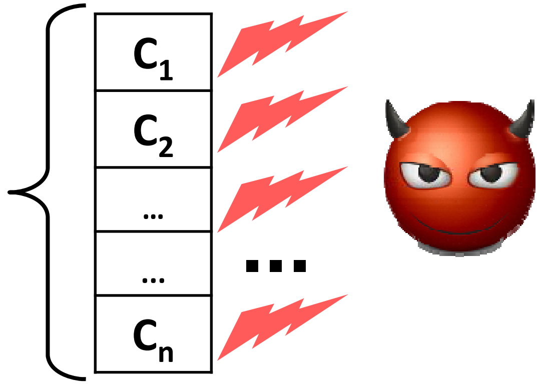


Leakage is c_i + Gaussian noise

Each share leaks but only noisy version of it

Noise model of PR13

Enc(s)



Any **p**-Noisy function **N**

➔ adversary learns **N**($\mathbf{C_i}$)

e.g. **N($\mathbf{C_i}$)**: compute Hamming weight and add Gaussian noise

<u>Prouff-Rivain 13:</u> technical definition

A simpler but equivalent definition:

uniform

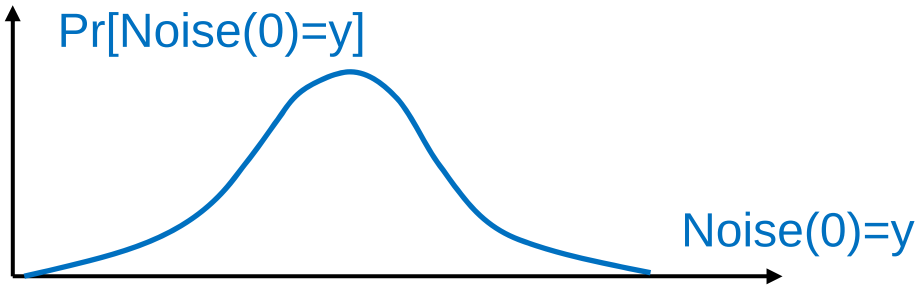
$\mathbf{X_0, X_1}$ → 1. b ← {0,1}   2. N(X_b)

$\mathbf{N(X_0), X_0, X_1}$ ≈ $\mathbf{N(X_1), X_0, X_1}$

N is **p**-noisy if statistical distance **< p**

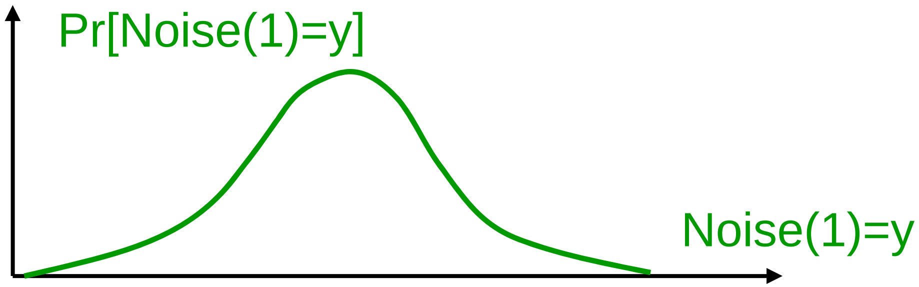
Some easy examples

Maximal noise **p = 0**:



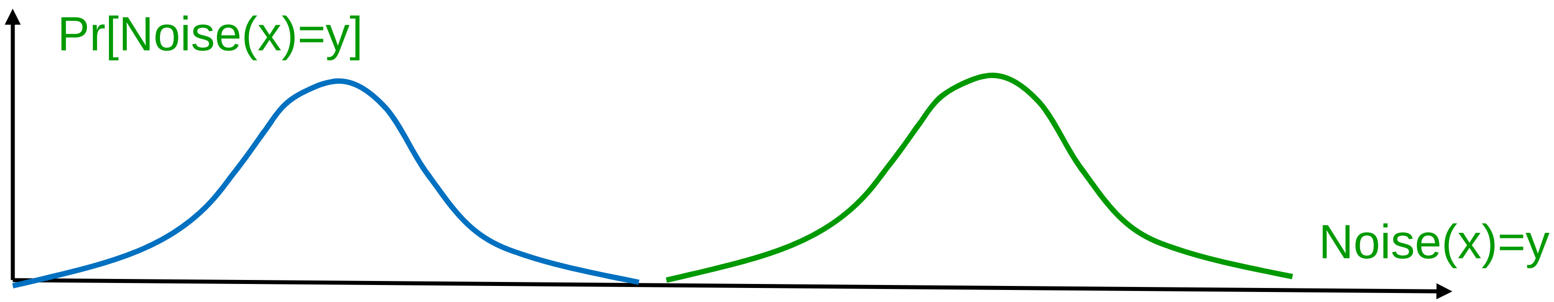
Some easy examples

Maximal noise **p = 0**:



