Lightweight TBC-Based Modes for Small Hardware Implementations

Mustafa Khairallah[1]

NTU, Singapore

December 15, 2019

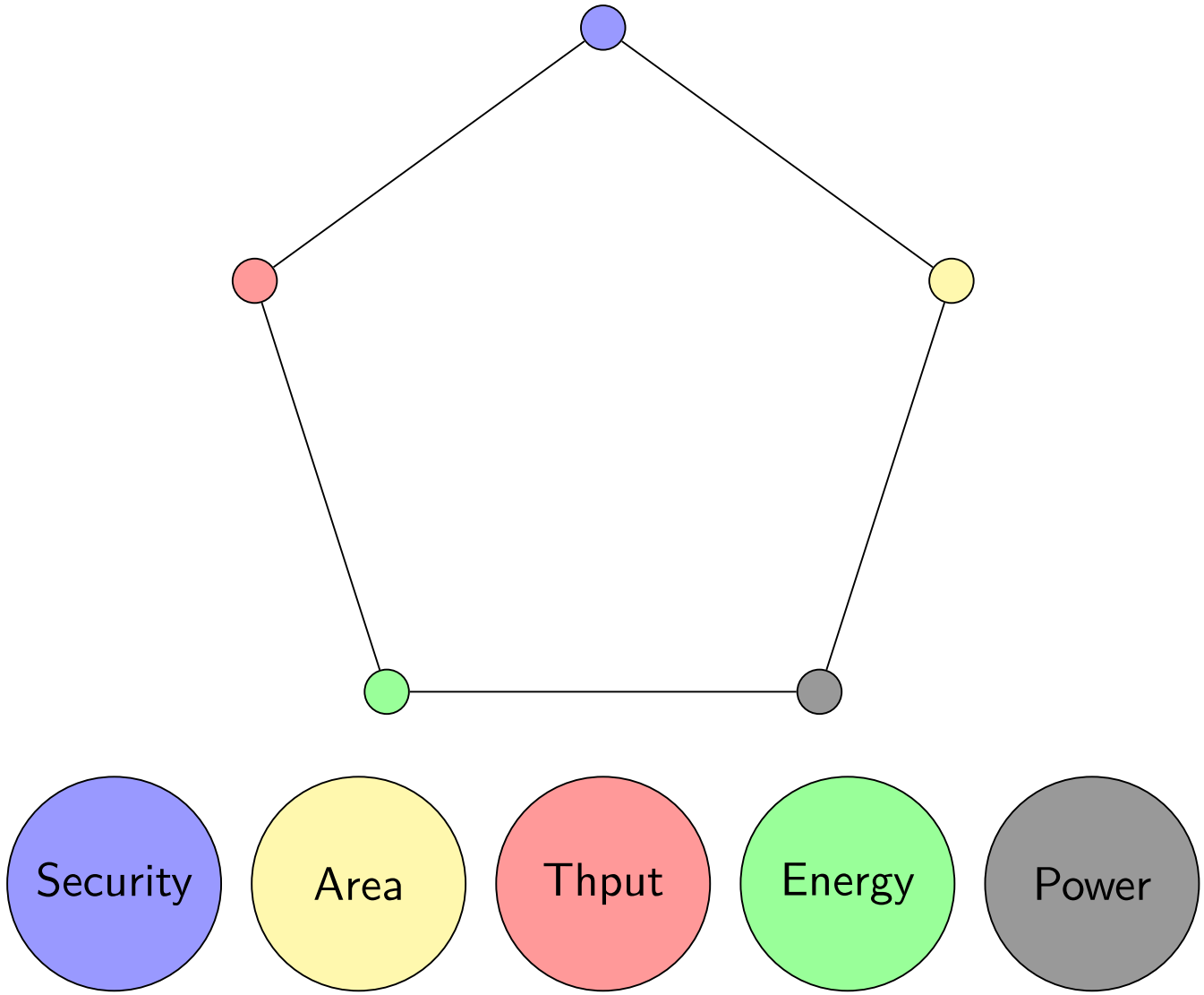
[1] Joint with Tetsu Iwata, Kazuhiko Minematsu and Thomas Peyrin

