Hardware Implementations of Romulus

Mustafa Khairallah and Shivam Bhasin

Nanyang Technological University

Outline

- Part 1: Misuse-Resistant Implementations
- Part 2: Masked Implementations

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Misuse-Resistant Implementations

Motivation

Nonce-misuse and release of unverified plaintexts are threats to lightweight implementations.

Romulus-M addresses both these issues (MRAE, RUP security).

Its implementations is not well-studied.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Is nonce-misuse a real issue?

- Commercial Samsung S series have been shown vulnerable to IV repetition.

Trust Dies in Darkness:
Shedding Light on Samsung's TrustZone Keymaster Design

Alon Shakevsky
shakevsky@mail.tau.ac.il

Eyal Ronen
eyal.ronen@cs.tau.ac.il

Avishai Wool
yash@eng.tau.ac.il

Tel-Aviv University

Implementation Goal

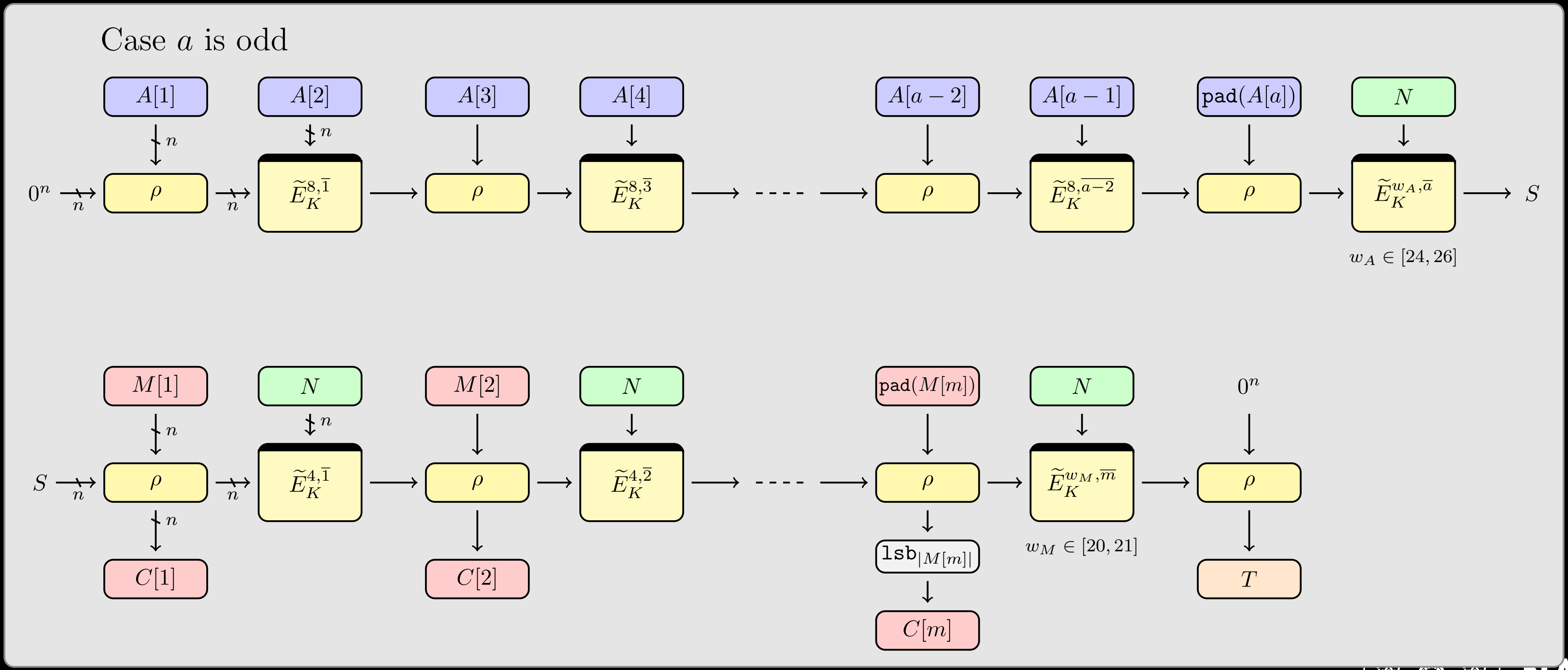
Have both Romulus-N and Romulus-M in the same implementation almost for free.

The user can switch between the two modes during runtime.

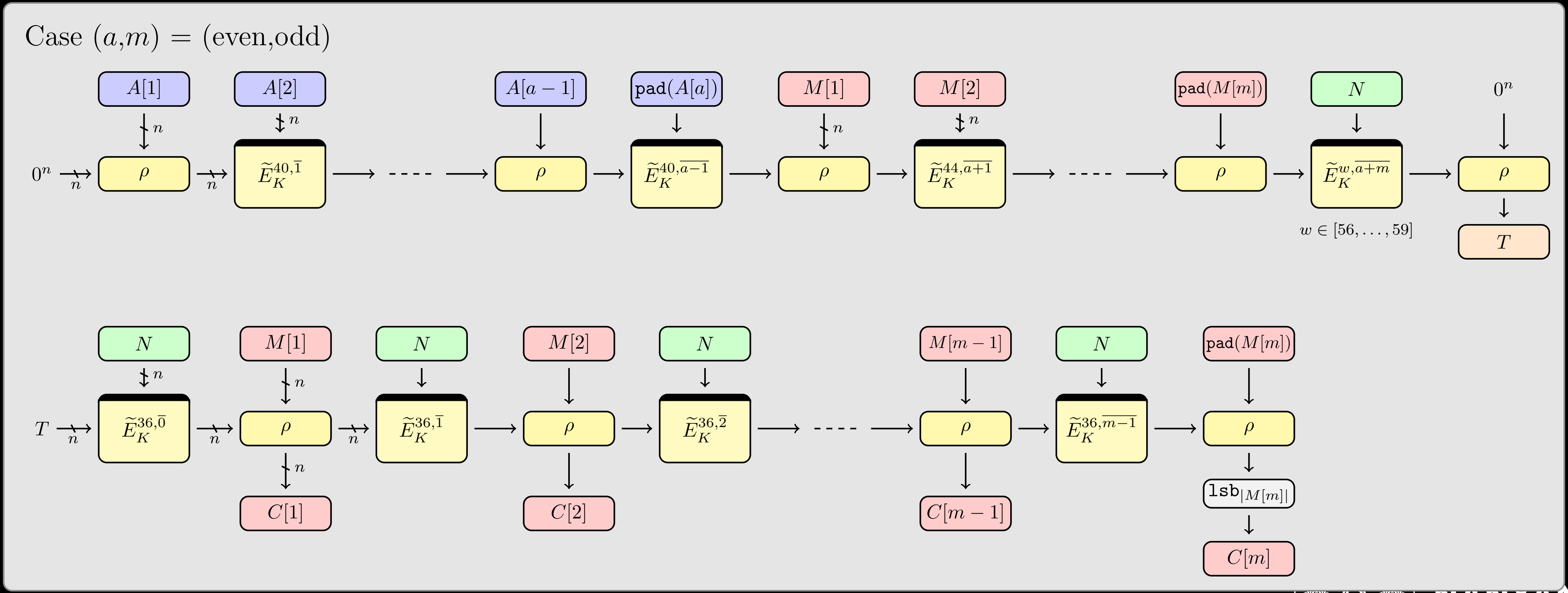
Design the implementation to be compatible with any 128-384 TBC with the proper interface.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Romulus-N


Case a is odd
$A[1]$
$A[2]$
$A[3]$
$A[4]$
$A[a-2]$
$A[a-1]$
$\mathtt{pad}(A[a])$
N
0^n
n
ρ
$\widetilde{E}_K^{8,\overline{1}}$
$\widetilde{E}_K^{8,\overline{3}}$
$\widetilde{E}_K^{8,\overline{a-2}}$
$\widetilde{E}_K^{w_A,\overline{a}}$
S
$w_A \in [24, 26]$
$M[1]$
N
$M[2]$
N
$\mathtt{pad}(M[m])$
N
0^n
S
$\widetilde{E}_K^{4,\overline{1}}$
$\widetilde{E}_K^{4,\overline{2}}$
$\widetilde{E}_K^{w_M,\overline{m}}$
$w_M \in [20, 21]$
$\mathtt{lsb}_{|M[m]|}$
$C[1]$
$C[2]$
$C[m]$
T