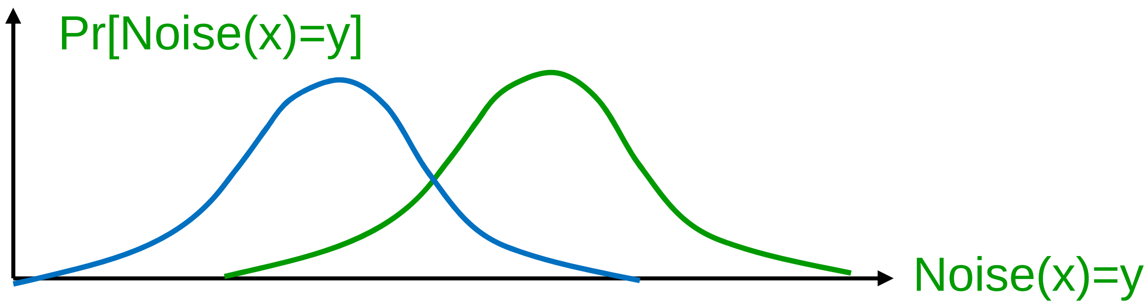
Adversary learns Noise(C_i): no knowledge about secret **s**

Small noise **p ≈ 1**:

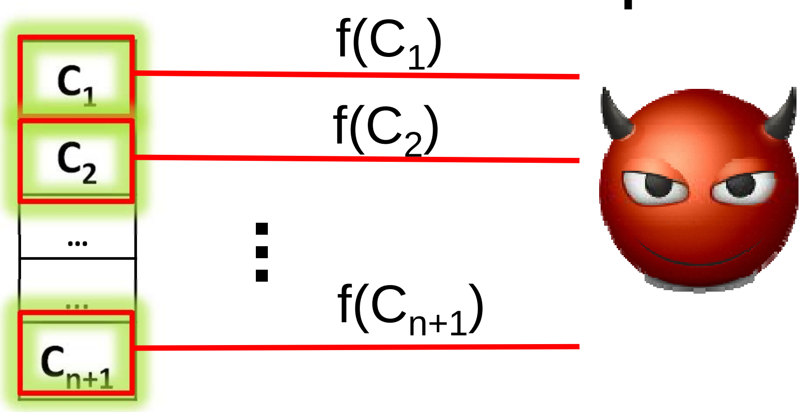


Adversary learns Noise(C_i): full knowledge about secret **s**

Interesting case



Reduce to random probing model (ISW03)



$f(C_i) = C_i$ with prob. q;
otherwise $f(C_i) =$ „?"

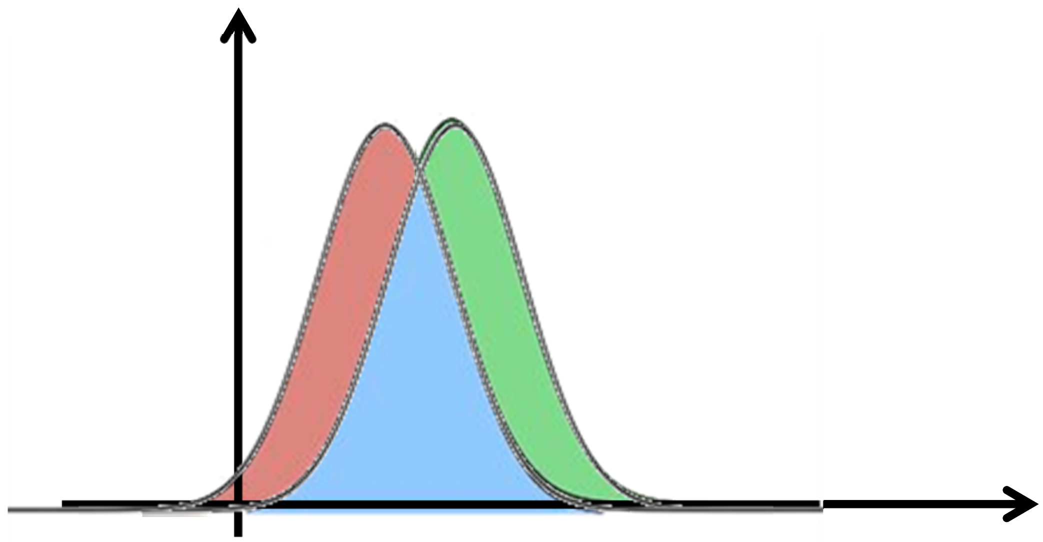
Adv. learns **S** only if „lucky" in each random probe

➡ Encoding secure in random probing model

Reduce to random probing

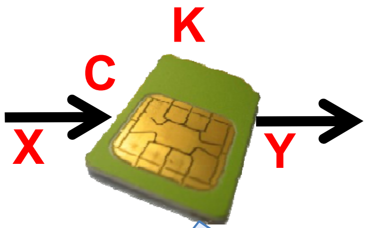
For any x and Noise there exists Noise‘ such that Noise(x) = Noise‘(f(x))

How to define Noise‘(.)?



Leakage resilient circuits

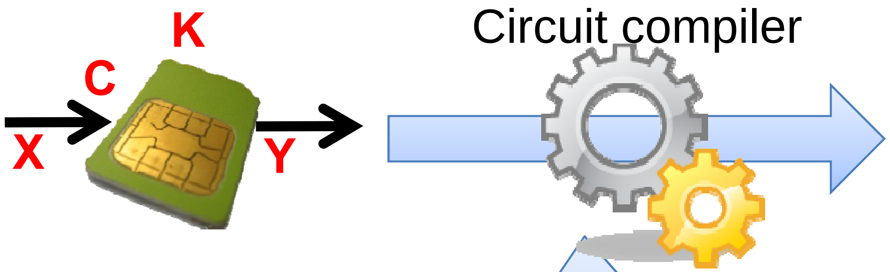
Formalization of masking by Ishai-Sahai-Wagner-03



Arbitrary algoritm with input **X**, output **Y** and state **K** described as a circuit, e.g., AES

