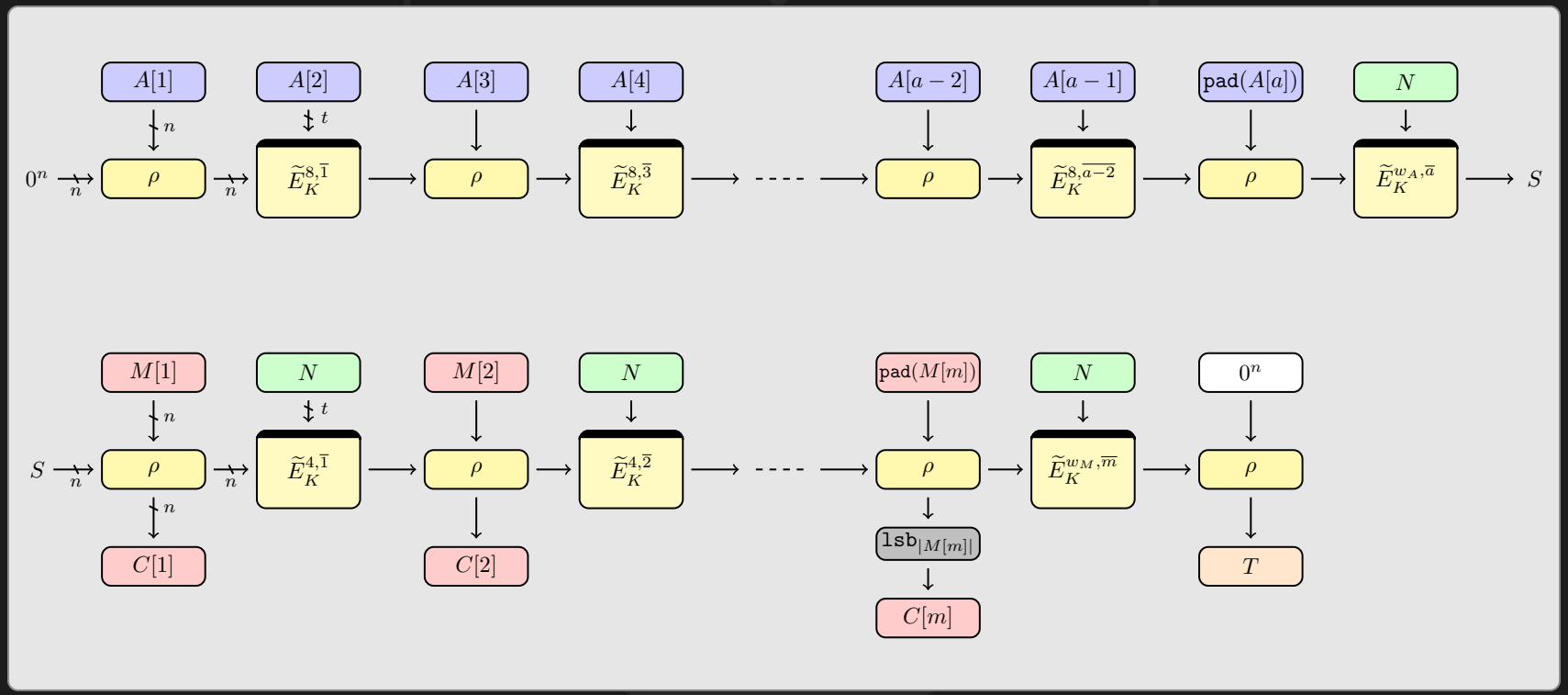
New Results on Romulus

T. Iwata, M. Khairallah, K. Minematsu and **T. Peyrin**

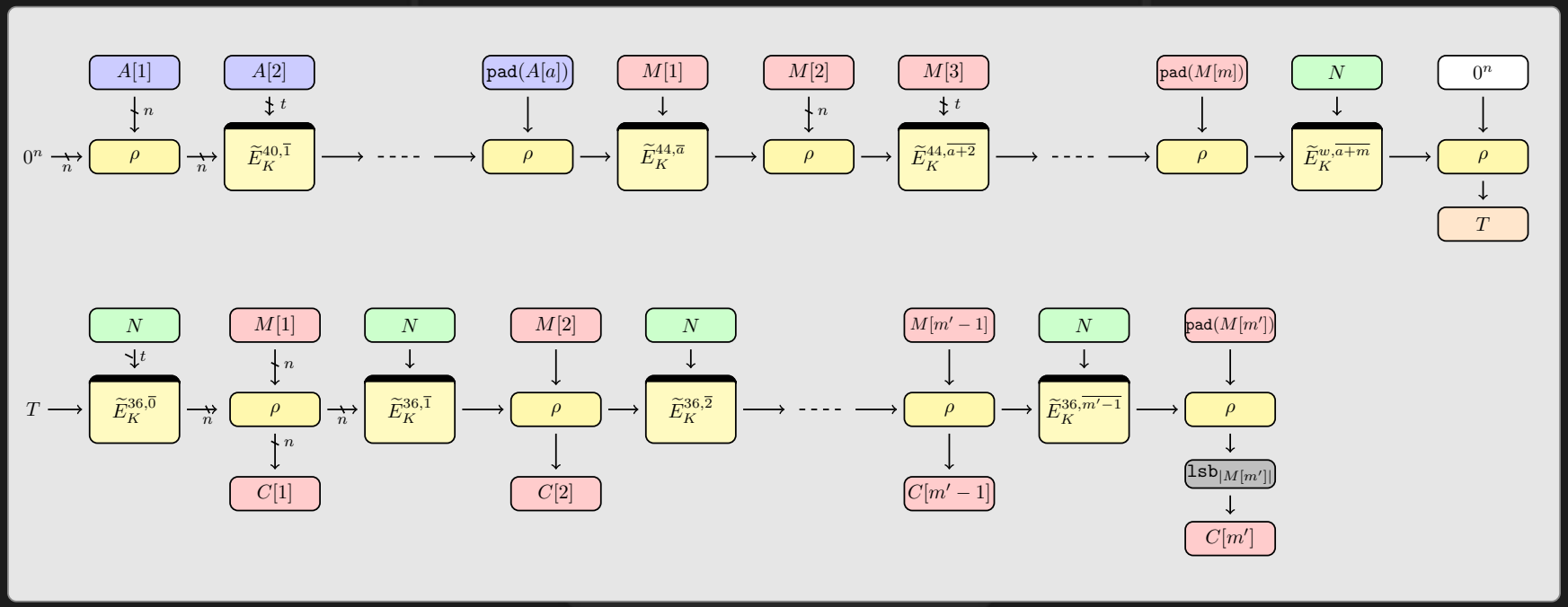
NIST LWC 2020

Virtual - October 19, 2020

Romulus-N : BBB nonce-respecting AEAD



Romulus-M : BBB nonce-misuse resistant AEAD



Summary of proposed updates and new results

We propose the following updates if selected for new round :

- reduce the number of rounds for the internal primitive
- simplify the submission by removing some variants
- add hash function Romulus-H
- add two leakage resilient modes Romulus-LR and Romulus-LR-TEDT

Additional new results :

- RUP security proof for Romulus-M
- new software/hardware implementations
- efficient threshold implementation

Update : round reduction for SKINNY-128/384

SKINNY :

- an ultra lightweight Tweakable Block Cipher (TBC)
- SKINNY is probably the most analysed primitive used in the competition (except AES or Keccak, already standardized)
- currently in Committee Draft stage at ISO (ISO/IEC 18033-7)