Romulus-M

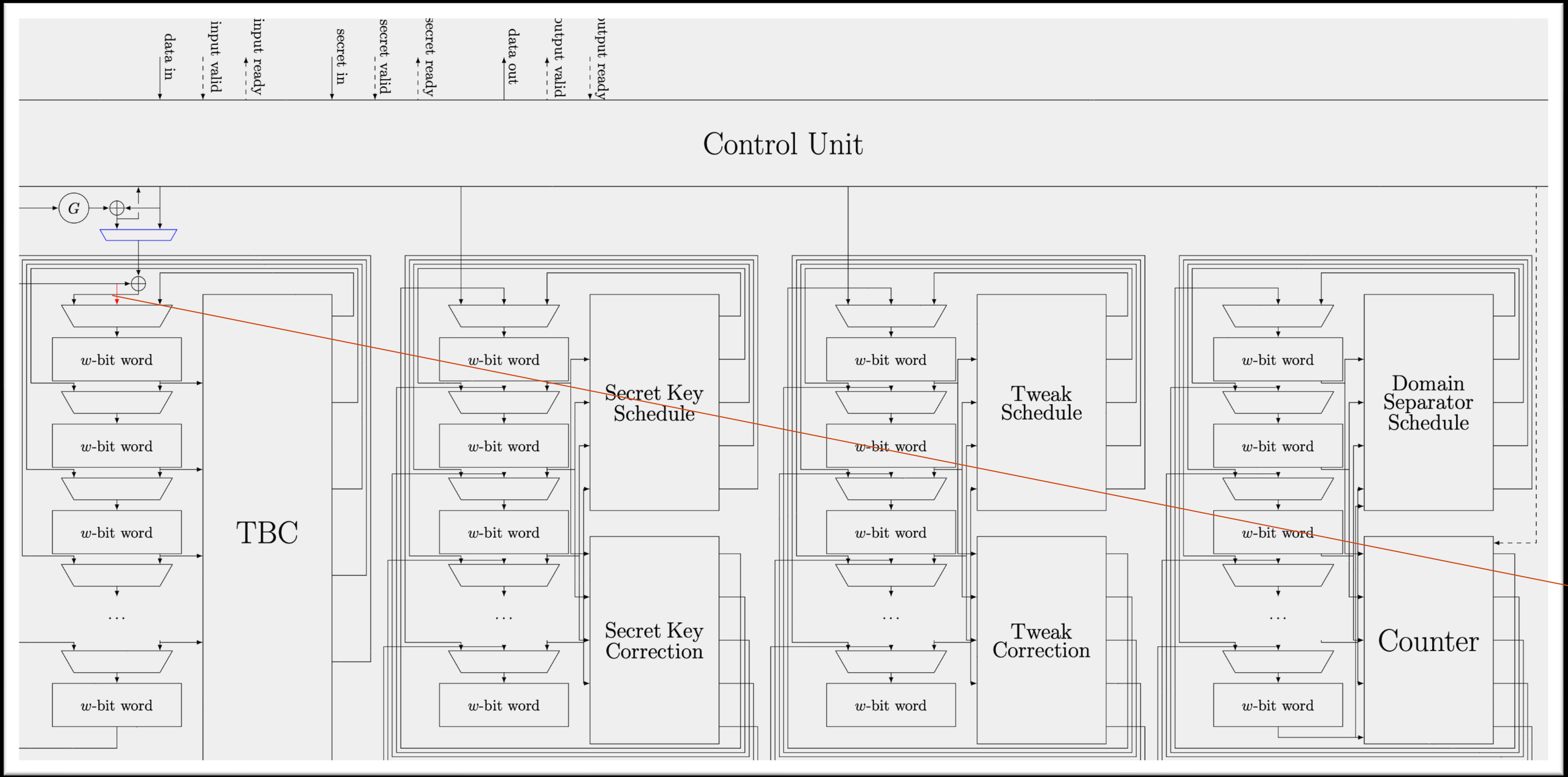

Case (a,m) = (even,odd)
A[1]
A[2]
A[a − 1]
pad(A[a])
M[1]
M[2]
pad(M[m])
N
0^n
ρ
w ∈ [56, . . . , 59]
T
N
M[1]
N
M[2]
N
M[m − 1]
N
pad(M[m])
lsb_|M[m]|
C[1]
C[2]
C[m − 1]
C[m]

ISO/IEC 18033-7

- The ISO/IEC 18033-7 is in the late stages of publishing.
- It specifies the two TBC families: SKINNY and Deoxys-BC.
- Both TBCs are compatible with the Romulus modes.
- We provide implementations for Romulus-N/M (based on Skinny) but also for Romulus-N/M using Deoxys.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Architecture



The architecture is based on in-place computations, where strictly no extra storage is needed except what is needed for the TBC and the FSM.

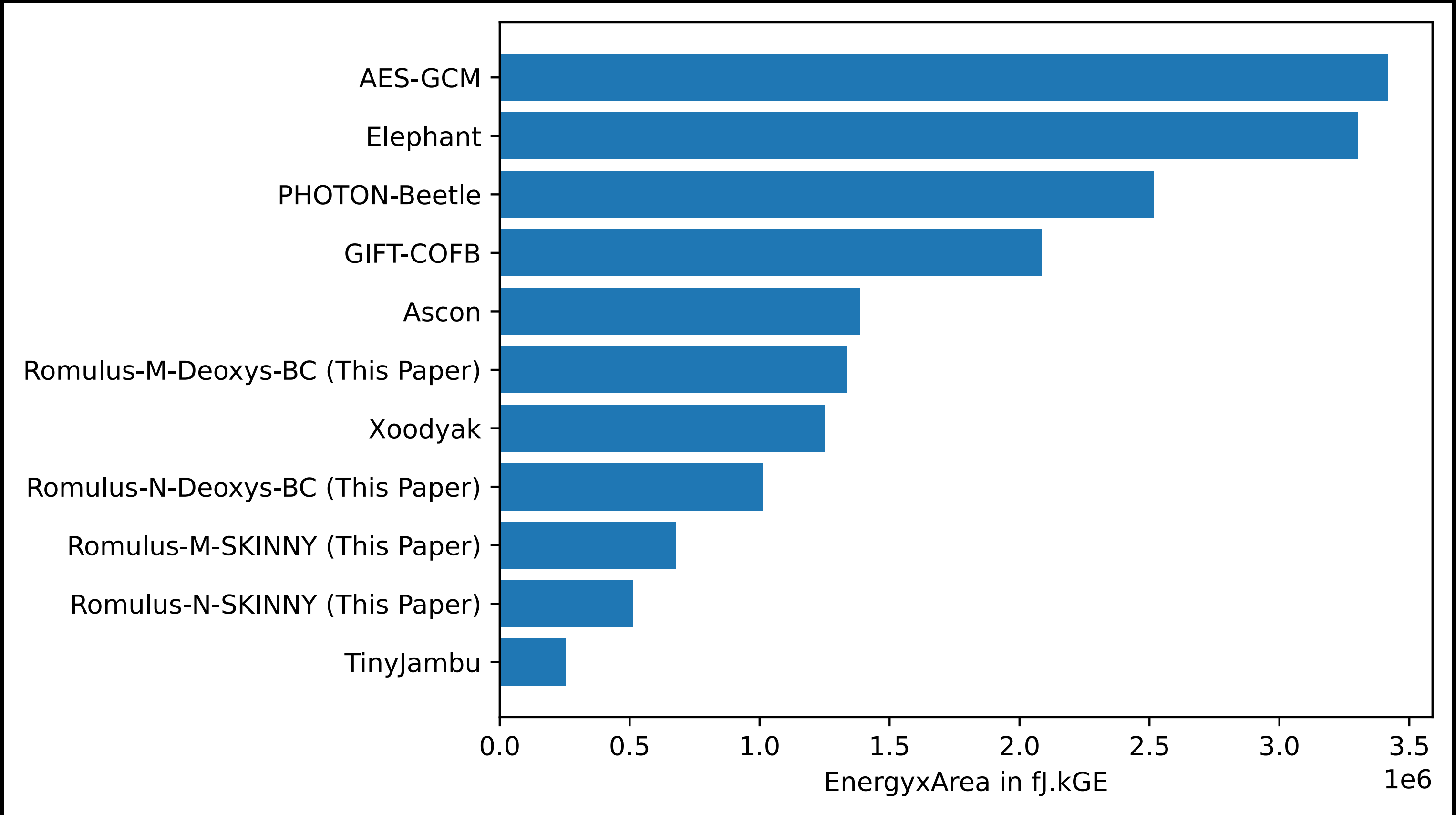
The red arrow (extra mux) is all what is needed to support Romulus_M