Leakage resilient circuits

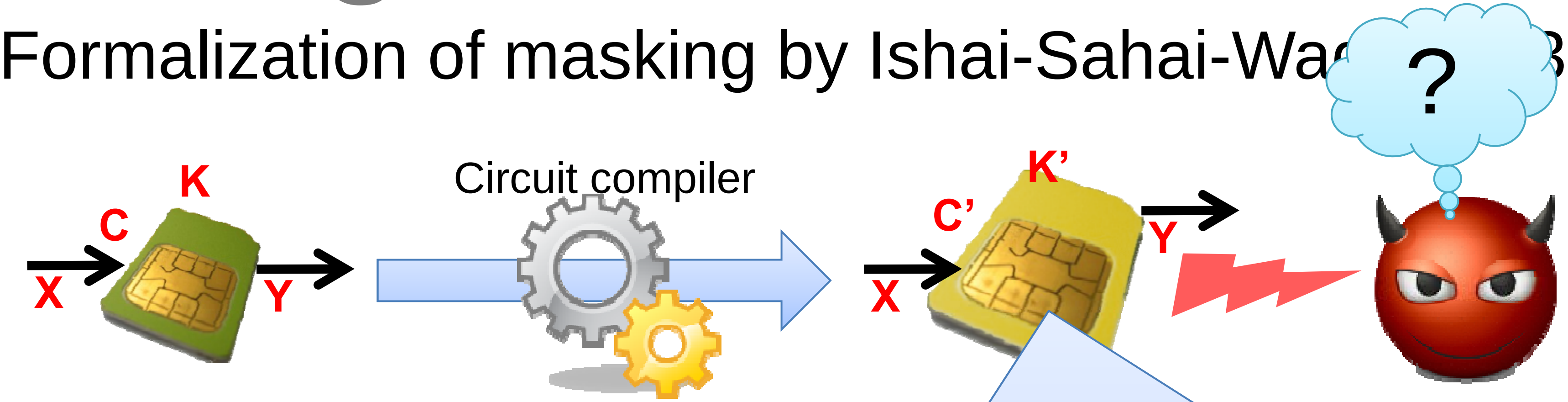
Formalization of masking by Ishai-Sahai-Wagner-03



Run only once at production time (no leakage!)

Leakage resilient circuits

Formalization of masking by Ishai-Sahai-Wa… 3



Output: Description of circuit **C’** with key **K’**

Correctness: **C[K]** and **C’[K’]** have same functionality

Additionally: **C’[K’]** **leakage resilient** for many executions

Security: adversary learns nothing “useful” from leakage

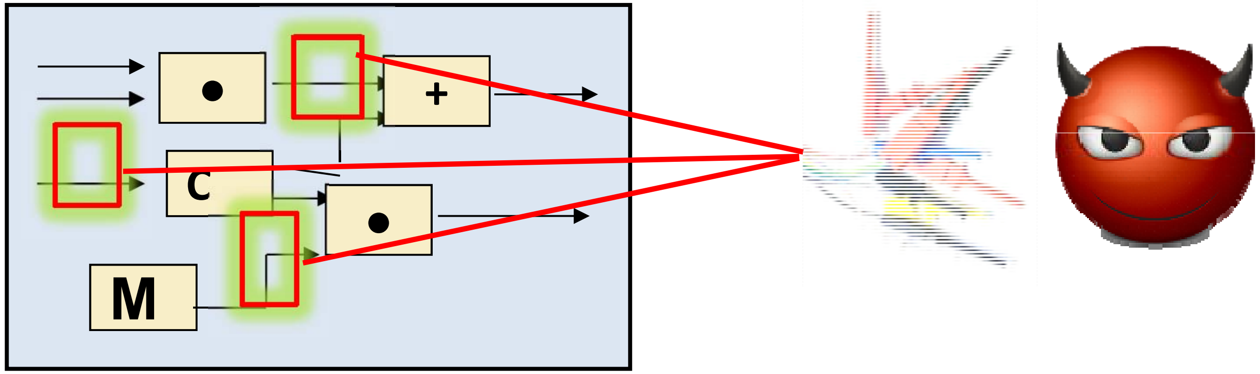
(formalized by simulation-based security)

What leakage do we consider?

n-Probing adversary (ISW03)

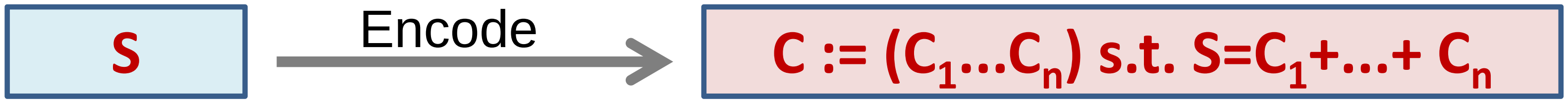
Adversary gets **n** intermediate values of computation