NANYANG TECHNOLOGICAL UNIVERSITY

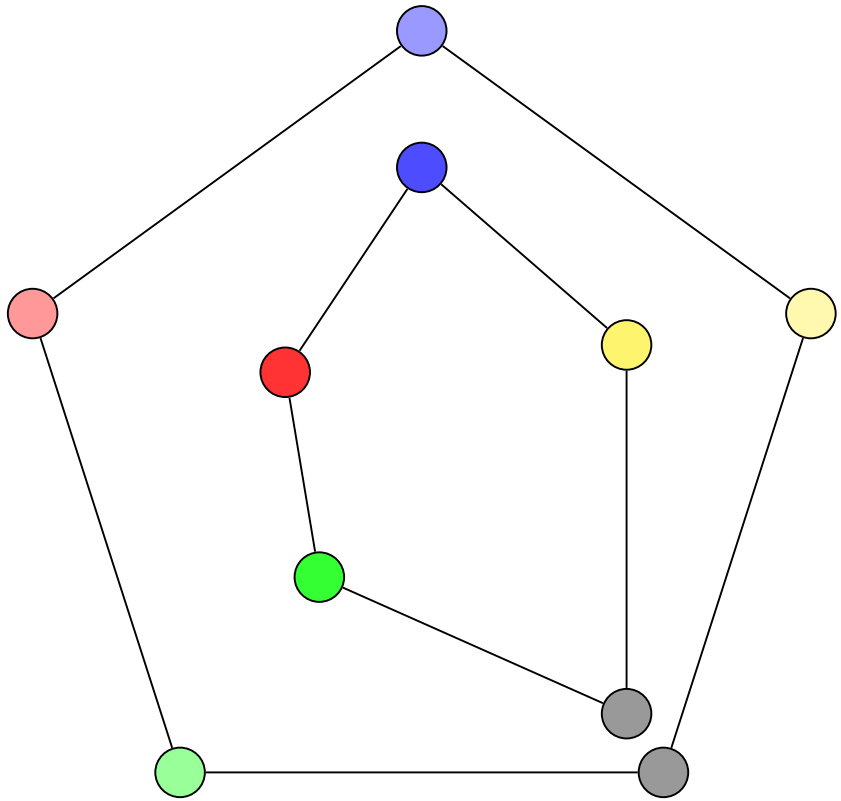
Overview

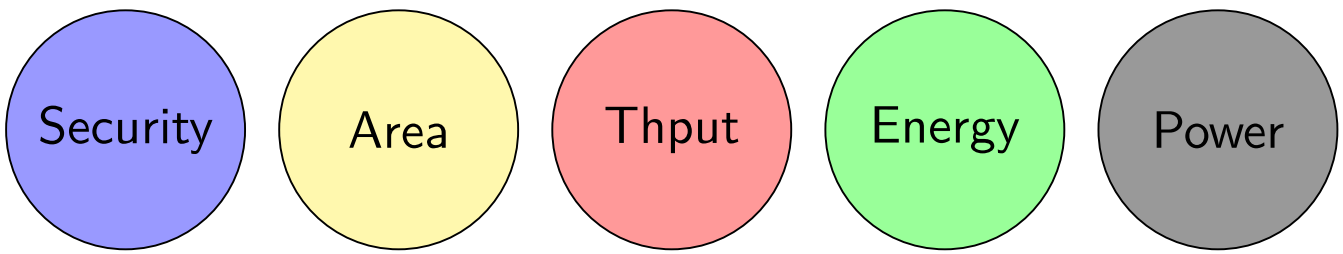
NANYANG TECHNOLOGICAL UNIVERSITY

Lightweight Cryptography Design Space



Lightweight Cryptography Design Space





NANYANG TECHNOLOGICAL UNIVERSITY

Lightweight Cryptography Design Space

NANYANG TECHNOLOGICAL UNIVERSITY

What went wrong?

1. Broken
2. Too slow
3. Too much energy
4. Too complex
5. Too big?!

Security Goals

1. Urgent Provable 128-bit security in the standard TBC model.
2. Urgent Easy to mask for side channel protection.
3. Optional Misuse resistance.

Area Goals

1. Urgent No extra state beyond the TBC.
2. Urgent No feed-forward.
3. Urgent No key/nonce storage.
4. Urgent Around 6000 GEs for round based implementations.
5. Urgent Below 10,000 GEs for threshold round based implementations.
6. Urgent Strictly inverse free.
7. Urgent Decryption is almost free.
8. Important Minimal use of multiplexers.
9. Important Around 3000 GEs for byte serial implementations.
10. Important A wide variety of trade-offs.

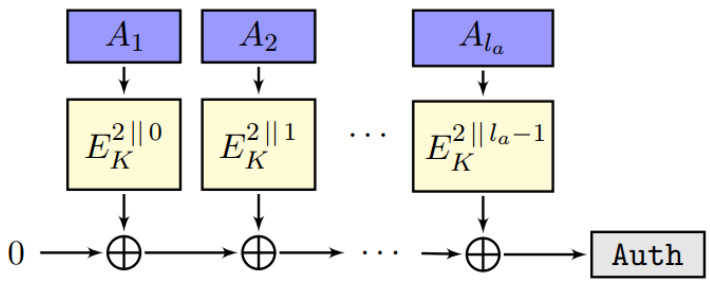
Performance Goals

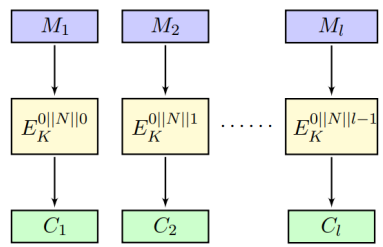
1. Urgent Fast for short messages and authentication.
2. Important Fast enough ($\gg 1$ Gbps on modern ASIC technologies) even with first order masking.
3. Important Wide performance range.
4. Important Competitive with AES-based designs and CAESAR lightweight portfolio (namely, Ascon).

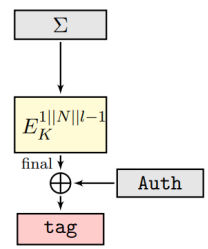
NANYANG TECHNOLOGICAL UNIVERSITY

Where to look?

- ΘCB3 has great features:
  - Standard TPRP assumption.
  - The security bound is independent of the length.
  - Parallelizable.
  - Not inverse free, needs n extra flip flops, needs an extra call.