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

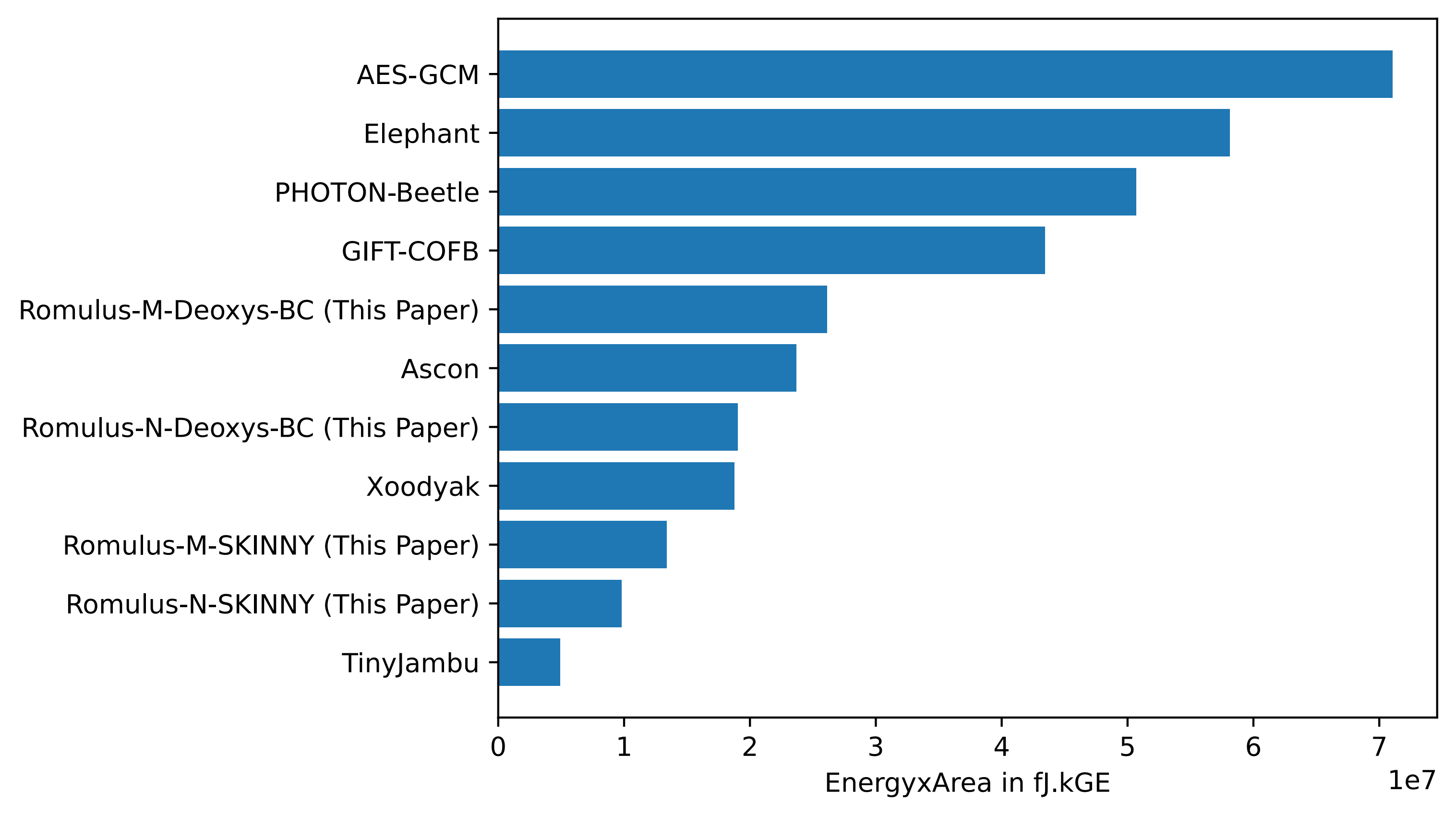
Results

		24MHz						High Speed					
TBC Rounds	Area (GE)	CP (ns)	Power (mW)	Th/put (Mbps) N	Th/put (Mbps) M	Energy (pJ/bit) N	Energy (pJ/bit) M	CP (ns)	Power (mW)	Th/put (Gbps) N	Th/put (Gbps) M	Energy (pJ/bit) N	Energy (pJ/bit) M
SKINNY-128-384+													
1	7348.61	1.11	0.22	89	66	2.47	3.33	1.5	0.74	2.48	1.84	0.30	0.40
2	7865.28	1.16	0.24	159	117	1.49	2.04	1.5	0.74	2.43	3.25	0.17	0.23
4	10124.24	1.9	0.32	263	190	1.21	1.67	2.0	0.70	5.49	3.96	0.13	0.18
5	12035.49	2.41	0.38	302	217	1.26	1.78	2.5	0.73	4.06	2.87	0.14	0.20
10	17767.99	5.03	0.59	431	303	1.36	1.94	5.2	0.73	3.46	2.43	0.21	0.30
40	55098.25	19.25	1.94	633	432	3.06	4.47	20	1.96	1.32	0.90	1.48	2.17
Deoxys-BC-128-384													
1	12659.75	1.91	0.431	163	127	2.47	3.33	1	0.74	6.81	5.31	0.30	0.40

Comparison: EnergyxArea - short messages



Comparison: EnergyxArea - long messages



Ongoing Work

- Security of misusing thiis combined implementation (reusing the key for both Romulus-N and Romulus-M).

- Adopting variable tag length.

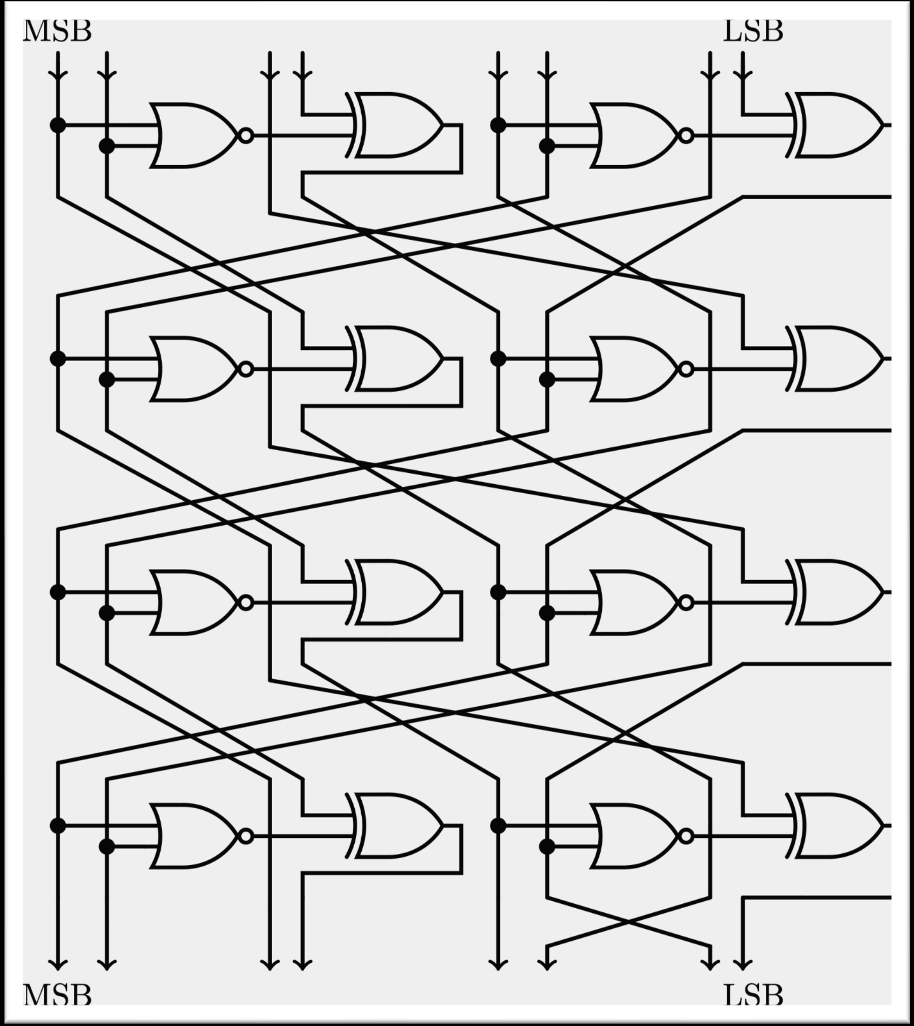
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Design and Test of First-Order Hardware Masked Implementations