- already used in practical applications

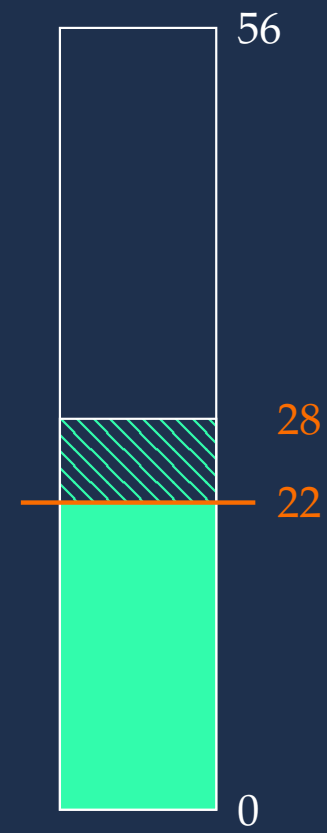
C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi,
T. Peyrin, Y. Sasaki, P. Sasdrich and S.M. Sim
CRYPTO 2016

https://sites.google.com/site/skinnycipher/

Update : round reduction for SKINNY-128/384

Security margin of SKINNY-128/384 is very (too?) large

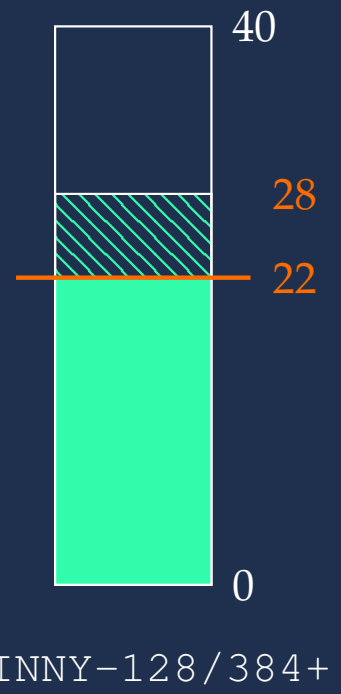
- SKINNY-128/384 has 56 rounds
- current best attack reaches 28 rounds with 2^{315} time, $> 2^{122}$ data
  (50% security margin!)
- for attacks with time/data limited to 2^{128}, best attack reaches 22 rounds
- SKINNY-128/384 was designed to handle even 384-bit keys, while Romulus uses it as a 128-bit security primitive



Update : round reduction for SKINNY-128/384

Security margin of SKINNY-128/384 is very (too?) large

- we reduce the rounds number from 56 to 40
- SKINNY-128/384+ has **40** rounds, proposed by SKINNY team
- still maintains **30% security margin**, even for unrealistic 2^{315} attacks
- 45% security margin if only considering $< 2^{128}$ time/data

