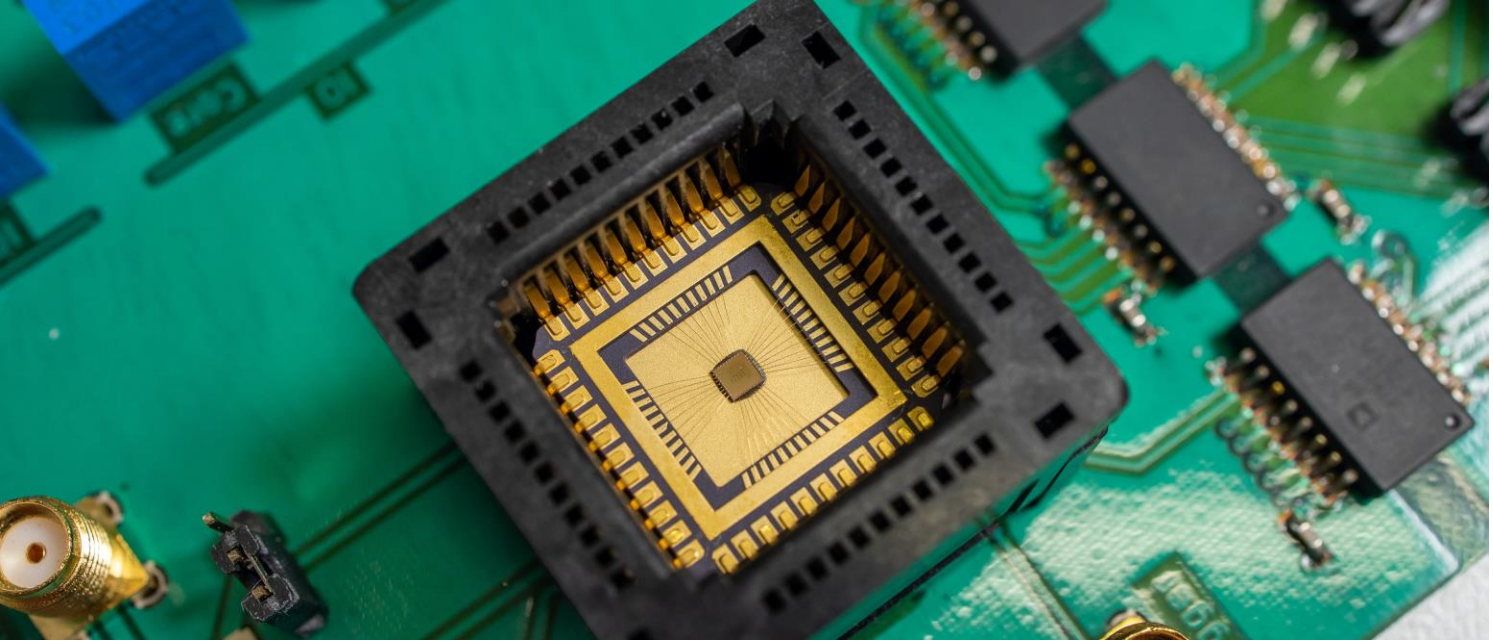


RUB
IMPLEMENTATION SECURITY

RUHR-UNIVERSITÄT BOCHUM

Composable Gadgets with Reused Fresh Masks - First-Order Probing-Secure Hardware Circuits with only 6 Fresh Masks

David Knichel and Amir Moradi

CHES, September 2022, Leuven, Belgium

CASA
Cyber Security in the Age of Large-Scale Adversaries

DFG

CONTRIBUTION

What is it all about?

We've built a hardware gadget

- Enable the composition of any first-order (glitch-extended) robust secure circuit
- No gadget-individual fresh randomness necessary
- Only 6 fresh random masks in total
- Leads to randomness-optimized designs
- **COMAR** (Composable Gadgets with Reused fresh Masks)

STATUS QUO

What do we have?