Hardware Masking

- A lot of work has been done on designing new masking schemes with provable security and formal verification.
- The security of such schemes in practice is still in question, e.g. DCEM18: “Hardware Masking, Revisited”.

Skinny 8-bit SBox



Basic Gadget

- NOR-XOR.

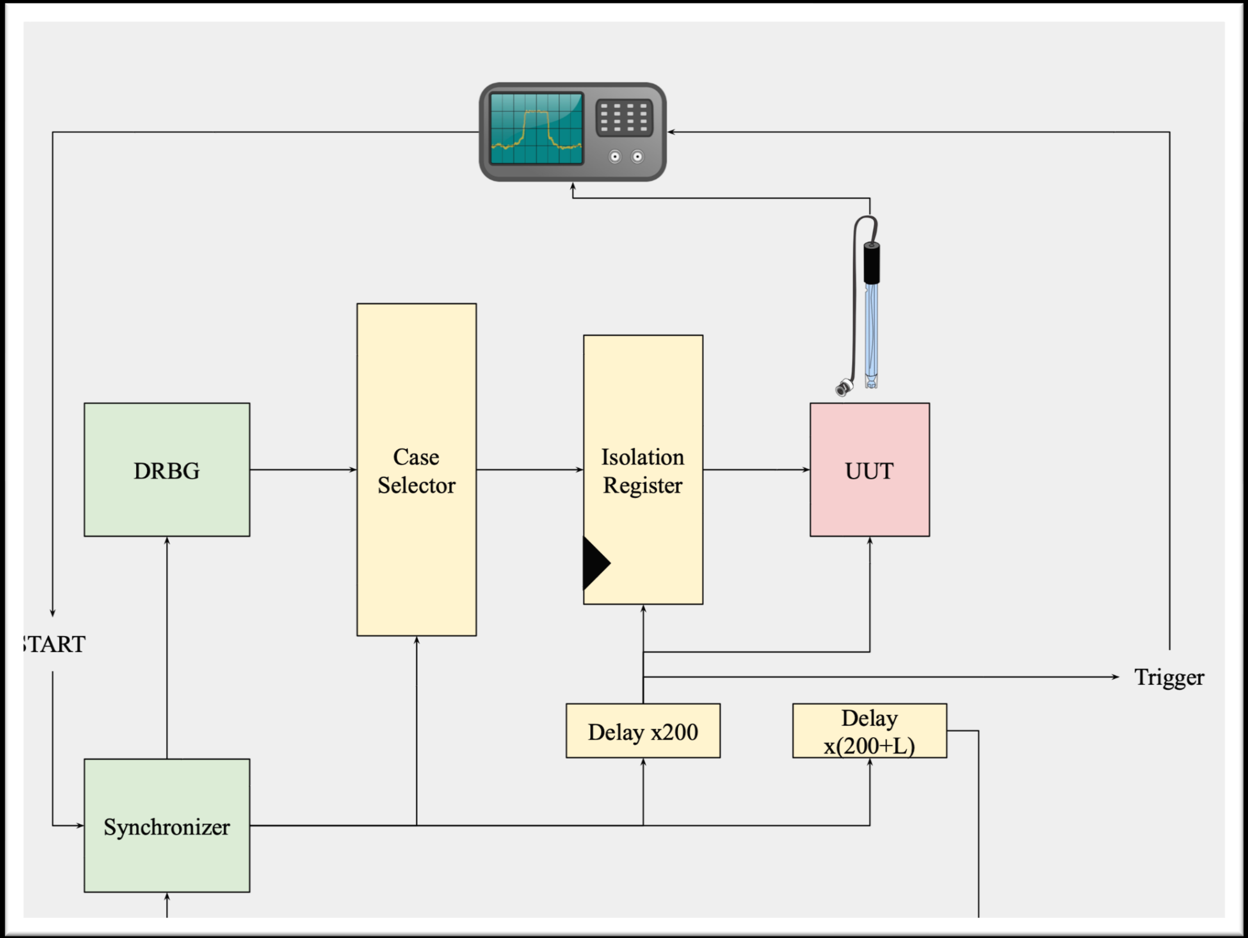
- Can be represented as AND-XOR (with inverted inputs) which is easier to mask by most masking schemes.

Formal verification of different SBox implementations

Scheme	Cycles	Shares	R	r	Probing	NI	SNI	PINI
DOM1	4	2	8	3	y	y	y	-
DOM1-Pipelined	4	2	8	3	+	y	y	-
DOM1-SNI	8	2	8	3	+	+	+	-
DOM1-Rapid	2	2	25	19	?	?	?	?
CMS1	4	2	32	12	+	+	y	-
CMS1-Rapid	2	2	76	56	?	?	?	?
ISW1	8	2	8	3	+	+	+	-
ISW1-PINI	12	2	16	6	+	+	+	+
HPC	12	2	8	3	+	+	+	-
HPC2	12	2	16	6	+	+	y	+
PARA1	8	2	16	6	+	+	+	-
PINI1	4	2	8	3	y	y	y	y
TI33	2	4	0	0	+	-	-	-
DOM1-NC	24	2	8	3	+	+	+	-
AND3-DOM1	1	2	3	3	+	+	y	-
AND4-DOM1	1	2	7	7	+	+	y	-
AND3-CMS1	1	2	8	8	+	+	+	-
AND4-CMS1	1	2	16	16	+	+	+	-

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Practical testing of the Sbox: Sasebo Gii