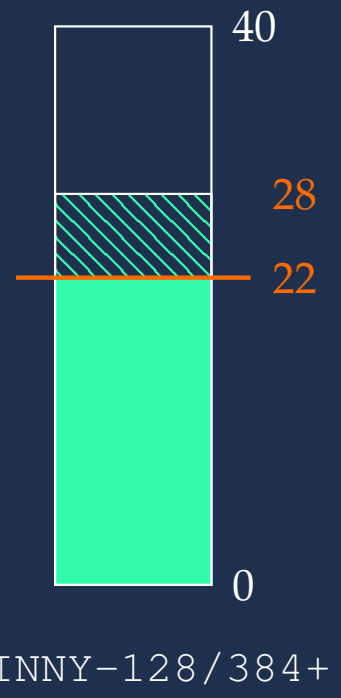


Update : round reduction for SKINNY-128/384

Security margin of SKINNY-128/384 is very (too?) large

- we reduce the rounds number from 56 to 40
- SKINNY-128/384+ has **40** rounds, proposed by SKINNY team
- still maintains **30% security margin**, even for unrealistic 2^{315} attacks
- 45% security margin if only considering $< 2^{128}$ time/data

We directly get a 1.4 performance gain on all current benchmarks



Update : only keep Romulus-N1 and Romulus-M1

We originally proposed 6 versions of Romulus to have several trade-offs.

Previous	Mode	Primitive	Comment
Romulus N1		SKINNY-128/384	
Romulus N2	Romulus N1	SKINNY-128/384	BBB nonce-respecting AEAD
Romulus N3		SKINNY-128/256	
Romulus M1		SKINNY-128/384	
Romulus M2	Romulus M1	SKINNY-128/384	BBB nonce-misuse resistant AEAD
Romulus M3		SKINNY-128/256	

Update : only keep Romulus-N1 and Romulus-M1

In order to simplify, we propose to only keep the main variants Romulus-N1 and Romulus-M1.

Previous	Mode	Primitive	Comment
Romulus-N1		SKINNY-128/384	
Romulus N2	Romulus N1	SKINNY-128/384	BBB nonce-respecting AEAD
Romulus N3		SKINNY-128/256	
Romulus-M1		SKINNY-128/384	
Romulus M2	Romulus M1	SKINNY-128/384	BBB nonce-misuse resistant AEAD
Romulus M3		SKINNY-128/256	

Update : only keep Romulus-N1 and Romulus-M1

Romulus : simpler and faster

New	Mode	Primitive	Comment
Romulus-N	Romulus N1	SKINNY-128/384+	BBB nonce-respecting AEAD
Romulus-M	Romulus M1		BBB nonce-misuse resistant AEAD

Update : only keep Romulus-N1 and Romulus-M1

Romulus : simpler and faster

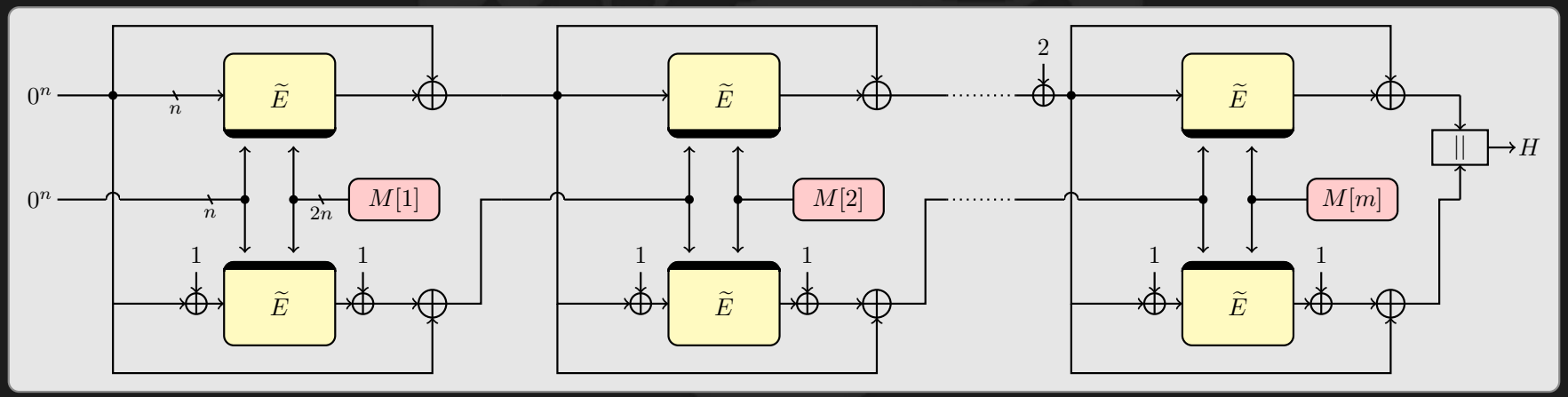
New	Mode	Primitive	Comment
Romulus-N	Romulus N1		BBB nonce-respecting AEAD
Romulus-M	Romulus M1		BBB nonce-misuse resistant AEAD
Romulus-H	MDPH	SKINNY-128/384+	Hash function / XOF
Romulus-LR	AET-LR		Leakage res. AEAD (CIML2 + CCAml1)
Romulus-LR-TEDT	TEDT		Leakage res. AEAD (CIML2 + CCAmL2)