Masking on an algorithmic Level

- Requires high expertise
- Prone to errors
- Often no formal security proof
- Leads to optimized designs

Gadget-based Masking

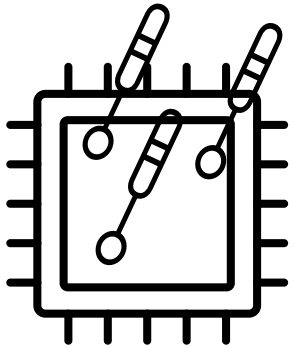
- Based on composability notions
- Can be automated
- Leads to provable secure designs
- Usually introduces higher overhead

ADVERSARY MODEL

What can an SCA adversary observe?

The ISW d-probing model

- Offers high abstraction
- Existing reduction into the *Noisy Leakage Model* (which is close to the real world)
- Extension to HW: robust d-probing Model



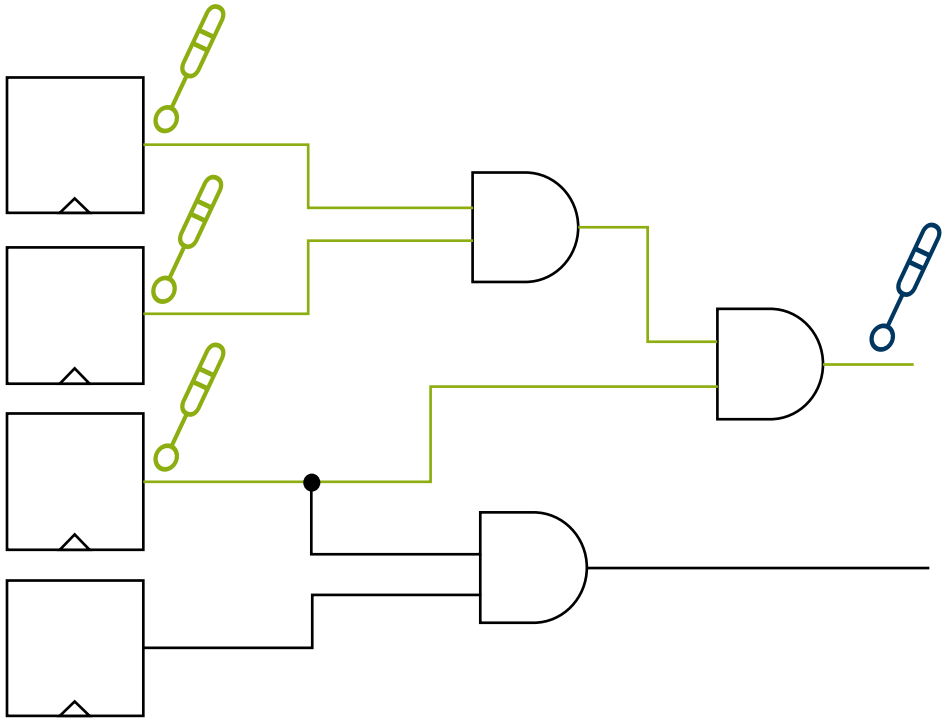
ADVERSARY MODEL

Glitch-extended robust probing model

Standard probe

⇩

Extended probe



PROBLEM

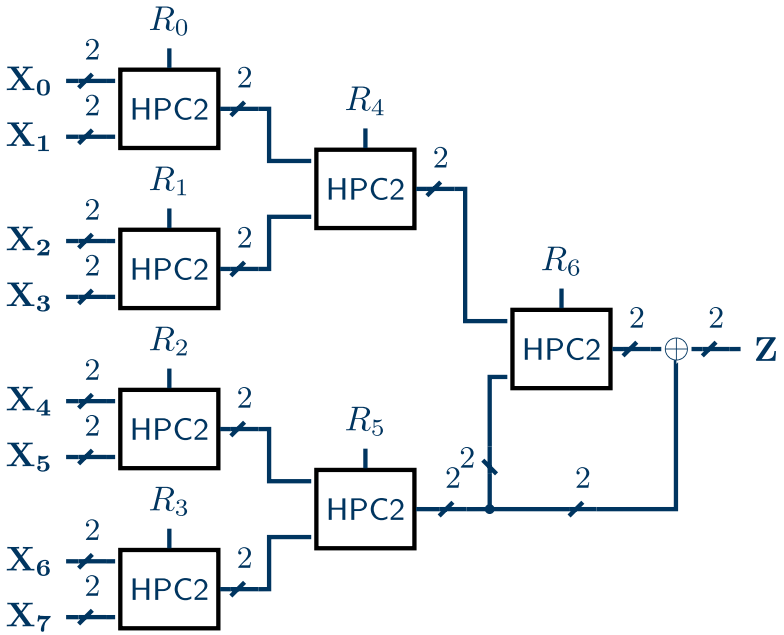
Why is our work important?