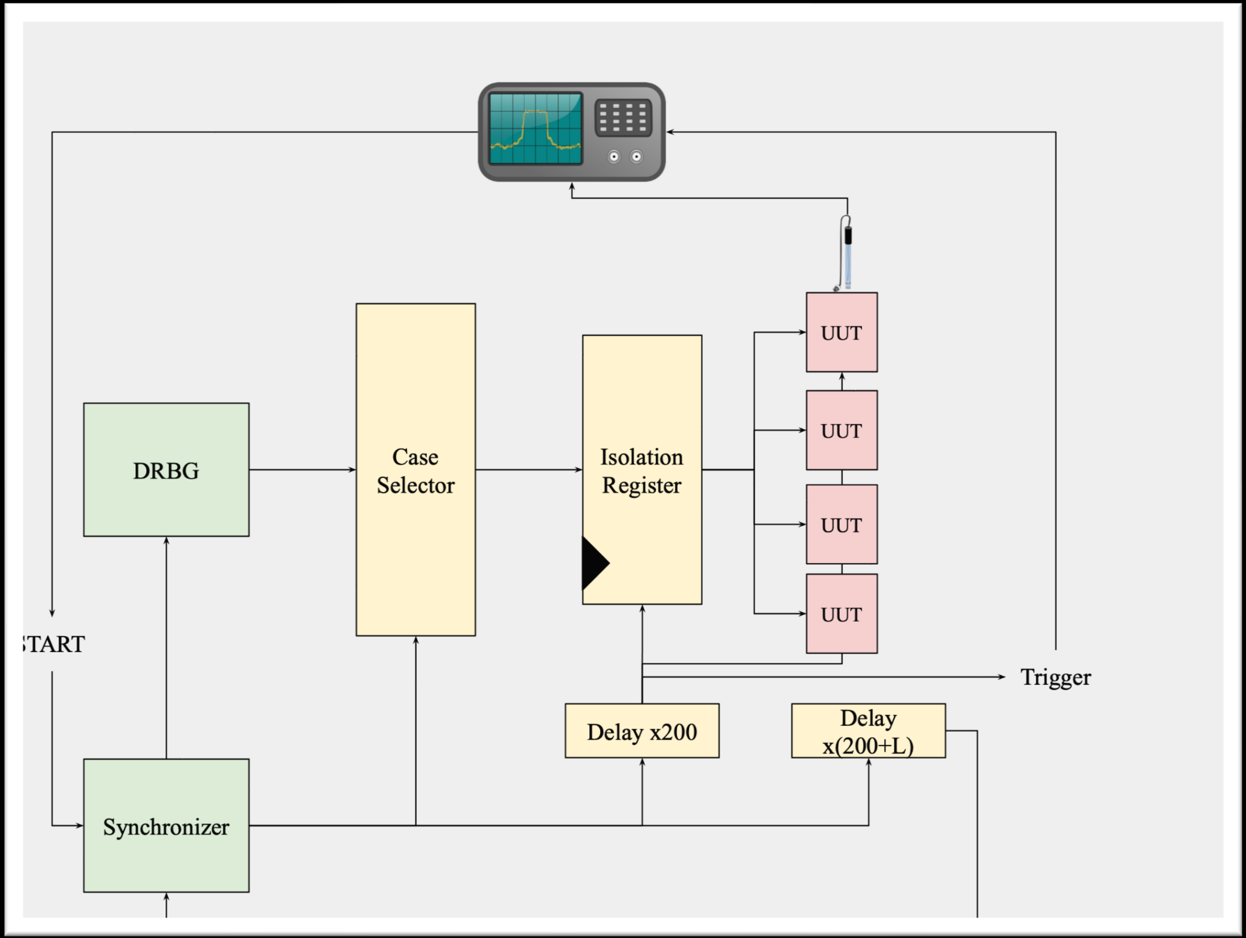


Challenges:

- Trace complexity.
- Trace acquisition speed.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Replication to increase SNR

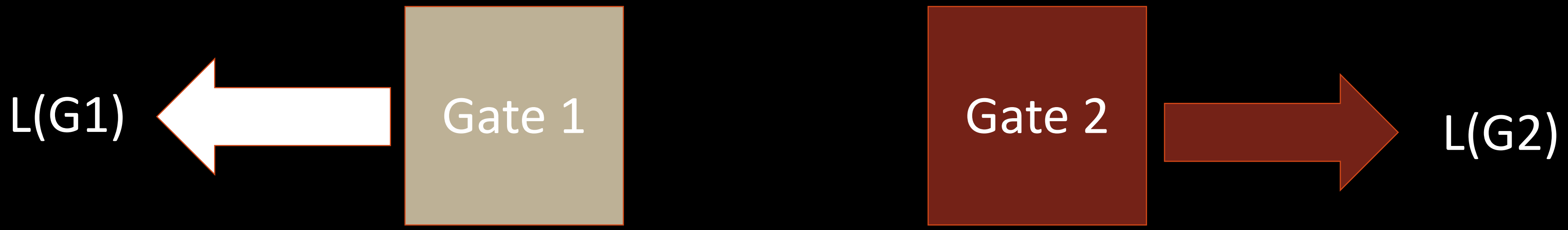


NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Practical Testing Results

Scheme	Replicas	SNR	Traces
Mask Off			
DOM-SNI	1	174.3	823
TI33	1	172.3	1,536
Mask On			
DOM	9	174.5	7,784
DOM-SNI	9	173.3	2,140
TI33	9	804.7	1,393
TI33	1	864.19	62,924
DOM-NC	99	2028.44	5,913,875
DOM-NC	9	1183.08	6,190,000
DOM-NC	1	33.57	>200,000,000

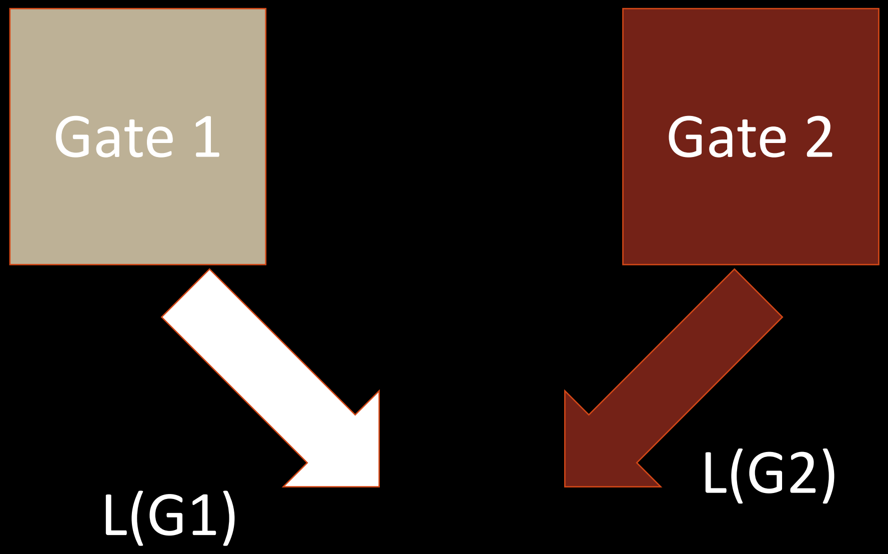
Leakage Assumption



L(G1) and L(G2) are independent

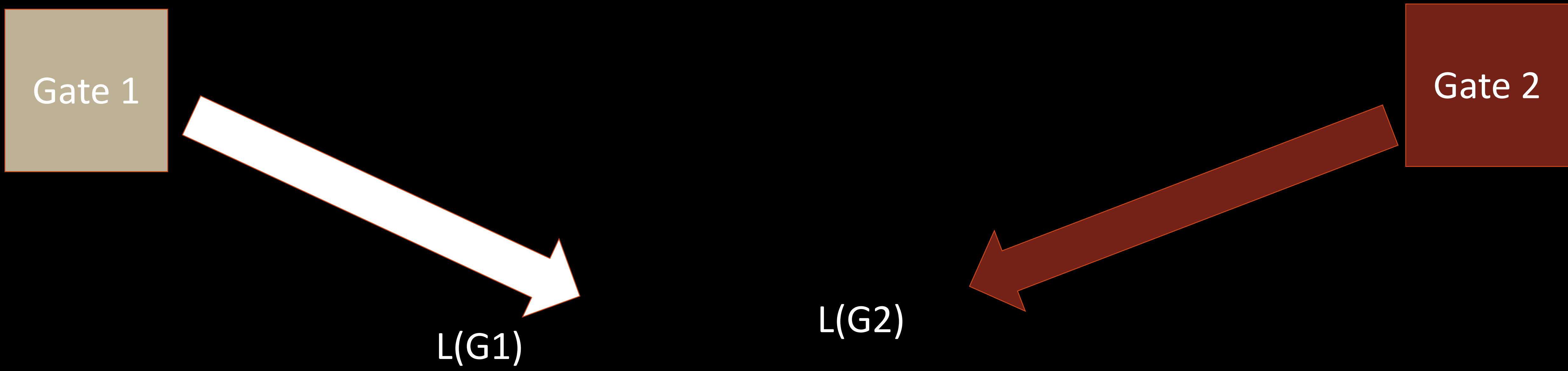
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Real Leakage (De Cnudde *et al.* [DCEM18])