Romulus-H : **hashing with** Romulus

Hashing with a 128-bit TBC is very easy with Naito's MDPH :

- build a 256-bit compression function h with the well-known **Hirose DBL** construction (rate 1) [FSE06]
- place h into the **Merkle-Damgård with Permutation** (MDP) mode [JoC12]

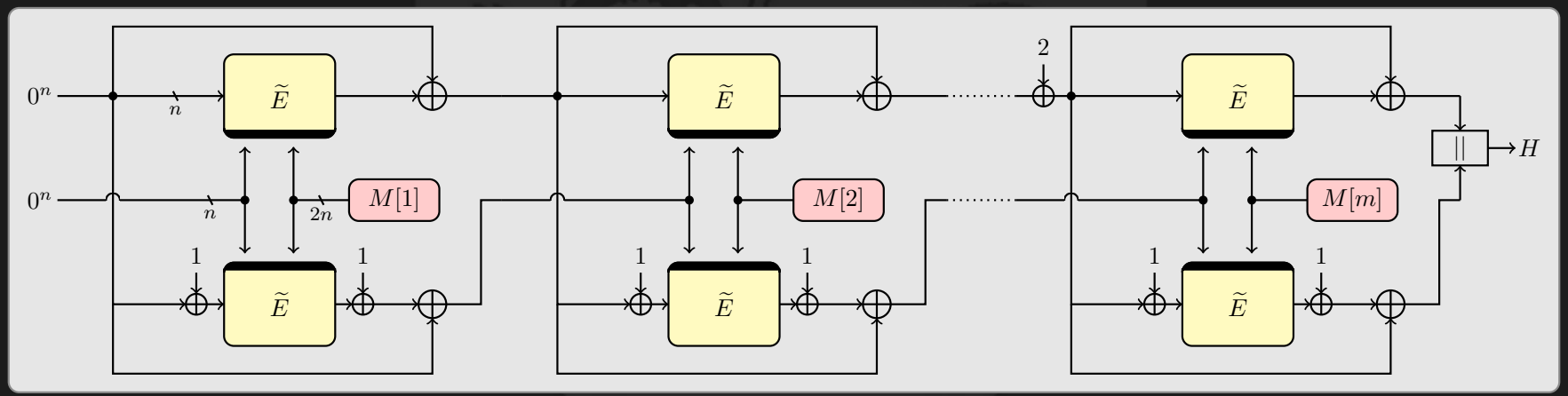
MDPH is indifferentiable from a (variable-input-length) random oracle up to about $(n - \log n)$ queries



Romulus-H : **hashing with** Romulus

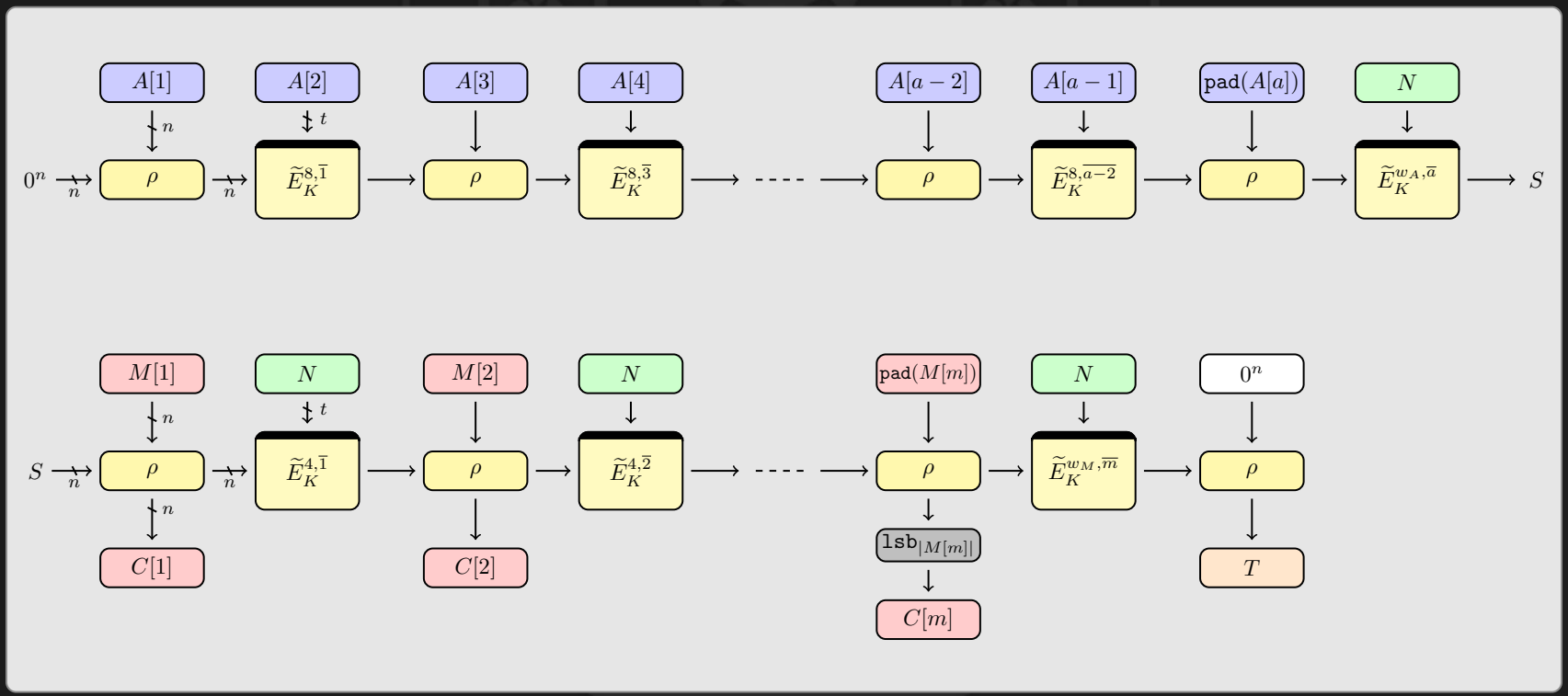
Extra features of Romulus-H :