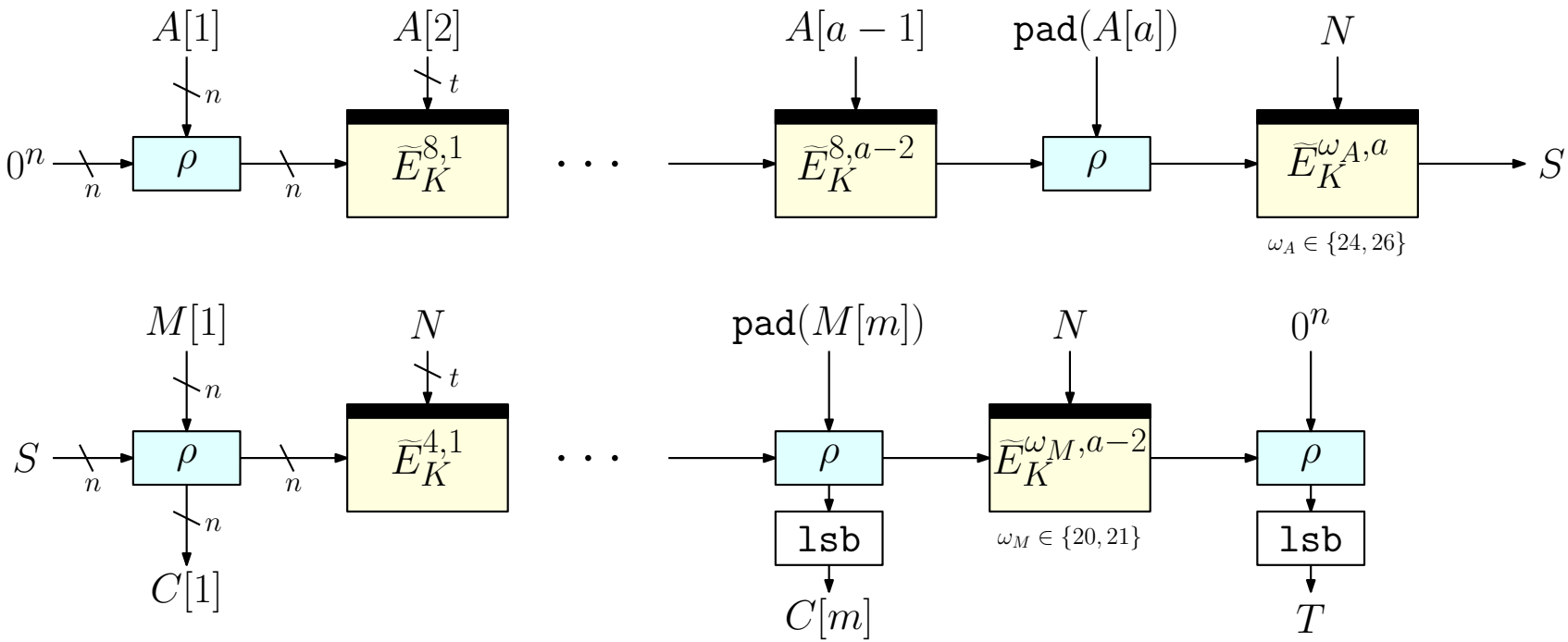


- ΘCB3 needs n extra flip flops at least.

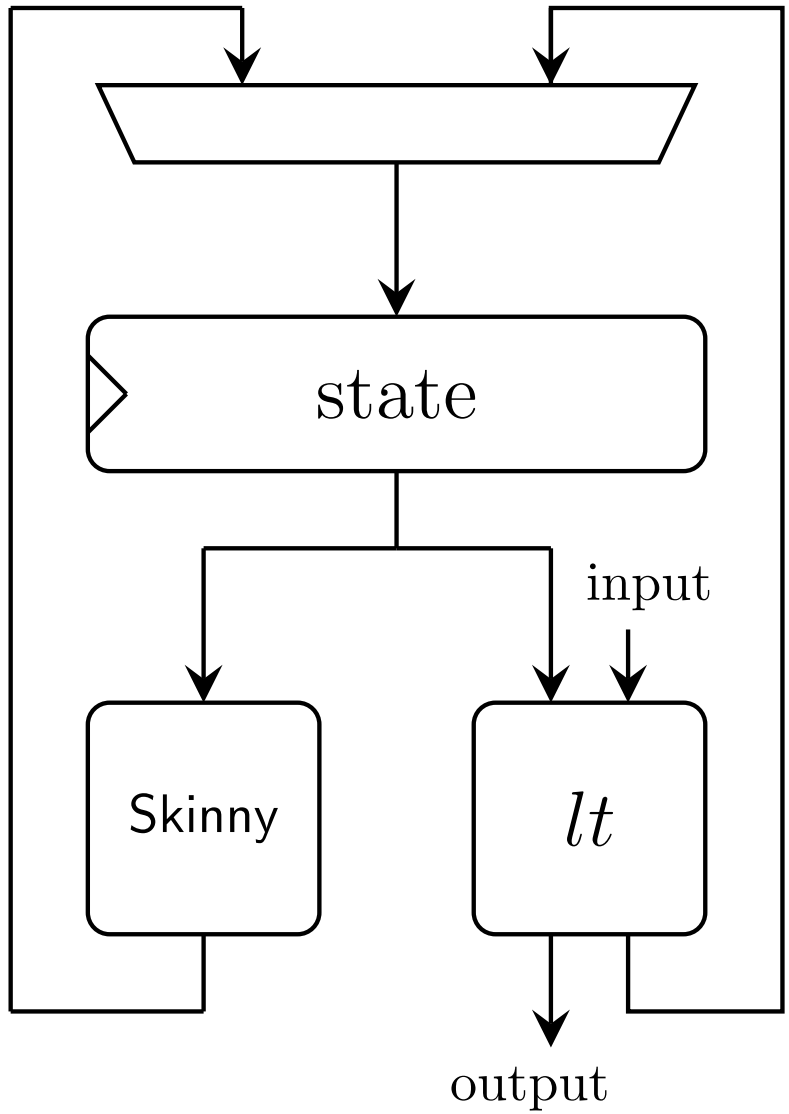
Where to look?