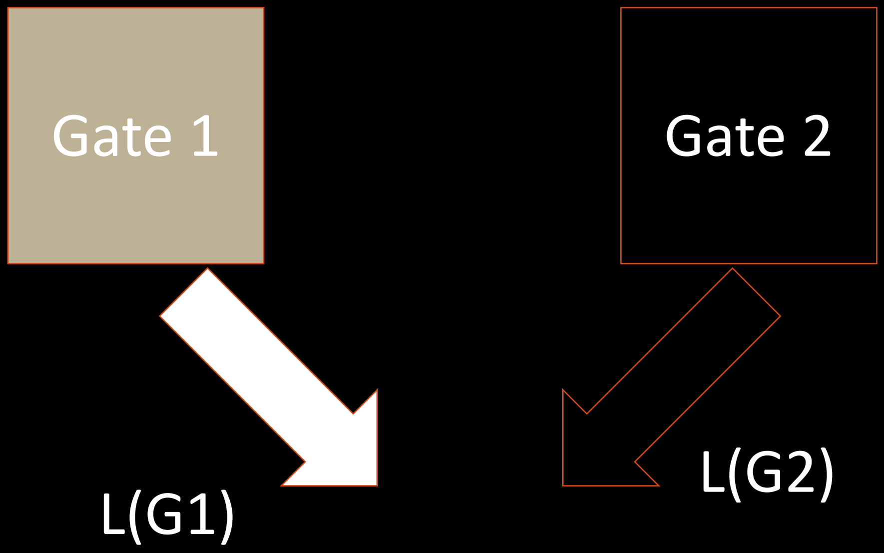
L(G1) and L(G2) are coupled

Solution 1: Spatial Decoupling



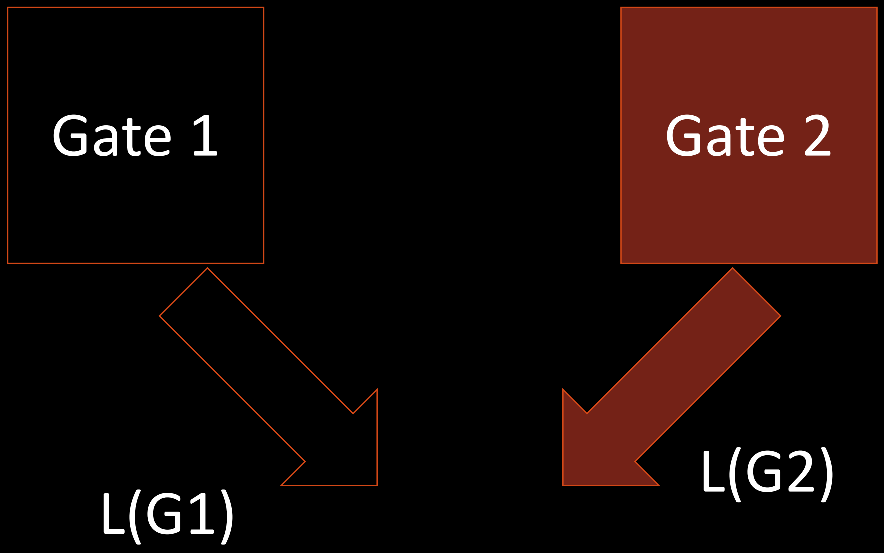
Area is too high, requires intensive circuit expertise, low level, not easy to replicate

Solution 2: Temporal Decoupling



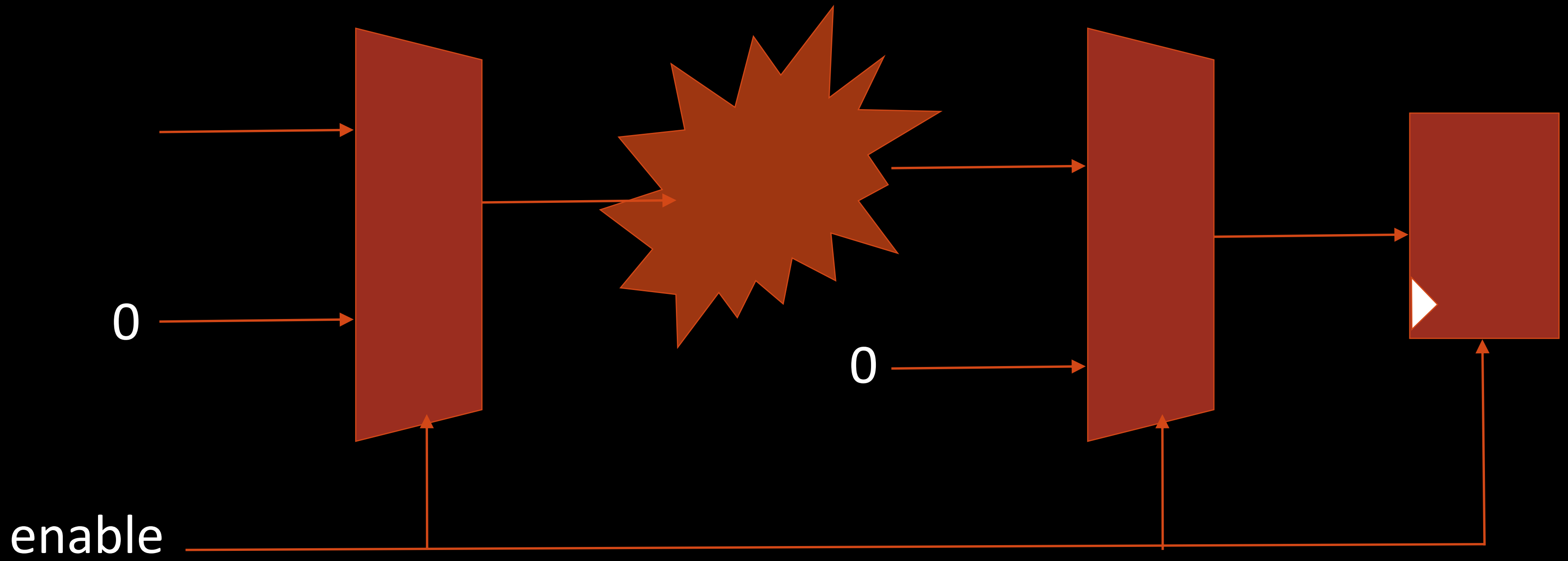
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Solution 2: Temporal Decoupling



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

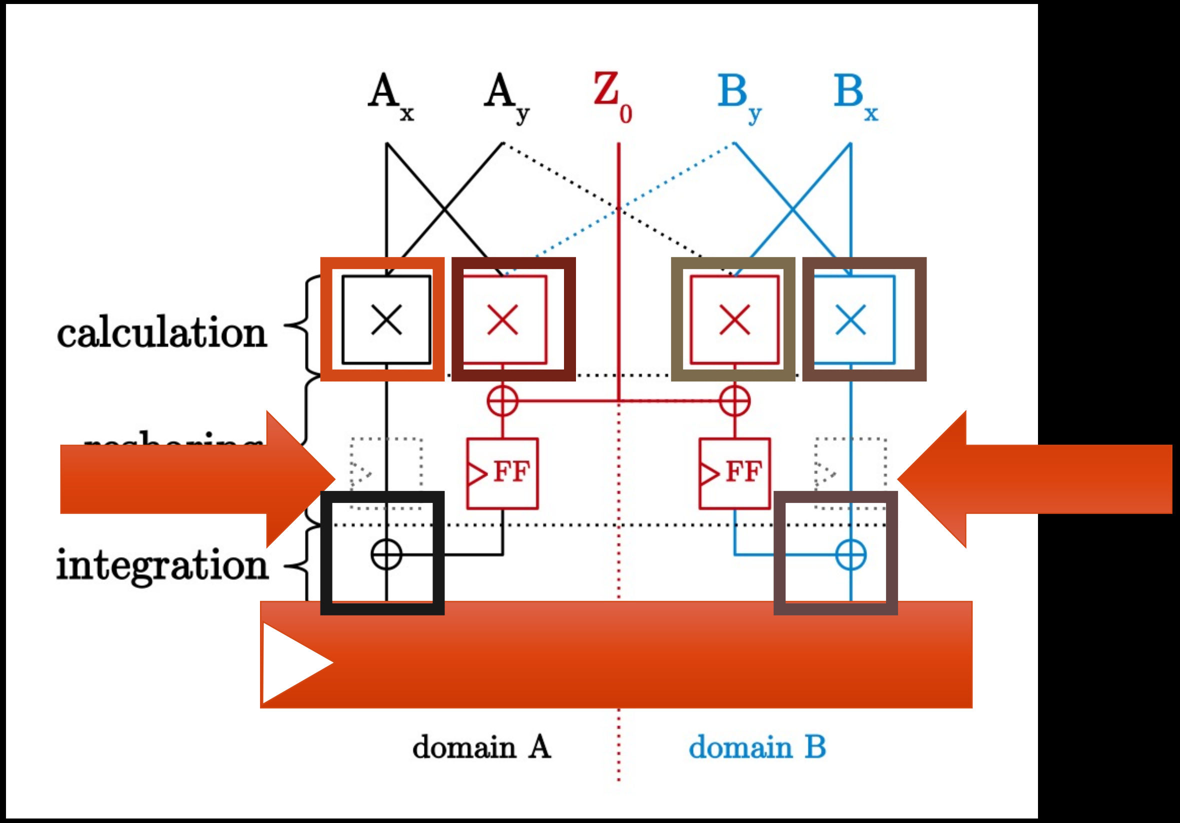
Power Gating



Pros and Cons

- Pros:
  - Easy to do in the RTL.
  - Easy to replicate and argue about.
  - Area is not too high.
- Cons:
  - 6 cycles instead of 2
- What is the price of security?