➔ **L** = { values on **n** adversarial chosen wires }



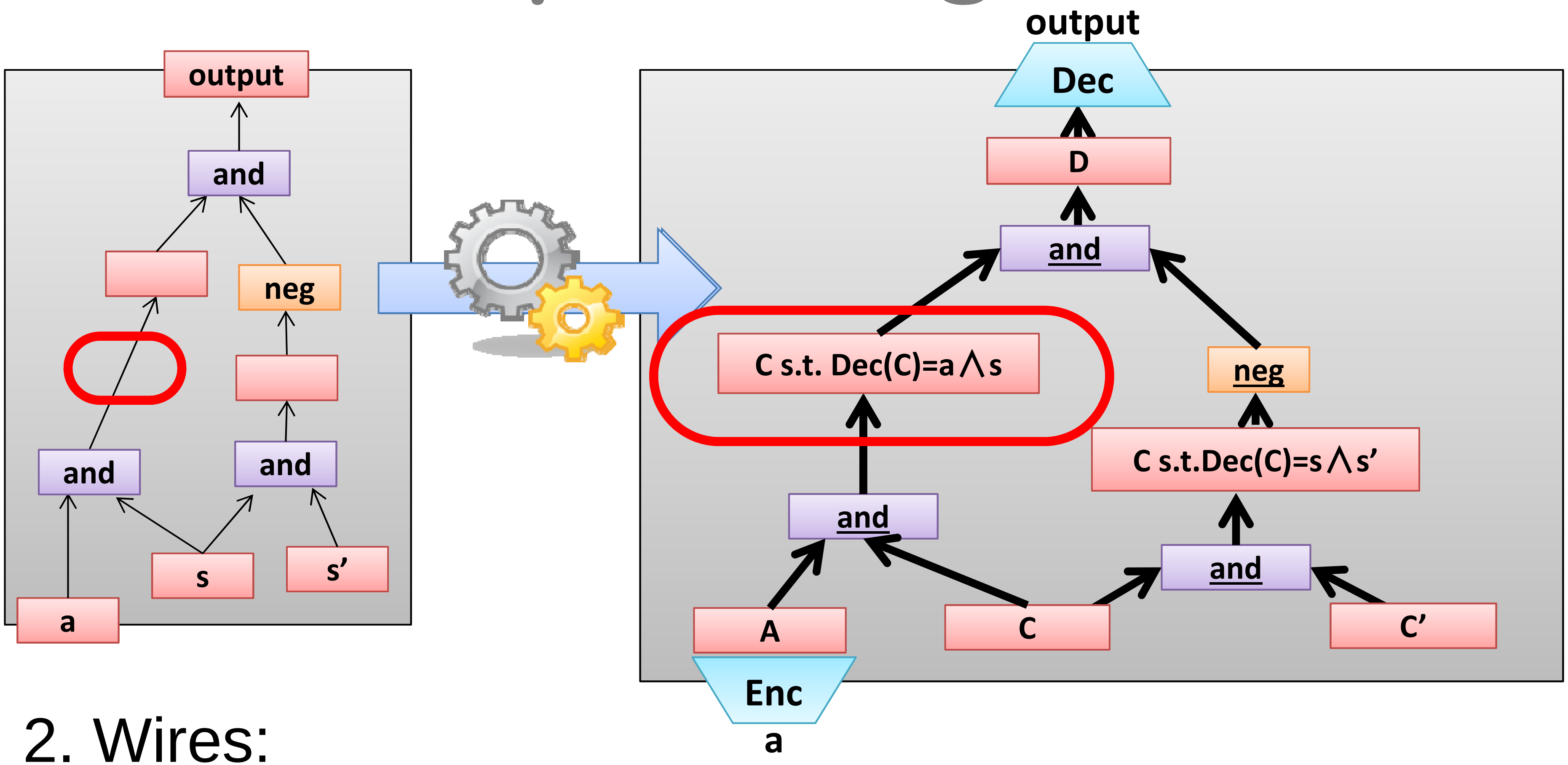
n-probing attack formalization of **n**-variate attacks

Basic ingredient: encoding scheme



How to carry out protected computation?

ISW Compiler: High level



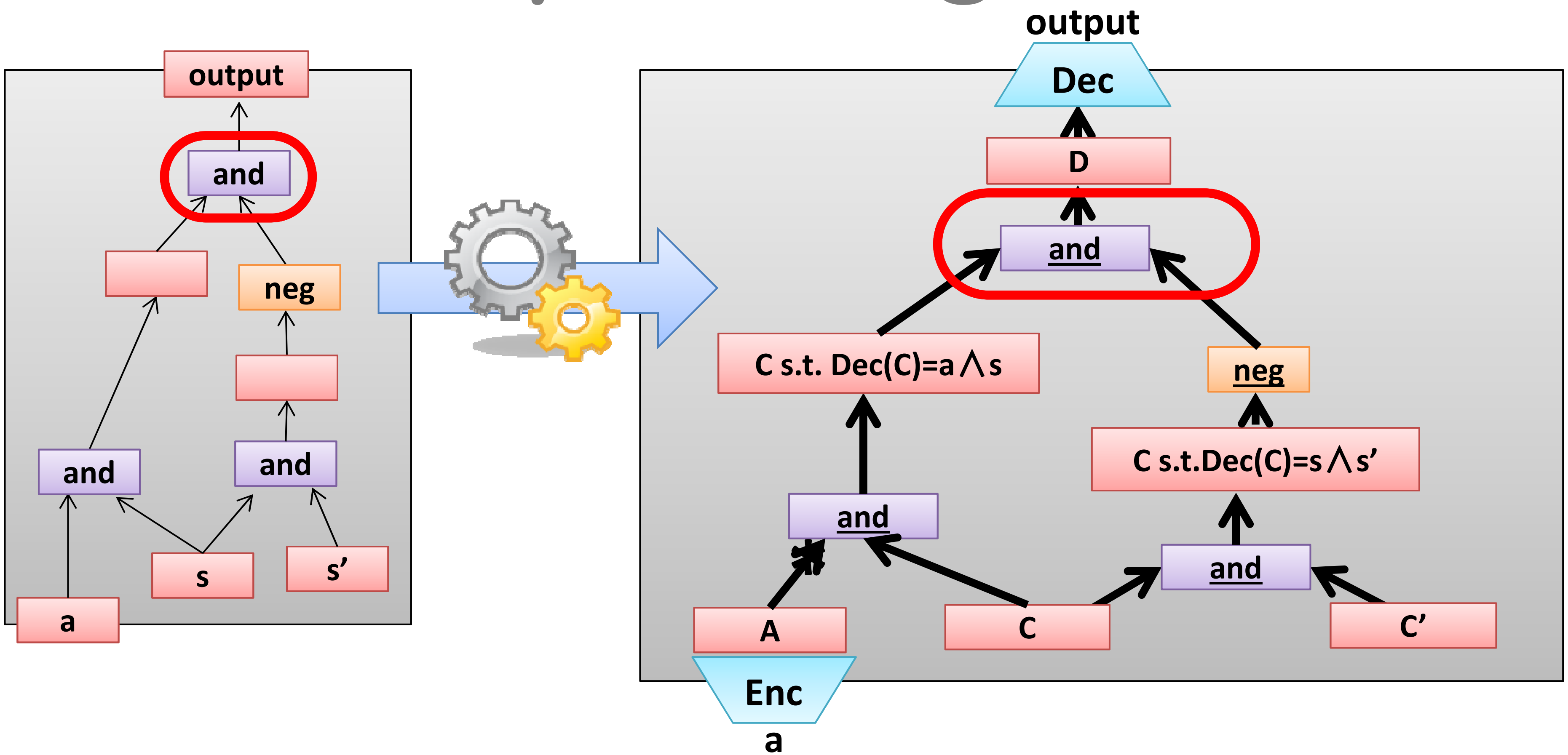
2. Wires:

Each wire
w = a ∧ b

Wire bundle carrying encoding **C** such that **w = Dec(C)**

Main challenge: computing on encoded inputs!

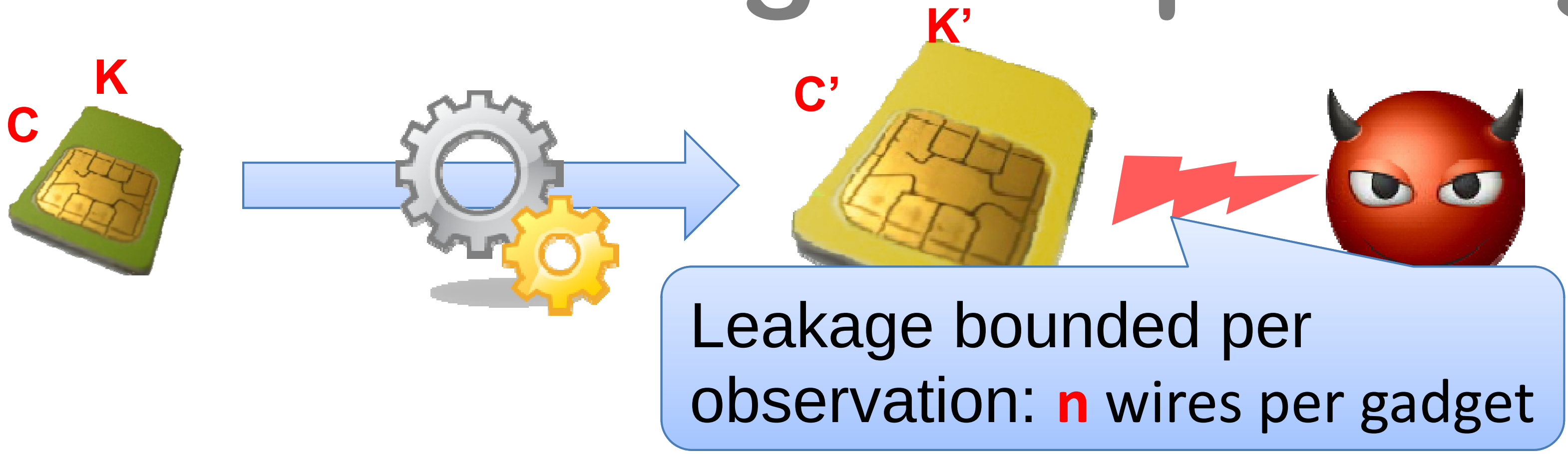
ISW Compiler: High level



<u>3. Gates:</u> → **Gadgets** built from standard gates operating on encodings

Main challenge: algorithm to securely compute AND!

ISW secure against probing



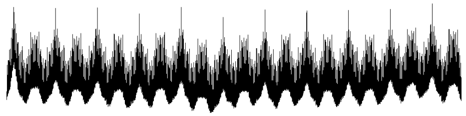
Proof-technique used in many works to prove soundness of masking schemes

Many interesting theoretical extensions:

Low-complexity classes, bounded leakage, …

Models have in common: bounded leakage

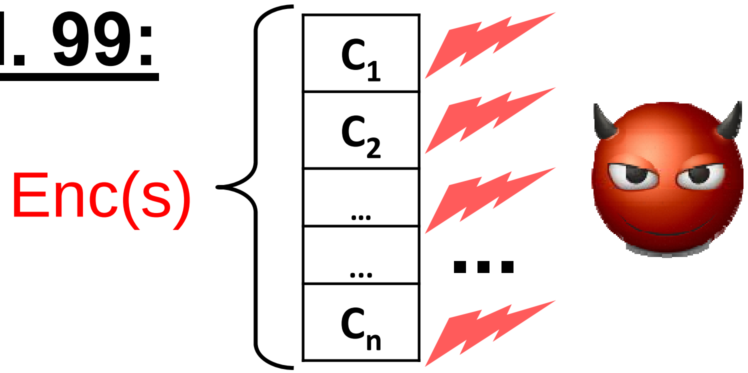
Bounded leakage? *Probably not!*

Measurements require large data 

Not clear how to guarantee bounded leakage

More realistic: no quantitative bound but noisy

Chari et al. 99:



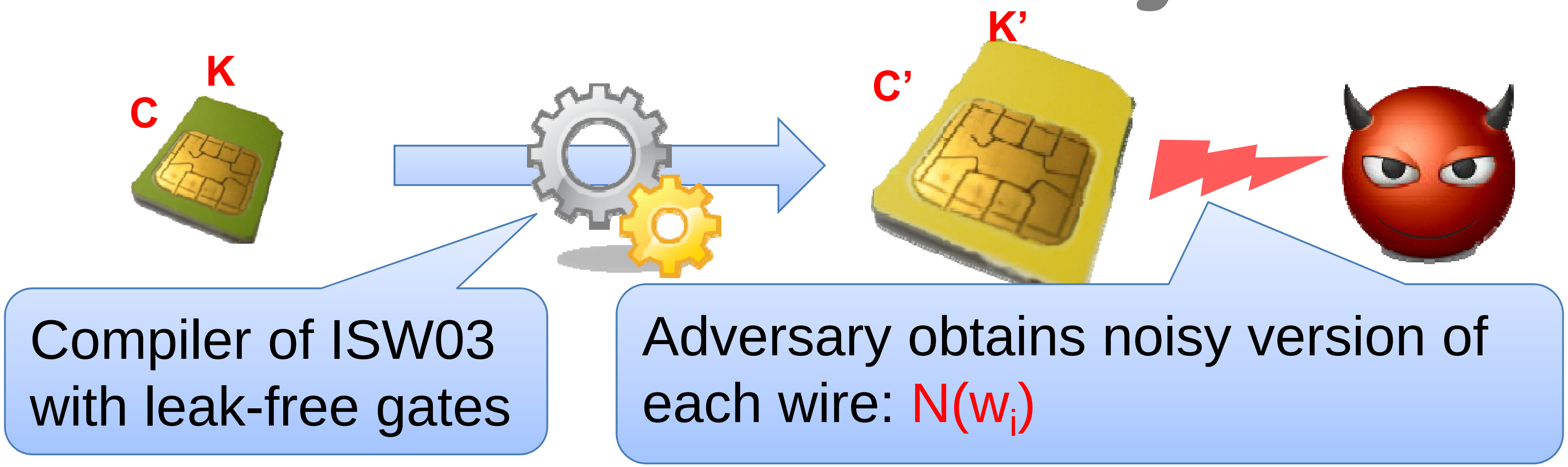
Leakage is c_i + Gaussian noise

(can be generalized)

Long-standing open question: Generlize to computation

Prouff-Rivain, Eurocrypt 13: Prove security of a masked implementation under noisy leakages

PR13: Circuit security



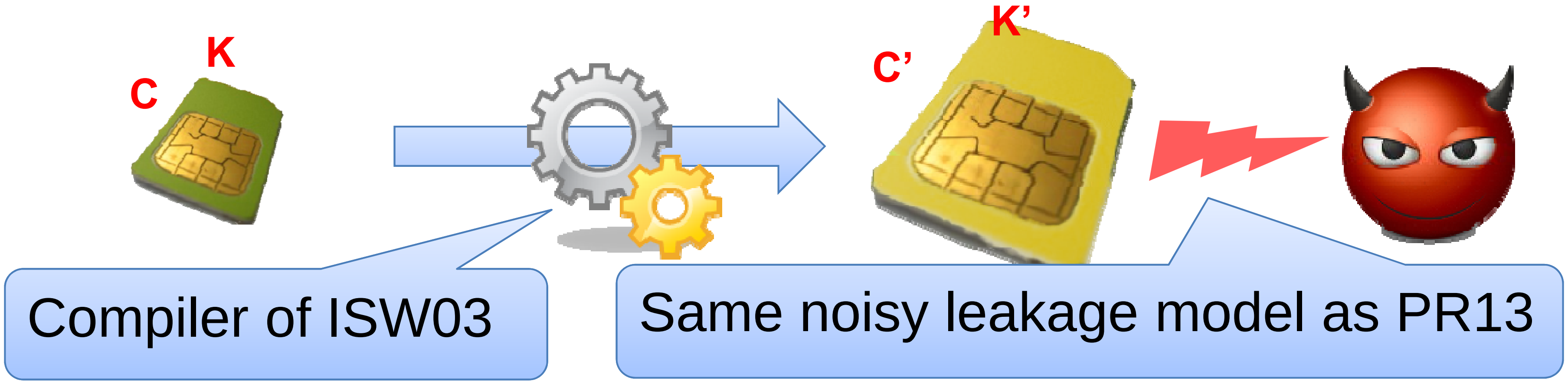
No quantitative bound on amount of leakage

Models physical measurements of power

Drawbacks of the analysis:

- Leak-free gates: no leakage from refreshing
- Security argument only for random-message attack
- Very technical proof

Our Results



ISW03 is secure against noisy leakages

- No leak-free gates
- Full simulation-based security analysis
- Unifying leakage models:

  n-probing security → security against noisy leakage

Useful tool: proofs in n-probing model much simpler than proofs in noisy model

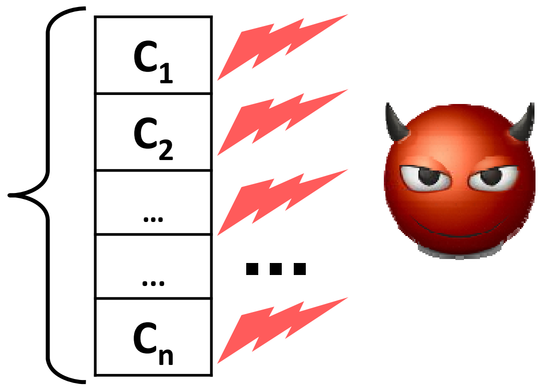
Rest of this talk

1. The noise model in detail

2. Proof outline

Noise model in detail…

Enc(s)



Any **p**-Noisy function **N**

➔ adversary learns **N**(**C_i**)

e.g. **N(C$_i$)**: compute Hamming weight and add Gaussian noise

Prouff-Rivain 13: rather complicated definition

We propose simpler equivalent definition:

uniform **X_0, X_1** → 1. b ← {0,1} 2. $N(X_b)$

$N(X_0)$, X_0, X_1 ≈ **$N(X_1)$, X_0, X_1**

N is **p**-noisy if statistical distance **< p**

Proof outline

1. Simpler noise model: random probing

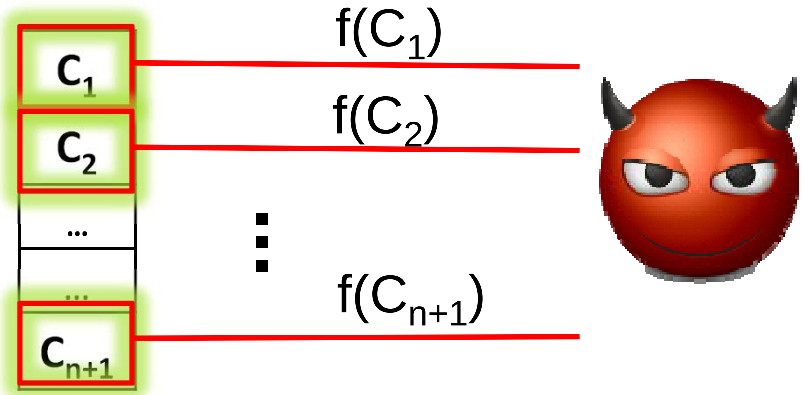
2. ISW03 secure in random probing

3. Random probing = noisy leakages

ISW03 secure against noisy leakages

Proof outline

Step 1: Random probing model (ISW03)



$f(C_i) = C_i$ with prob. q;
otherwise $f(C_i)$ = „?“

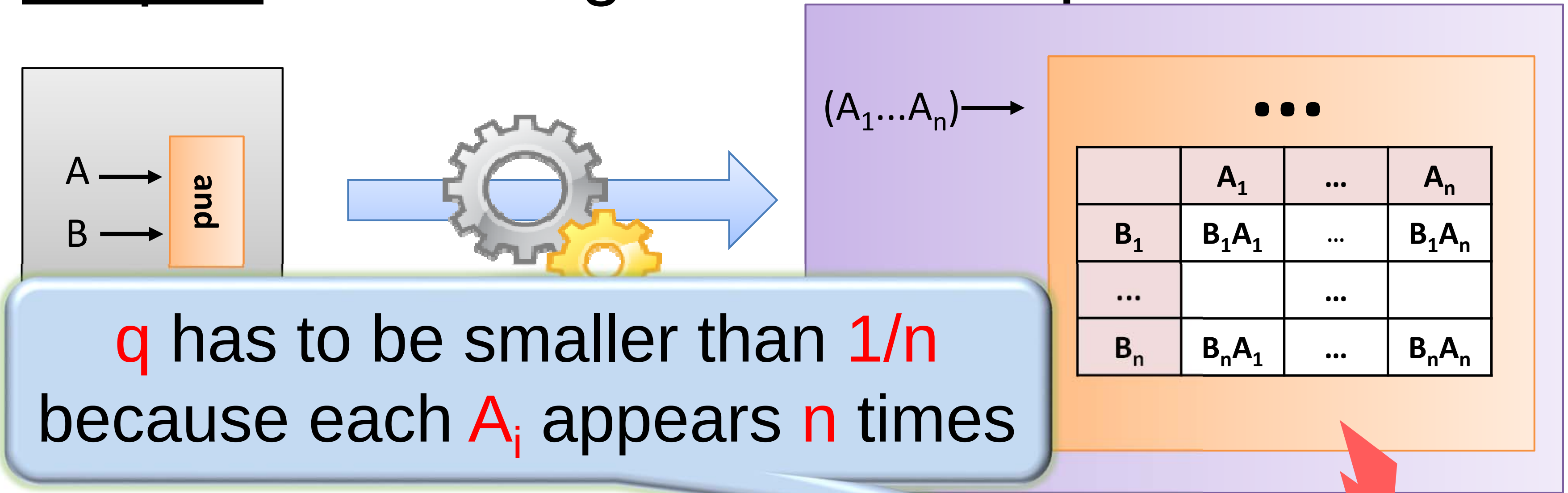
Adv. learns **S** only if „lucky“ in each random probe

➡ Encoding secure in random probing model

How to extend to leakage from computation?