Composable gadgets are typically realizing atomic gates

- AND, OR, NAND, NOR, ...

This introduces individual randomness overhead per gadget

- Results in high overall overhead w.r.t. the randomness requirements



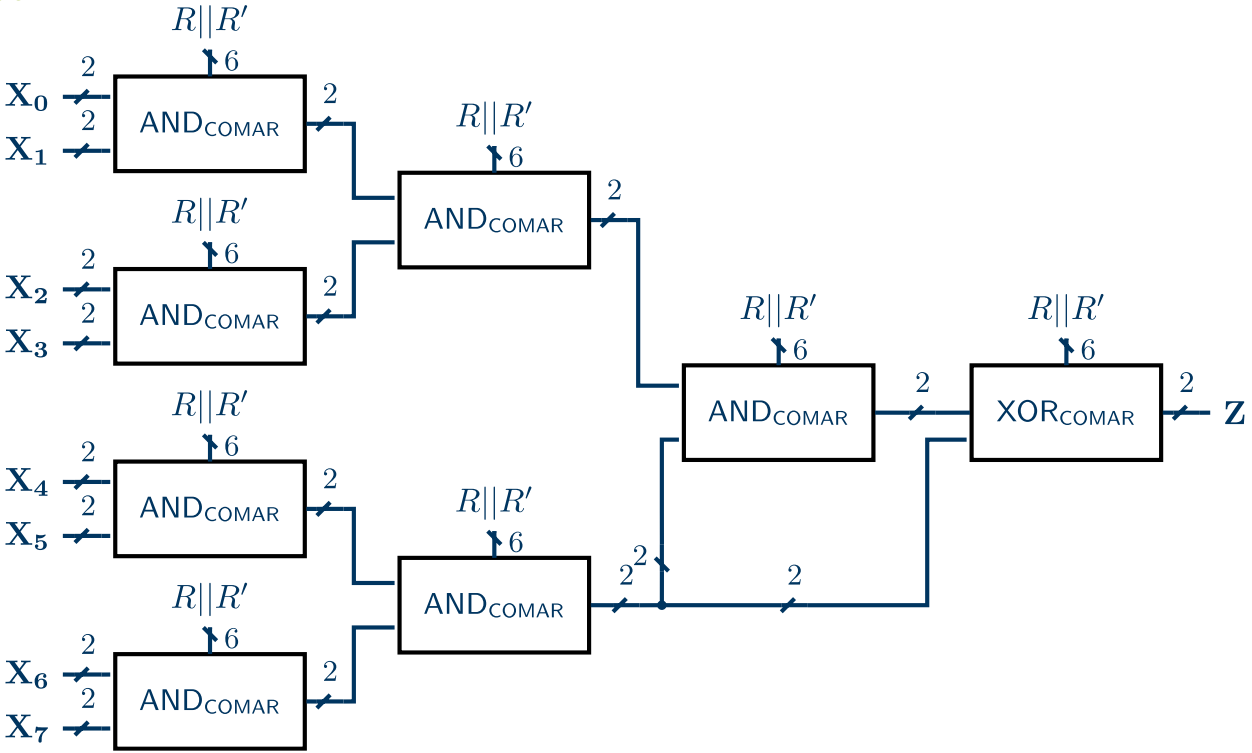
RESEARCH QUESTION

What was the goal?

Can we build gadgets where we can re-use the same randomness in every gadget?