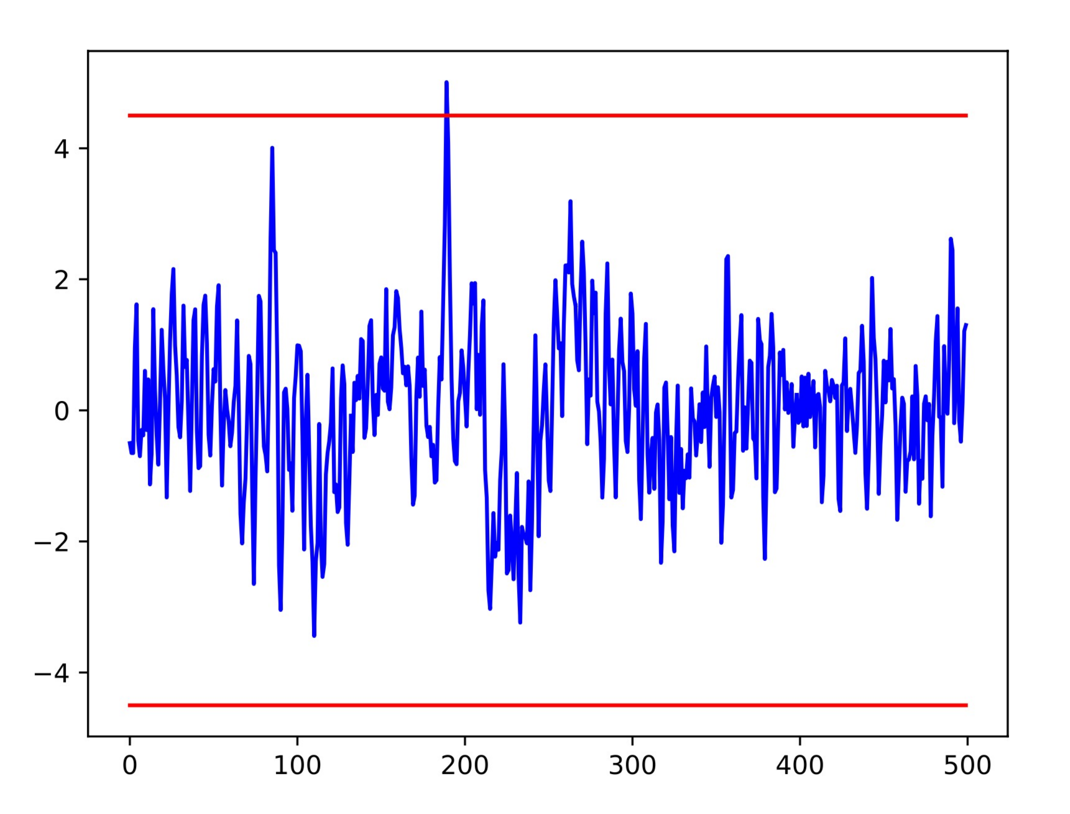
Power-Gated Domain-Oriented Masking



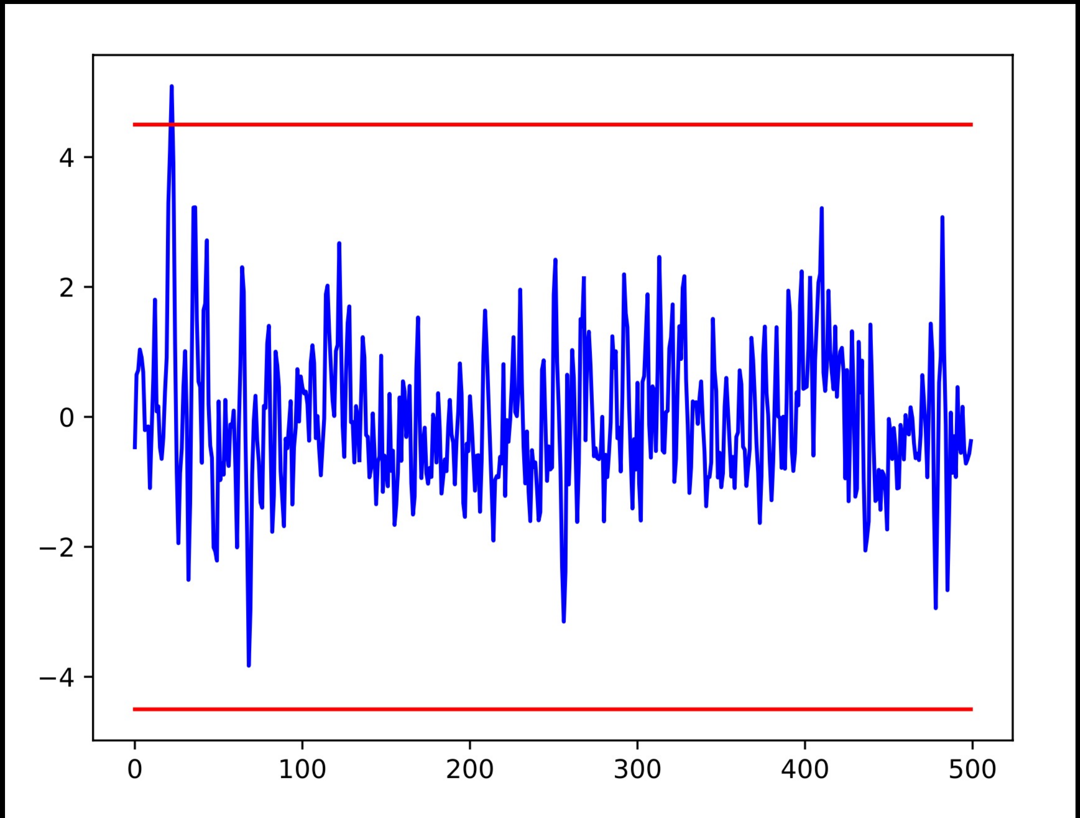
Practical Testing Results

Scheme	Replicas	SNR	Traces
Mask Off			
DOM-SNI	1	174.3	823
TI33	1	172.3	1,536
Mask On			
DOM	9	174.5	7,784
DOM-SNI	9	173.3	2,140
TI33	9	804.7	1,393
TI33	1	864.19	62,924
DOM-NC	99	2028.44	5,913,875
DOM-NC	9	1183.08	6,190,000
DOM-NC	1	33.57	>200,000,000