Proof outline

<u>Step 2:</u> Extending to masked operation



	A_1	…	A_n
B_1	B_1A_1	…	B_1A_n
…		…	
B_n	B_nA_1	…	B_nA_n

q has to be smaller than 1/n because each A_i appears n times

learns each value with probability q

➔ at least one share B_i, A_j is not learnt

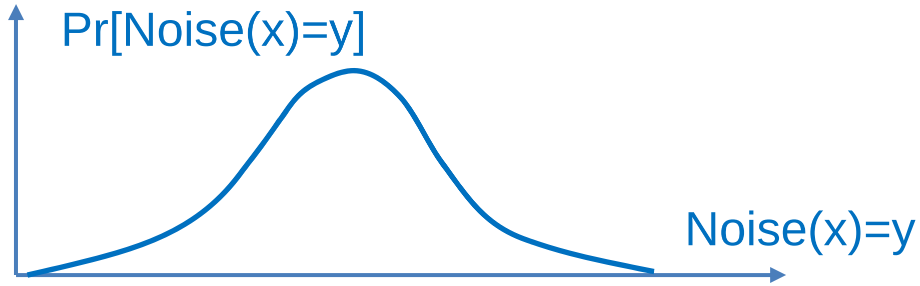
ISW03 is secure in random probing model

How to get to noisy leakage model?

Proof outline

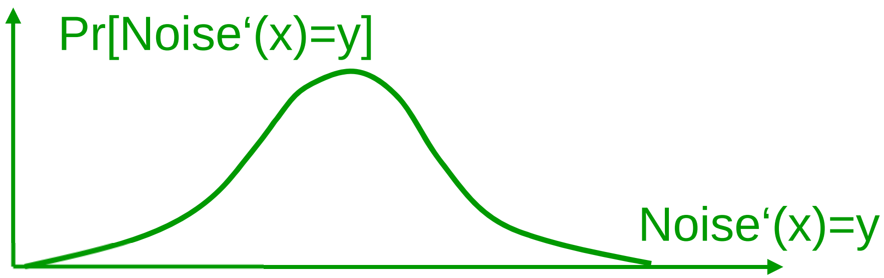
Step 3: Random probing = Noisy leakages

For any p-noisy function Noise(.):



such that for any **x** =

There exists simulated noise distributions Noise‘(f(x)):



P

S

f is **q**-random probing function with **q < p|X|**

⇒ p < 1/n|X|

Conclusion

ISW03 secure in practically motivated model:

- No leak free gates
- Full simulation-based security
- Usefull tool: probing ➔ security against noisy

Main drawback: requires high nois rate

Upcoming work: improve bounds (soon to appear)!

Open problems:

Eliminate independence

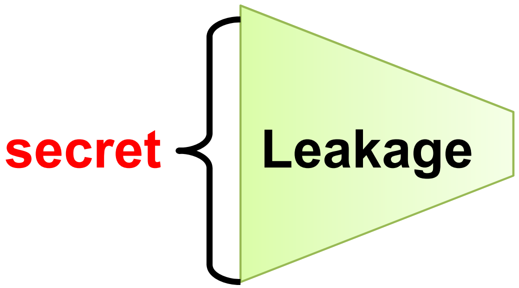
Practical estimation of noise parameter

Thank you!

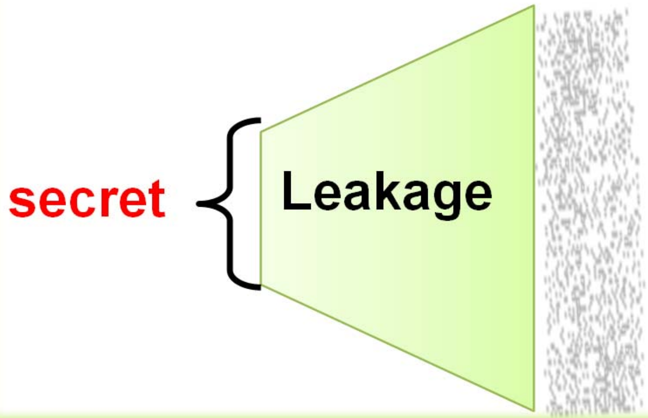
Our result

Analyze common countermeasure in practically motivated model

Bounded leakage

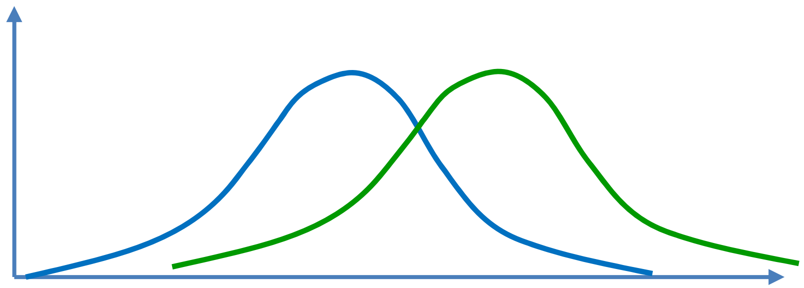


Noisy leakage



Main tool of our work:

Noisy leakage = Bounded leakage



Leakage resilient crypto

Stream ciphers PKE Signatures

Many constructions protocols

IBE ABE Zero-knowledge

Many great ideas! Q: Can I use it to protect my implementation?

Constructions run in PPT but practically inefficient

I want a countermeasure for my AES implementation