- XOF : simply use $H(M||0)$, $H(M||1)$, $H(M||2)$, etc.
- Romulus-H can naturally adapt to very constrained area environments by reducing its message block size



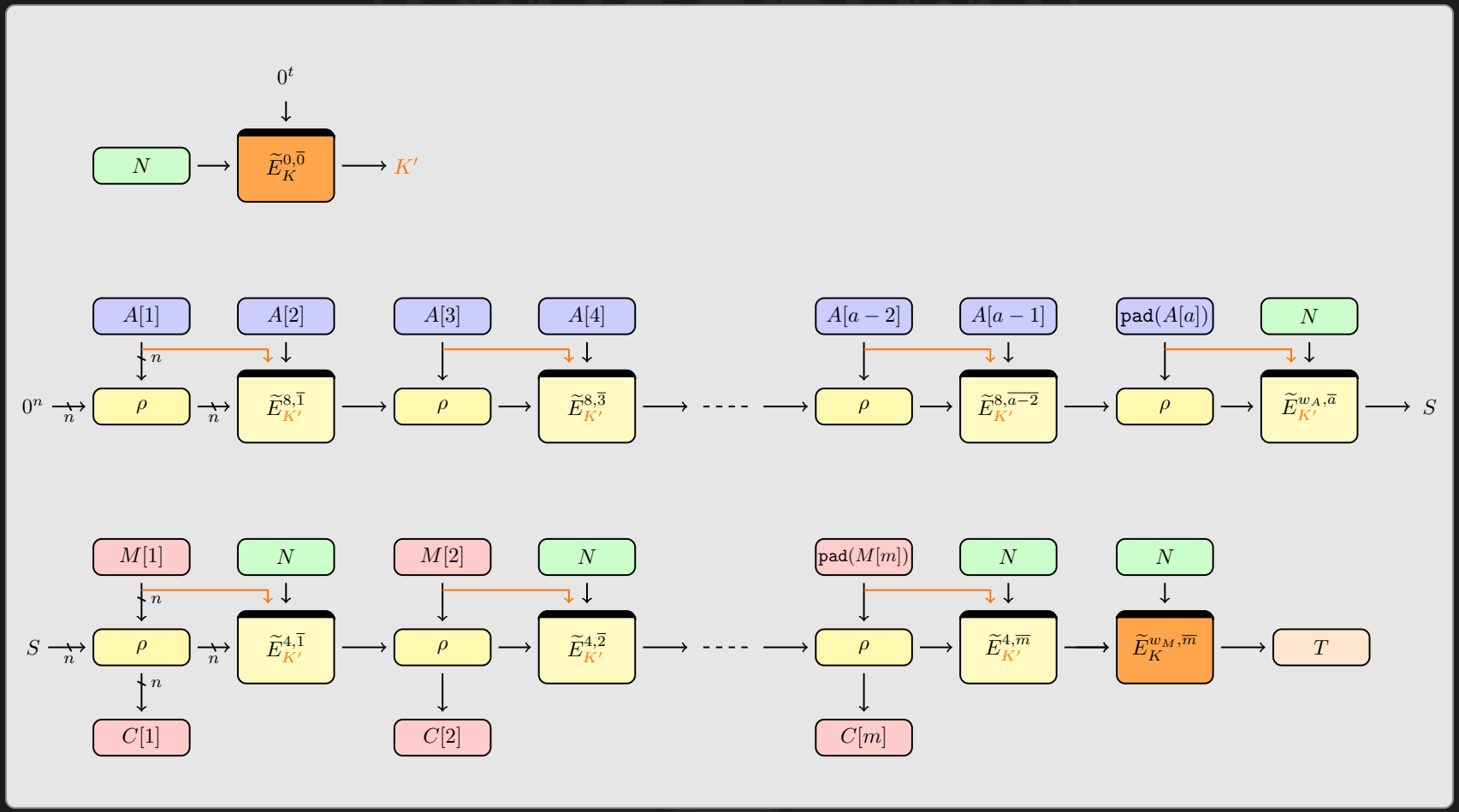
Romulus-LR : **leakage resilience with** Romulus

One can get some leakage resilience by simply feed-forwarding message block into the tweak input in Romulus-N + key/tag protect



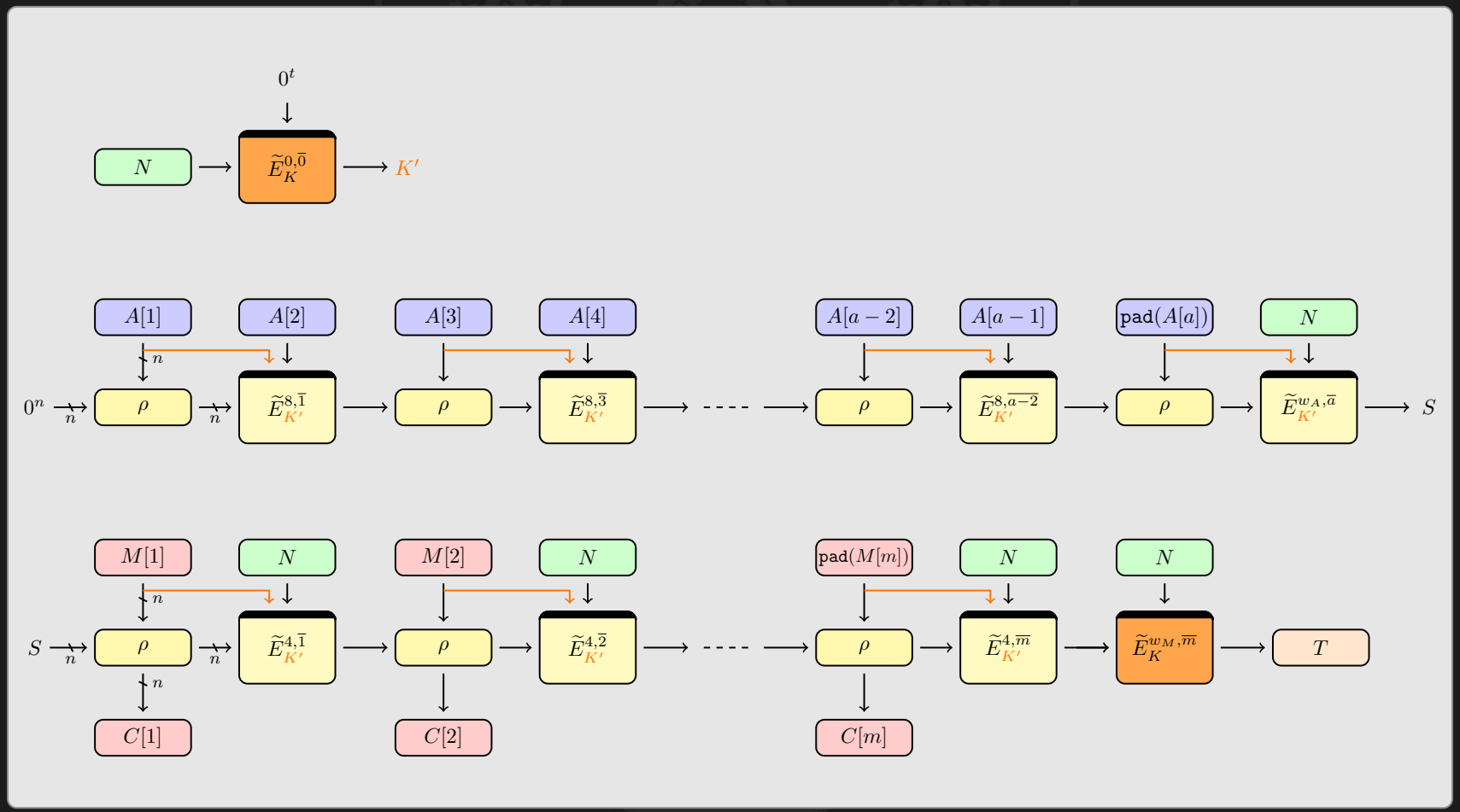
Romulus-LR : **leakage resilience with** Romulus

One can get some leakage resilience by simply feed-forwarding message block into the tweak input in Romulus-N + key/tag protect



Romulus-LR : **leakage resilience with** Romulus

Romulus-LR ensures CIML2 (best for integrity) + CCAml1



Romulus-LR-TEDT : strong leakage resilience

One can get some strong leakage resilience by simply using TEDT mode [CHES20] with `SKINNY-128/384+`

Romulus-LR-TEDT ensures
CIML2 (best for integrity) + CCAmL2 (best for privacy)

RUP security of Romulus-M

RUP security notion (relevant in case of limited memory) : result of decryption (possibly an unauthentic plaintext) is leaked before the verification result is obtained.

- integrity : Romulus-M is INT-RUP secure (both nonce-respecting and nonce-misuse adversary)
- privacy : Romulus-M is PA1 secure (Plaintext Awarness)

Software performances of Romulus

Cipher	Uno[1] avg. time [µs]
schwaemm256128v1	2038.020
tinyjambu128	2206.980
giftcofb128v1	2339.870
knot128v1	2362.620
hyenav1	2455.900
gimli24v1	2722.500
ascon128v12	2733.600
estatetwegift128v1	3010.040
xoodyakv1	3068.450
sundaegift96v1	3163.310
orangezestv1	3890.570
romulusn1	3901.450
spoc128sliscplight256v1	3995.640
spook128su512v1	4161.550
subterraneanv1	4236.020
saturninctrcascadev2	4782.230
twegift64lotusaeadv1	4809.160
⋮	⋮

Cipher	Uno[1] avg. time [µs]
schwaemm256128v1	2038.020
tinyjambu128	2206.980
giftcofb128v1	2339.870
knot128v1	2362.620
hyenav1	2455.900
gimli24v1	2722.500
ascon128v12	2733.600
romulusn1+	2870.170
estatetwegift128v1	3010.040
xoodyakv1	3068.450
sundaegift96v1	3163.310
orangezestv1	3890.570
spoc128sliscplight256v1	3995.640
spook128su512v1	4161.550
subterraneanv1	4236.020
saturninctrcascadev2	4782.230
twegift64lotusaeadv1	4809.160
⋮	⋮

Software performance rankings on AVR (8-bit) from
`lwc.las3.de/table.php`

Hardware performances of Romulus : FPGA

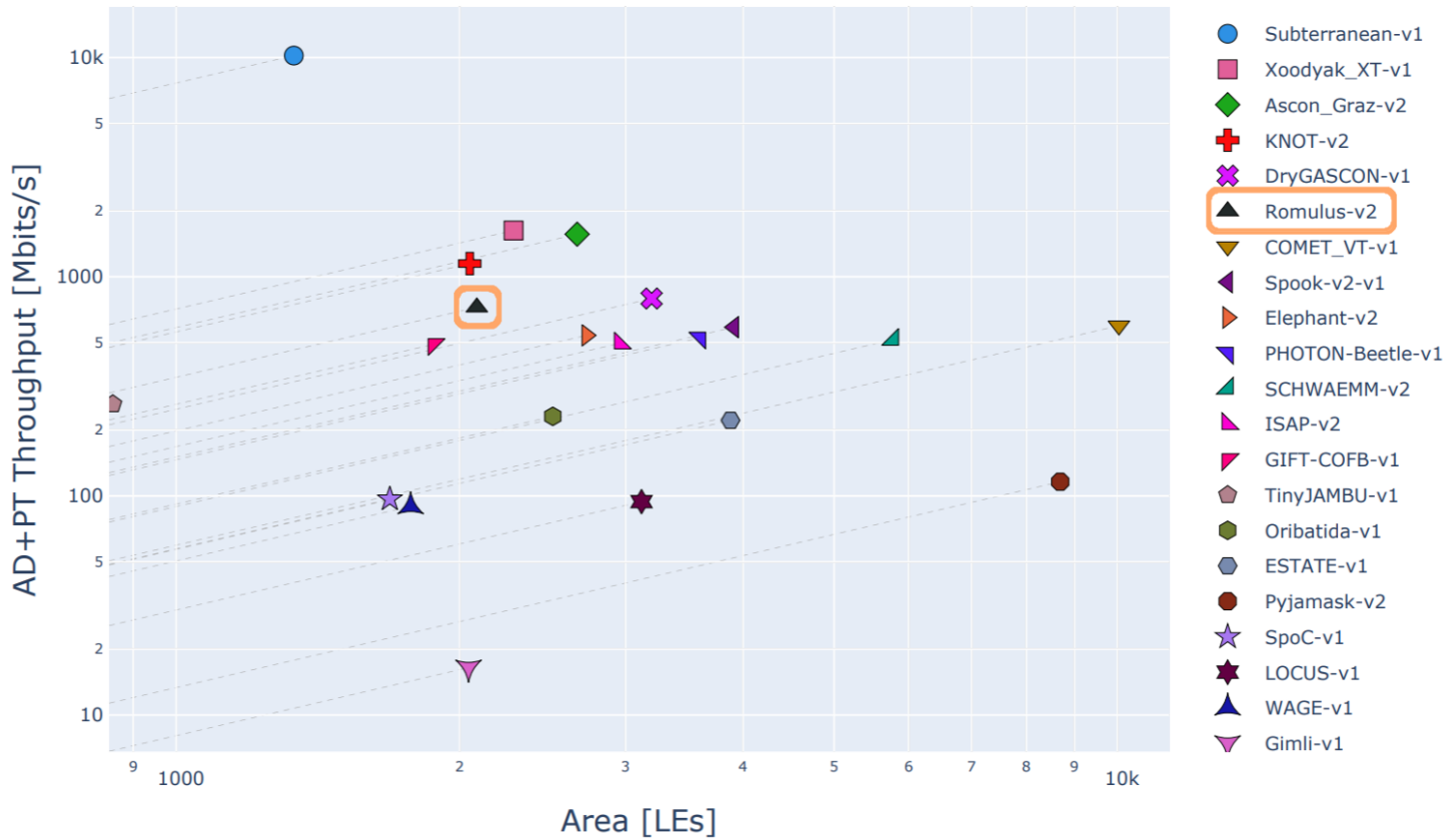


Figure 8: Cyclone-10-LP Encryption AD+PT Throughput for Long Messages vs LEs

FPGA performance from GMU

Hardware performances of Romulus : ASIC

Candidate	Throughput			Area	Power	Energy			Performance Efficiency			3 Mbps		
	16	64	1536			16	64	1536	Th./Area	Th./Power	Energy×Area	Th./Area	Th./Power	Energy×Area
DryGascon	4	4	4	7	7	4	4	4	4	4	5	6	8	7
Elephant	6	6	6	5	5	6	6	6	7	6	7	7	7	6
PHOTON-Beetle	5	5	5	6	6	5	5	5	6	5	6	5	5	5
Pyjamask	8	8	8	8	8	8	8	8	8	8	8	8	6	8
Romulus	3	3	3	2	2	2	2	2	3	2	2	2	2	2
Subterranean	1	1	1	3	3	1	1	1	1	1	1	3	4	3
TinyJambu	7	7	7	1	1	7	7	7	5	7	4	1	1	1
Xoodyak	2	2	2	4	4	3	3	3	2	3	3	4	3	4

TABLE – ASIC performance ranking from `https://github.com/mustafam001/lwc-aead-rtl/raw/master/asic-report.pdf`

Threshold implementation of Romulus

Threshold implementation for TBCs

As shown in [Spook,NaitoSS-EC20], TBC are great primitives for thres. impl. compared to BCs or sponges (only n-bit state to be protected)

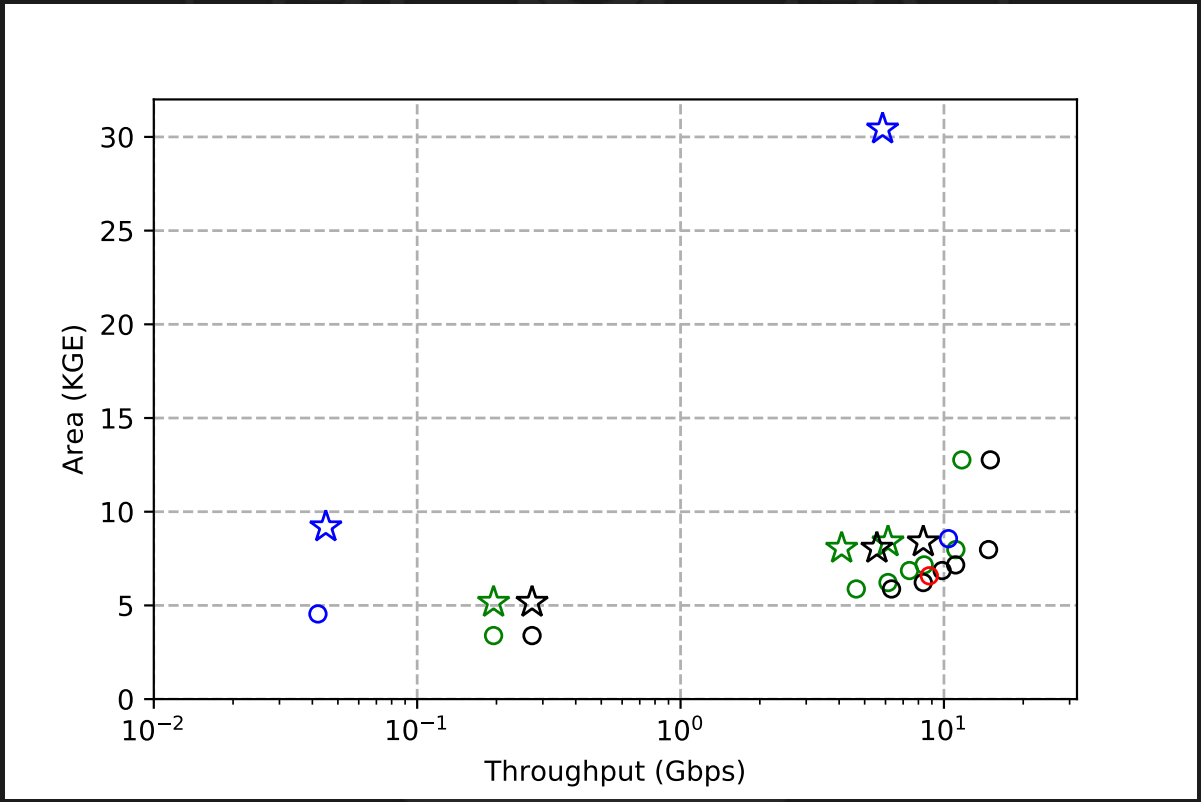


FIGURE – Throughput vs. Area trade-offs. Black : Romulus-N, Green : Romulus N1, Red : ACORN, Blue : ASCON. ○ : unprotected impl., ⋆ : threshold impl.

Romulus **features :**

- provably secure in standard model (unlike most LWC candidates)
- full 128-bit security (BBB unlike most LWC BC-based candidates)
  Romulus-N priv. bound is 0, auth is $q_d/2^\tau$, doesn't depend on #enc queries (unlike most LWC candidates)
- SKINNY is a stable and well studied primitive, large security margin, no distinguisher (unlike many LWC sponge-based candidates)
- easy nonce-misuse resistance mode (unlike most LWC candidates)
  birthday with graceful degradation so ~full security in practice
- no or low overhead for small messages (unlike all LWC sponge-based candidates)
  1 AD and 1 M n-bit blocks need 2 TBC calls with Romulus
- among the very top hardware efficient LWC candidates
- among the top-tier software efficient LWC candidates
  (among top for 4 or 8-bit)
- side-channel protection :
  implementation protection : efficient TBC threshold impl.
  mode protection : Romulus-LR and Romulus-LR-TEDT

Thank you !