RESEARCH QUESTION

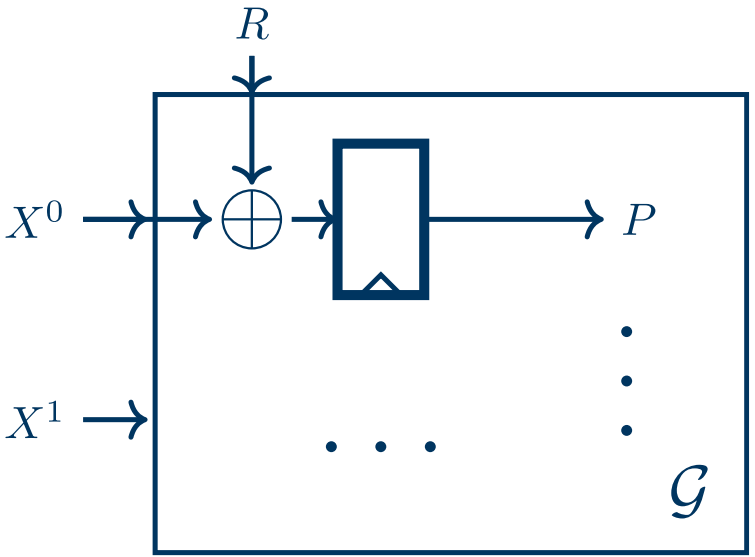
What was the goal?



REQUIREMENTS

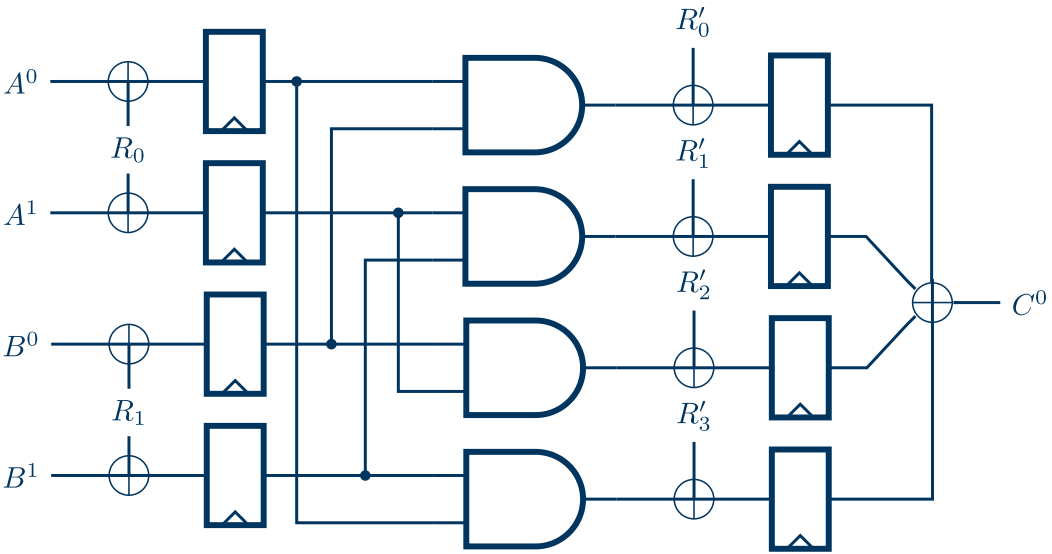
Why is fresh randomness necessary?

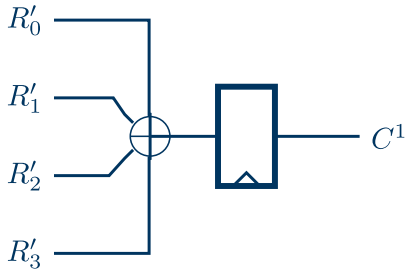
Gadget-individual fresh mask allow to argue about simulatability of probes within a gadget because the input is independent of any other fresh mask.



COMAR-AND

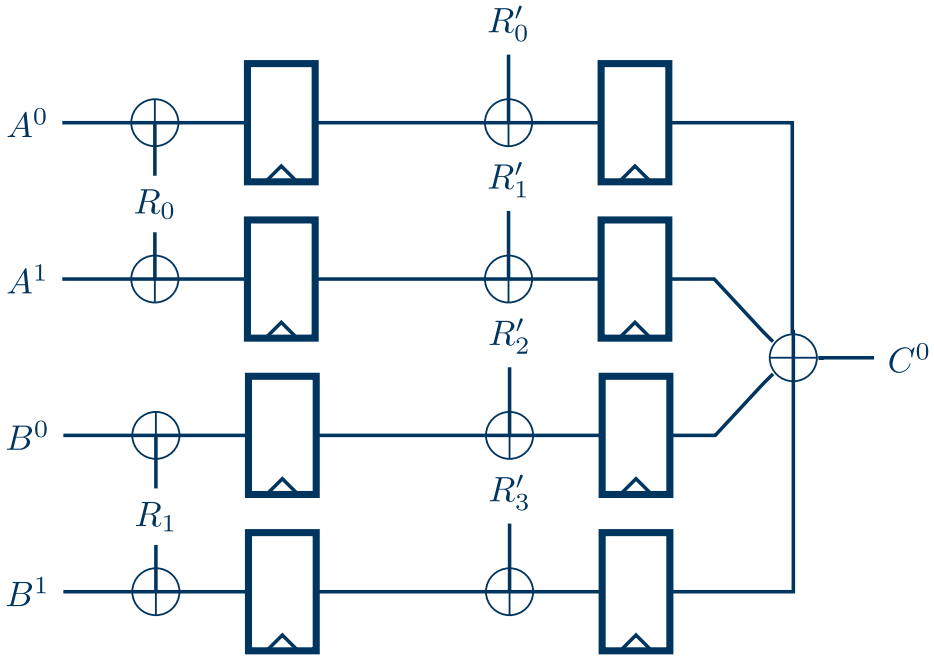
Our AND gadget

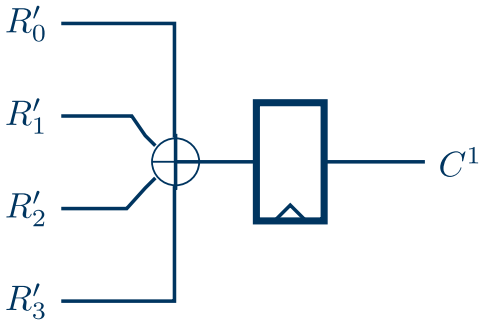




COMAR-XOR

Our XOR gadget





ARBITRARY INPUT WIDTH

We can extend our 2-input gadgets

We also constructed n-input ANDs

- and all other non-linear gates which can easily be derived from AND

$n + 2^n$ fresh masks are needed for the whole circuit.

- Here n is the maximal input width used within the circuit.

DISCUSSION

Is it worth it?

The randomness requirements are drastically improved by our approach

- Breakeven point compared to for example HPC2 is 6 AND gates

Additional latency is the price we pay here

- XOR is not for free w.r.t. latency as for example in the PINI framework

Multiple-input gadgets mitigate this disadvantage

CASE STUDIES

Let's see some results

Scheme	Fresh Random [bit/cycle]	Latency [cycle] added	Latency [cycle] full	Critical Path Delay [ns]	Area [GE]
AES-128					
HPC2	680	8	99	2.04	52 597
GHPC	680	8	99	1.48	67 193
$GHPC_{LL}$	2720	4	55	2.28	52 450
COMAR	6	42	473	1.23	140 214
Skinny64-64					
HPC2	64	4	165	0.55	6 895
GHPC	64	2	99	0.80	22 850
$GHPC_{LL}$	1024	1	66	0.85	18 705
COMAR	6	22	759	0.58	22 090

CASE STUDIES

Considering the randomness source

Scheme	Area [GE]			
	LFSR 31-bit	LFSR 64-bit	Keccak,	Variant
AES-128				
HPC2	247 281	437 205	409 195,	[800]
GHPC	261 673	451 393	423 791,	[800]
GHPC_{LL}	830 370	1 589 250	1 557 778,	[1600]×2
COMAR	141 932	143 608	144 534,	[25]
Skinny64-64				
HPC2	25 218	43 093	82 009,	[200]
GHPC	41 154	59 010	97 964,	[200]
GHPC_{LL}	311 569	597 265	771 369,	[1600]
COMAR	23 808	25 483	26 410,	[25]

Thanks!

Any Questions?

david.knichel@rub.de