1. iCOFB is interesting starting point: lightweight.
2. ZAE has higher rate for authentication compared to encryption.

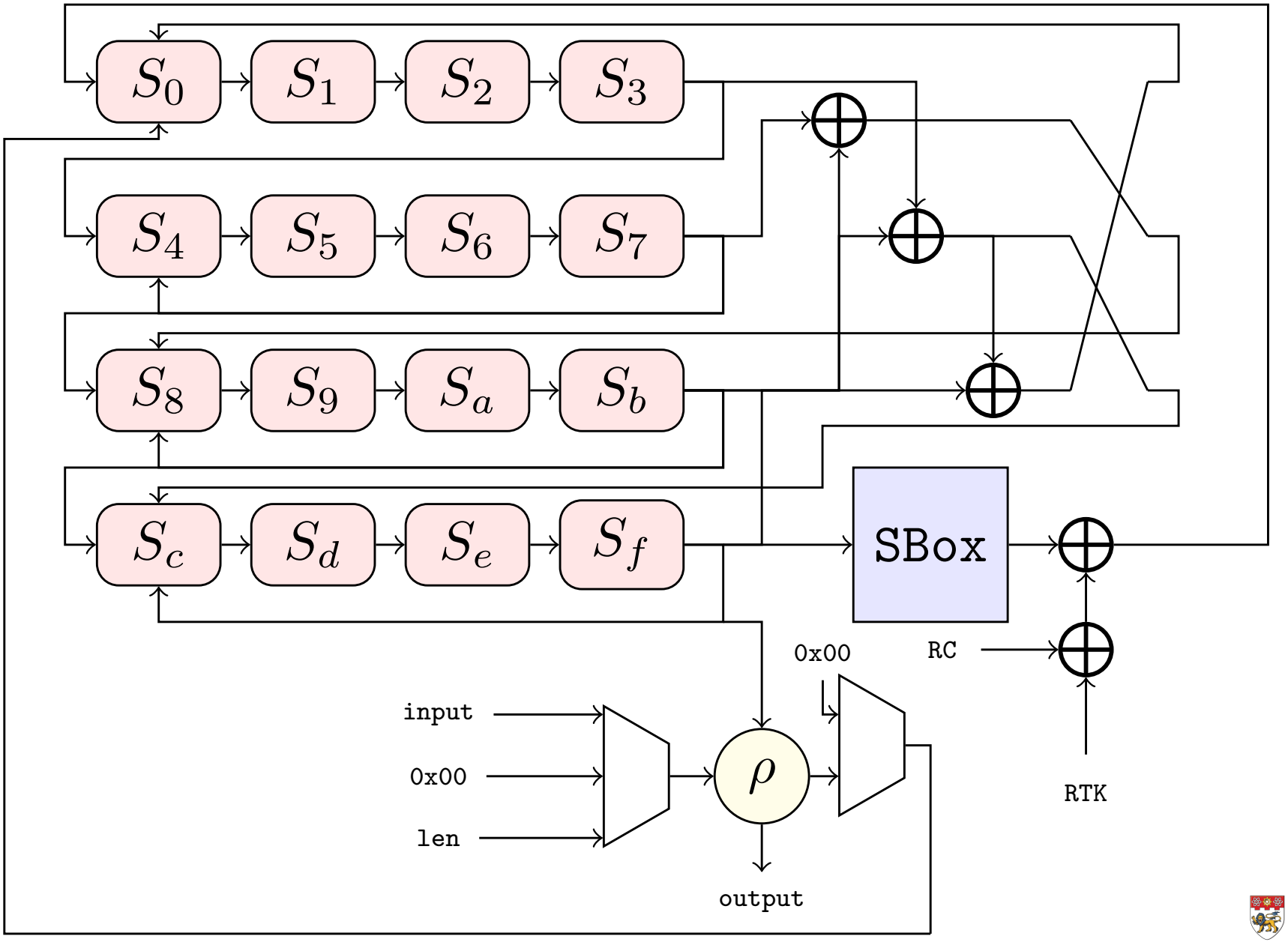
Romulus-N


$A[1]$
$A[2]$
$A[a-1]$
$\mathtt{pad}(A[a])$
N
0^n
ρ
$\widetilde{E}_K^{8,1}$
$\widetilde{E}_K^{8,a-2}$
$\widetilde{E}_K^{\omega_A,a}$
S
$\omega_A \in \{24, 26\}$
$M[1]$
N
$\mathtt{pad}(M[m])$
N
0^n
$\widetilde{E}_K^{4,1}$
$\widetilde{E}_K^{\omega_M,a-2}$
$\mathtt{lsb}$
$C[1]$
$C[m]$
T
$\omega_M \in \{20, 21\}$

Rx Architecture



Sx Architecture



Why Serial?

Parallelization is not that efficient in hardware.

Table: Synthesis results of the Deoxys-I-128 implementation using TSMC 65nm technology with x4 parallelization

Impl.	Area (KGE)	Max. Freq. (MHz)	Throughput (Mbps)	Efficiency (Mbps/KGE)
[KCP17]	59.53	847	7,227	121.40
ATHENa	53.37	549	4,684	87.76

NANYANG TECHNOLOGICAL UNIVERSITY

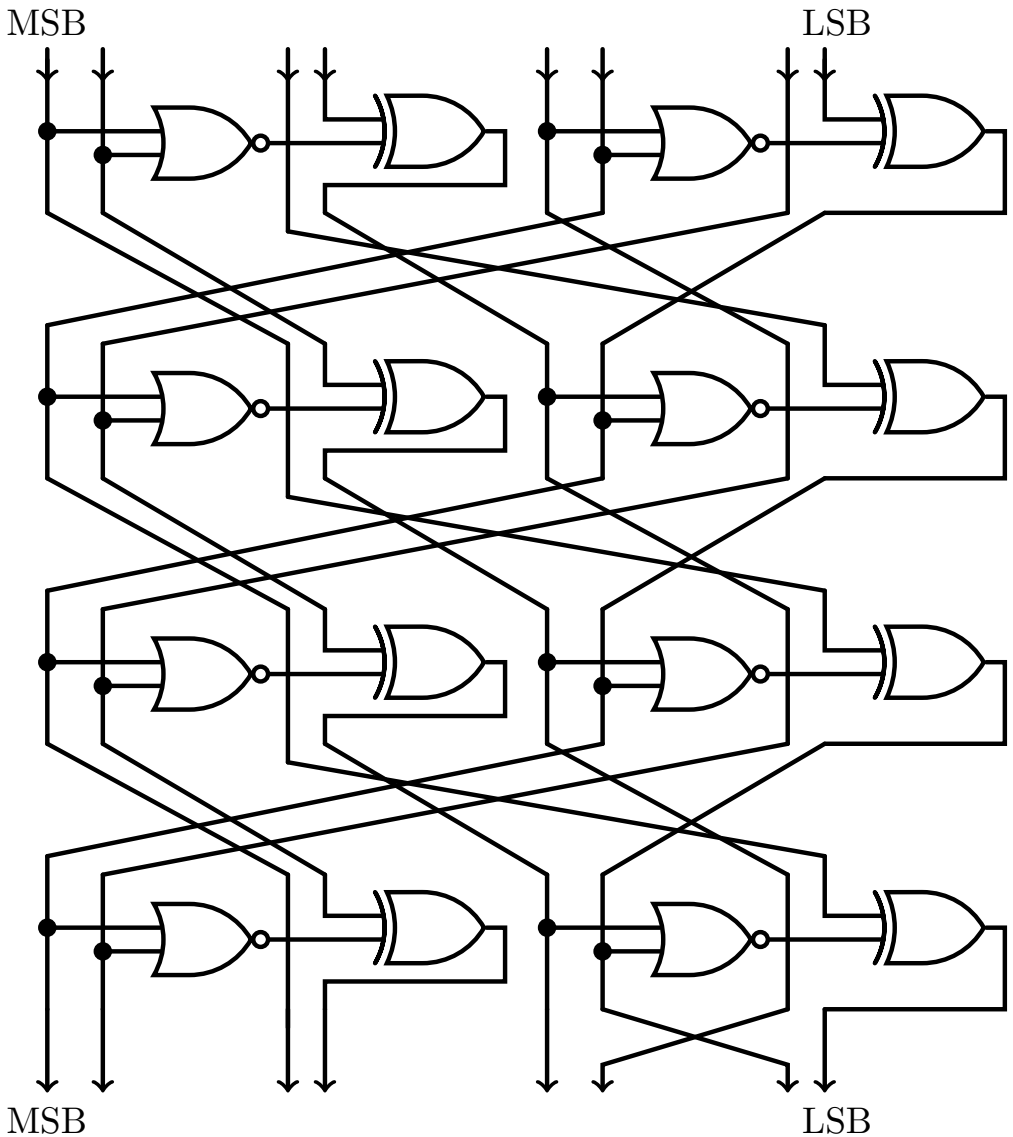
Why Serial?

- PFB: designed independently by Naito and Sugawara [NS19] around the same time of our work.
- It has partial parallelization for encryption only.
  - Doesn't work for authentication and decryption.
  - Makes the feedback function and padding complicated.

Why Skinny?

- Cheap.
- Wide variety of trade-offs.
- Easy to mask.
- Well analyzed with large security margin.
- Large tweakey space.
- Designed and supported by a strong team of researchers providing different implementations.

Skinny SBox



NANYANG TECHNOLOGICAL UNIVERSITY

Skinny Tweakey Schedule: Where the magic happens

Utilize the fully linear tweakey scheduling, mostly routing and renaming bytes

- Reverse tweakey schedule at the end of every TBC call, instead of keeping input
- Very low area, only 67 XOR gates! including both key correction and block counter.
- If we were to maintain tweakey state (due to modes/TBC), at least 320 FFs

Performance Trade-offs

Lightweight core is suitable to unroll, excellent trade-off

- Speeding up $\times 2$ by two-round unrolling : $\approx +$ 1,000 GEs, + 20 % of total area

Why LFSR Counter?

- Counters are the bottle neck of TBC based designs.
- A 50 bit arithmetic counter costs ~ 600 GEs, with depth $= 100$.
- An LFSR counter costs $7 \sim 15$ GEs.

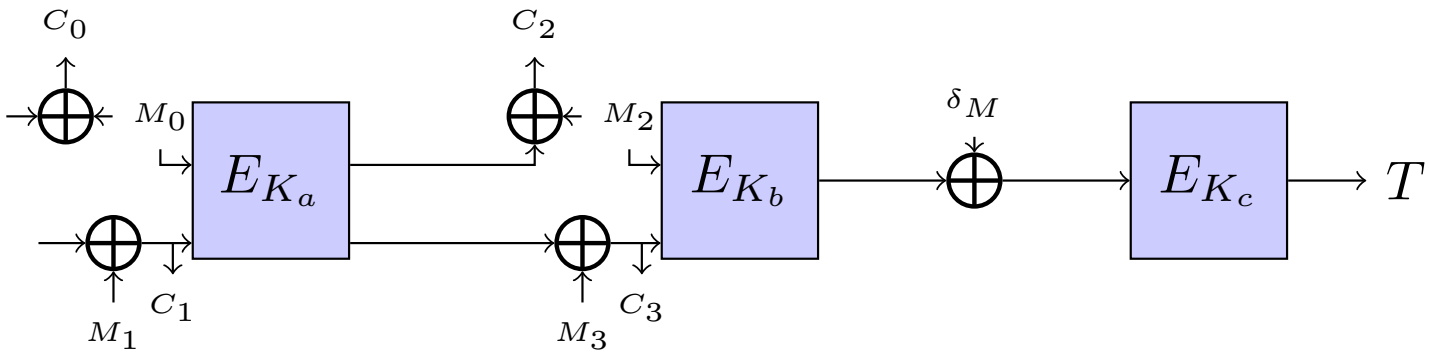
Why ρ?

Possible feedback types:

- Plaintext Feedback.
- Combined Feedback.
- Hybrid Feedback.

Is it all about the gate count?

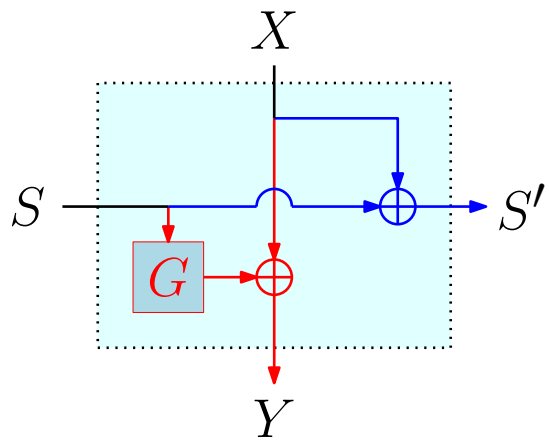
Hybrid Feedback (e.g. HyENA, mixFeed): n XORs.



128 XORs. 32 XORs and 32 MUXes for a 32-bit bus.

Is it all about the gate count?

- COFB Feedback: 288 XORs
- In order to serialize, we need 32 Flip Flops and 32 MUXes.



1. $S_0 = S_1$
2. $S_1 = S_2$
3. $S_2 = S_3$
4. $S_3 = S_3 \oplus S_0$

1. $T \leftarrow S_0, S_0 \leftarrow S_1$
2. $S_1 = S_2$
3. $S_2 = S_3$
4. $S_3 = S_3 \oplus T$

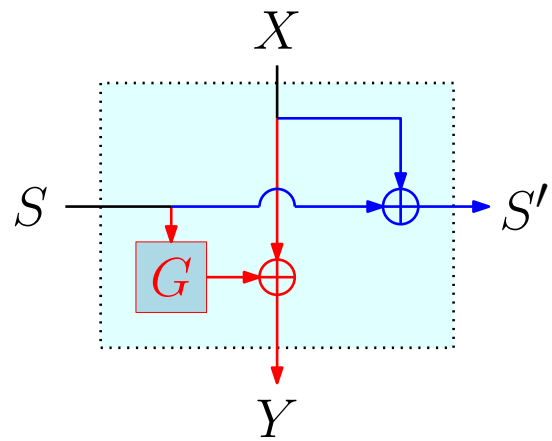
G goal: 0 XORs

- Impossible!!
- What about 1? Possible, yet needs extra flip flops and MUXes to serialize.
- What about 1 XORs for byte serial, 4 for 32-bit bus, 16 in total? Romulus Feedback.

ρ Feedback Function

Simple operation defined over bytes

- Each input byte affects one and only one byte.
- Rotation happens within the same byte.
- Computation is on the fly even for 8-bit buses.

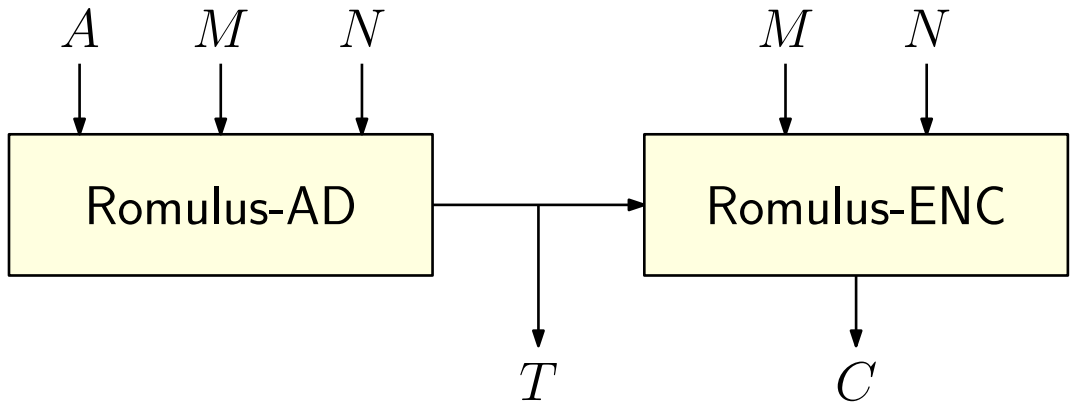


Choice of G

1. $S_0 = G_4(S_0)$
2. $S_1 = G_4(S_1)$
3. $S_2 = G_4(S_2)$
4. $S_3 = G_4(S_3)$

$$G_4 = \begin{pmatrix} G_s & 0 & 0 & 0 \\ 0 & G_s & 0 & 0 \\ 0 & 0 & G_s & 0 \\ 0 & 0 & 0 & G_s \end{pmatrix}$$

Romulus M-variants



- (Fully) Nonce-misuse-resistance via SIV [RS06].
- Greatly shares Romulus-N components (easy to implement both using the same ciruit).
- 1.5 Rate only (3 TBC calls to process two blocks).

Last pieces of the puzzle

- Cryptographers often neglect the cost of the control logic and padding, which is drastic for LWC.
- Kumar [KHK+17] showed that a naive/reference implementation of the CAESAR Hardware API requires 4 KGE, almost as much as the lightweight scheme itself.
- Our goals for the FSM: simple logic, simple padding, low area and low power.

Padding.

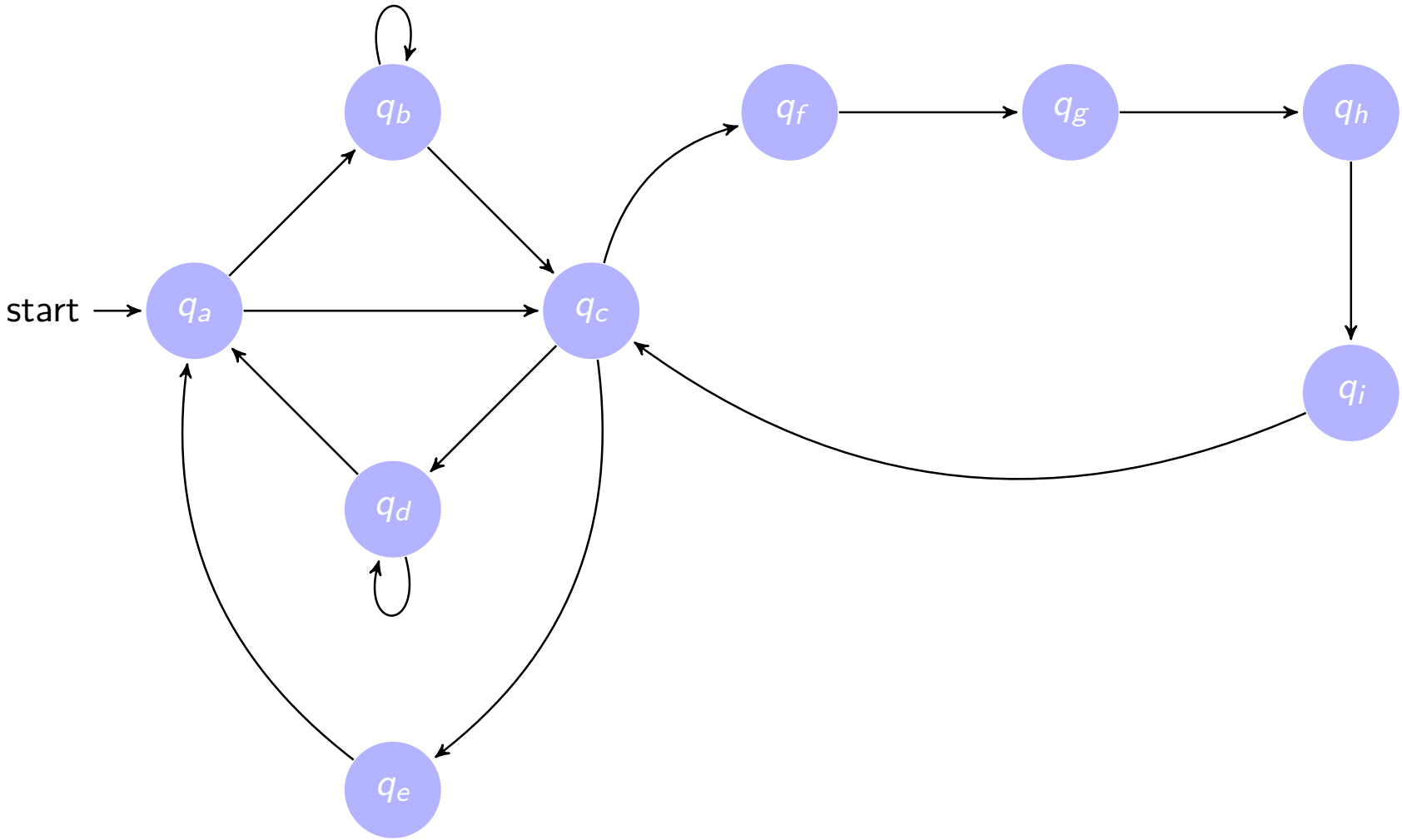
For $X \in \{0,1\}^{\leq 128}$ let

$$\mathtt{pad}_l(X) = \begin{cases} X & \text{if } |X| = 128, \\ X \,\|\, 0^{l-|X|-8} \,\|\, \mathtt{len}(X), & \text{if } 0 \leq |X| < 128, \end{cases}$$

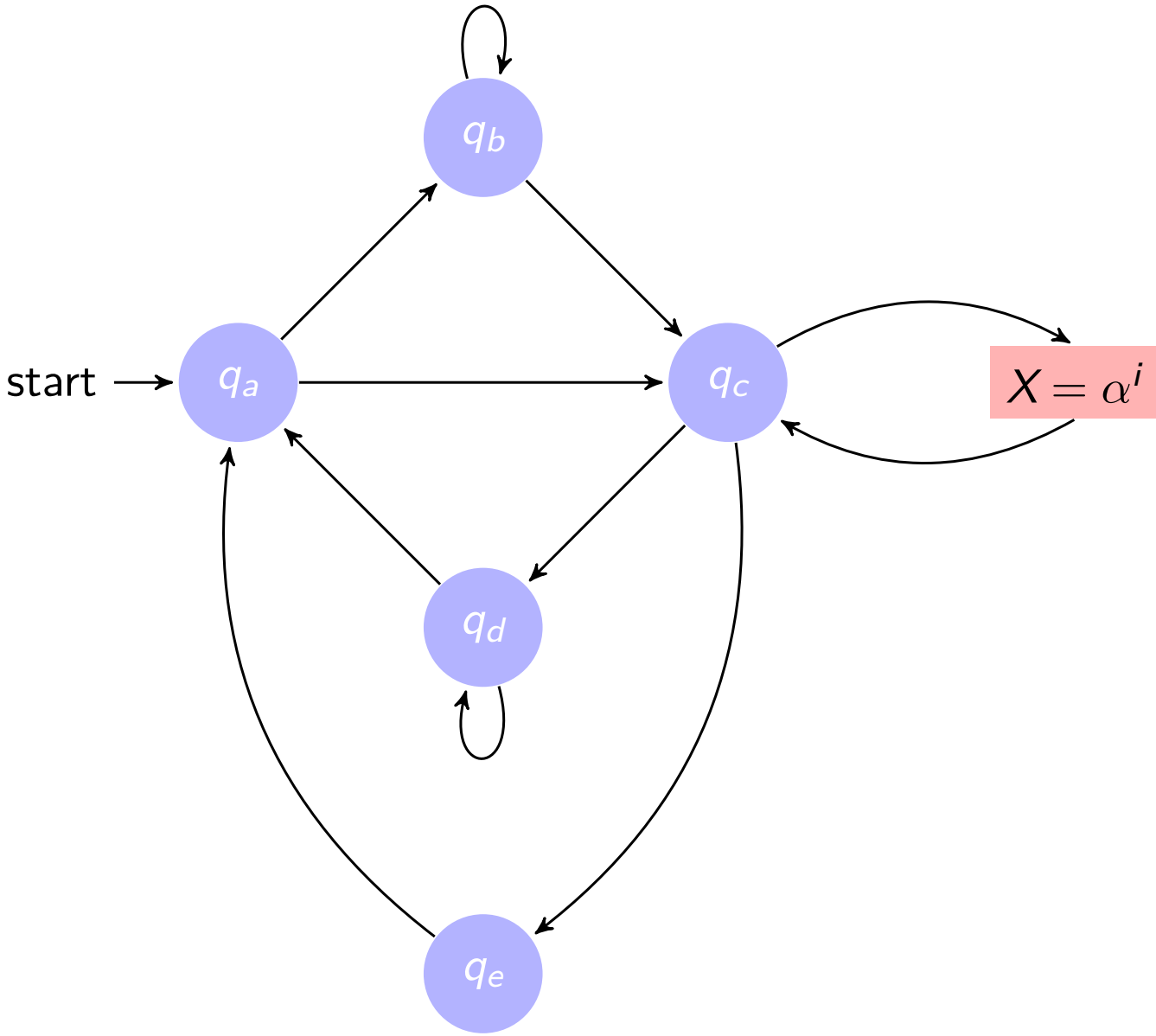
- A bus of width w needs only $w+4$ MUXes for padding, the length is already stored in the FSM so no decoding required.
- 10^* padding requires $\frac{n}{8}$ bit decoder and $\frac{9w}{8}$ MUXes.
- Security is exactly the same.

NANYANG TECHNOLOGICAL UNIVERSITY

FSM Optimization



FSM Optimization



Current Design Corners for Romulus-N1 on TSMC 65nm

Arch.	Area (GE)	Power (mW)	Energy (Enc/Auth) (pJ)	Throughput (Enc/Auth only) (Gbps)
R1	5772	0.2	24/13	4.2/8
R2	6635	0.25	16/9	8/14.2
R4	8740	0.32	13.8/8.5	8.8/14.5
R8	12990	0.45	21.1/14.4	7.3/10.6
S1	3318	0.15	489/247.5	0.131/0.259
P1	8048	0.28	36/19.2	4.2/8
PS	5154	0.21	772/391	0.131/0.259

NANYANG TECHNOLOGICAL UNIVERSITY

Comments and future work

- Romulus is not the limit: Remus-N2 acheives the same bit-security for smaller state and and smaller TBC, but under different assumptions (ICM instead of TPRP).
- Focus on low power/energy implementations.
- Study the design space of the threshold implementations more closely.
- Experiments on Romulus-M.
- Architectures to combine Romulus with TBC-based hash functions.
- Higher level Side-Channel protection solutions.

Thank you!

NANYANG TECHNOLOGICAL UNIVERSITY