DOM TVLA Test

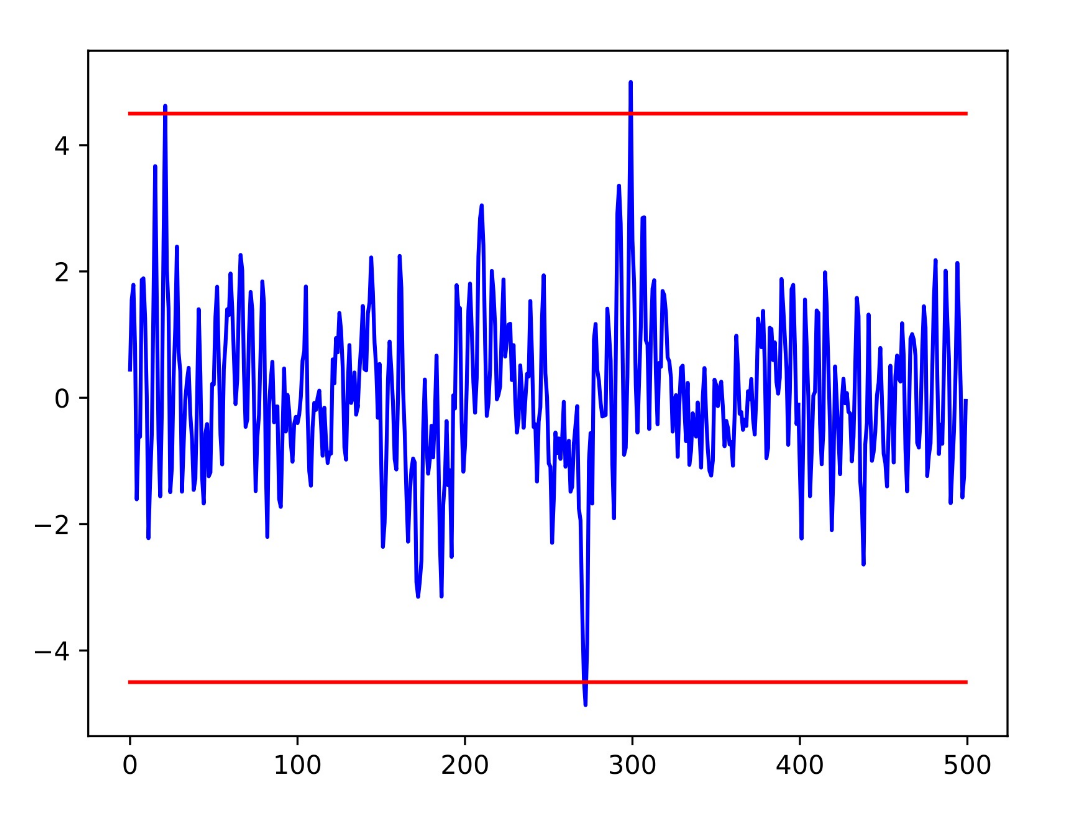


DOM-SNI TVLA Test



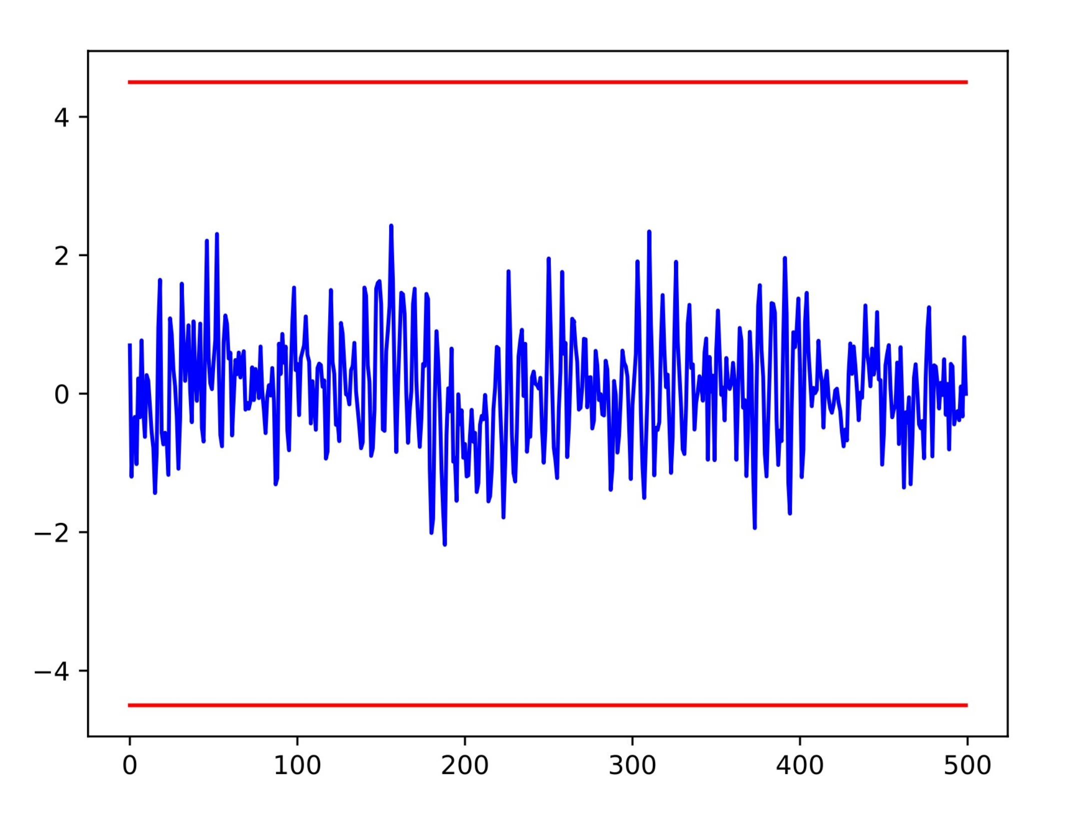
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Amplified Test



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Non-Amplified Test



Overall results: Area in GE

Implementation	Protected Key SE	Protected Key DE	Unprotected Key SE	Unprotected Key DE
DOM1	13395.97	15269.5†	11889.72	13579.51†
DOM1-Pipelined	14619.5	14886.5	13068.47	13276.52
DOM1-Rapid	20634.3	22230.6†	19103.47	20716.25†
DOM1-SNI	15818.3	15977.99	14481.73	14441.25
DOM1-Dep.	15557.2	18265.97†	13945	16670.49†
CMS1	15912.7	16165.97	14372.01	14595.28
CMS1-Rapid	23344	24570.5†	21811.74	22474.72†
HPC	18585	18830.76	17338.75	17234.76
HPC2	19344	19905.48	18397.22	18280.28
ISW	16055.5	16264.72	14667.01	14541.74
ISW-PINI	17626.5	17944.1	16422.01	16266.52
PARA	15048.3	15139.5	13589.2	13577.01
PINI	16286.7	17991.25	14625.97	16321.5
TI33	31137.99	34550.97	29433.27	33131.25
DOM1-NC	16455	17272.5	14825.1	15029.7

SE: single edge
DE: double edge
Clock frequency is 2 GHz

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Overall results: 1600 bytes of A and M

Implementation	Cycles	Critical Path(ns)	Throughput (Gbps)	Area (GE)	Goal
Unmasked, 4 rounds/cycle	2318	2	5.52	10124.24	-
Unmasked, 1 round/cycle	6048	1.11	3.81	7348.61	-
Masked, 1 cycle/round	8636	0.65	4.56	33131.25	P
Masked, 2 cycles/round	12088	0.6	2.35	20716.25	P
Masked, 3 cycles/round	18128	0.5	2.82	13276.52	P
Masked, 5 cycles/round	30208	0.5	1.69	14441.25	SNI
Masked, 7 cycles/round	42288	0.5	1.21	16266.52	PINI
Masked, 14 cycles/round	84568	0.5	0.6	15029.7	C

Conclusions

- We confirm observations made by other researchers that almost all masking schemes are based on assumptions that are not true for hardware implementations, mainly, the independence of leakage from different shares and composability.
- In order to obtain reliable benchmarking results, we suggest using strategies that avoid coupling of shares, *e.g.* DOM-NC, where the gadgets can be evaluated successfully. These strategies can be applied to any cipher.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Conclusions

- Romulus incurs between 2x and 4x the area for first-order masking.
- Using TI33, proposed in Indocrypt 2021 by Caforio et al., we were able to get an implementation that requires only 40 cycles per.
- The overhead due to the high-level mode of Romulus is almost negligible and almost all the cost comes from the underlying TBC.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Future work

- This work shows the need for large tweak masking-friendly TBCs.
- Applying the NC strategy to other designs. We believe comparisons should be between implementations designed with the same strategies.
- Studying the cost and trade-offs related to randomness needed.
- Higher-order masking.
- Studying the exploitability of the detected leakage and the attack costs if any.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE