Romulus as NIST LWC Finalist

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

NIST LWC 2022

Virtual - May 11, 2022

Romulus **versions**

Version	**Mode**	**Primitive**	**Comment**
Romulus-N	Romulus-N1	`SKINNY-128/384+`	BBB nonce-respecting AEAD
Romulus-M	Romulus-M1		BBB nonce-misuse resistant + RUP AEAD
Romulus-T	TEDT		Leakage res. AEAD (CIML2 + CCAmL2)
Romulus-H	MDPH		Hash function

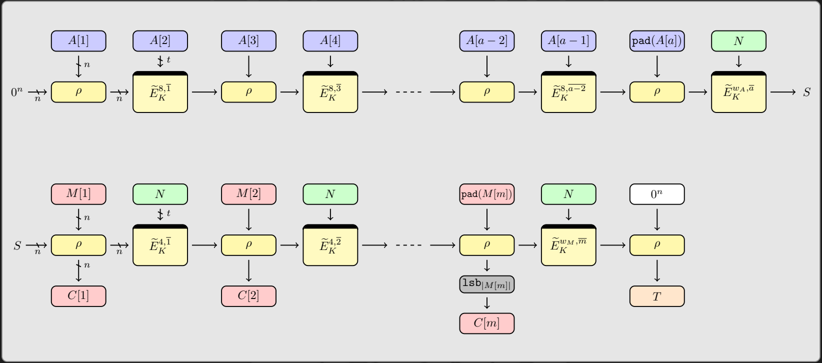
All our versions provide $\sim$ **128-bit security** - time and data

(in contrary to many remaining candidates)

Romulus-N/Romulus-M security proofs are in the **standard model**

(in contrary to all remaining candidates except `GIFT-COFB`)

Romulus-N : BBB nonce-respecting AEAD

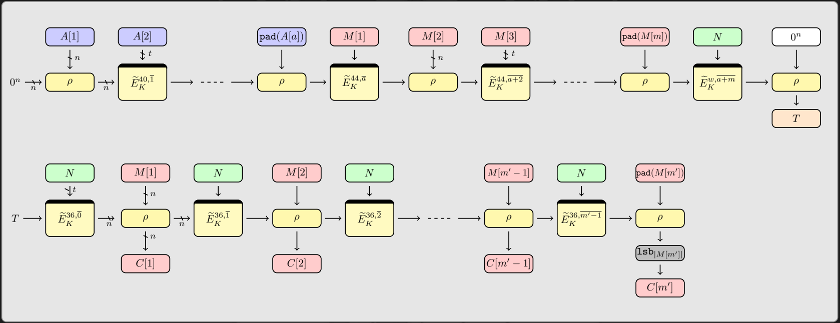


Provides **BBB 128-bit security** - data and time

(in contrary to many remaining candidates)

New : Provides **nonce-misuse resilience**

Romulus-M : BBB nonce-misuse resistant AEAD


$A[1]$
$A[2]$
pad$(A[a])$
$M[1]$
$M[2]$
$M[3]$
pad$(M[m])$
N
0^n
n
t
0^n
ρ
$\widetilde{E}_K^{40,\overline{1}}$
$\widetilde{E}_K^{44,\overline{a}}$
$\widetilde{E}_K^{44,\overline{a+2}}$
$\widetilde{E}_K^{w,\overline{a+m}}$
T
N
$M[1]$
$M[2]$
$M[m'-1]$
pad$(M[m'])$
T
$\widetilde{E}_K^{36,\overline{0}}$
$\widetilde{E}_K^{36,\overline{1}}$
$\widetilde{E}_K^{36,\overline{2}}$
$\widetilde{E}_K^{36,\overline{m'-1}}$
$C[1]$
$C[2]$
$C[m'-1]$
lsb$_{|M[m']|}$
$C[m']$

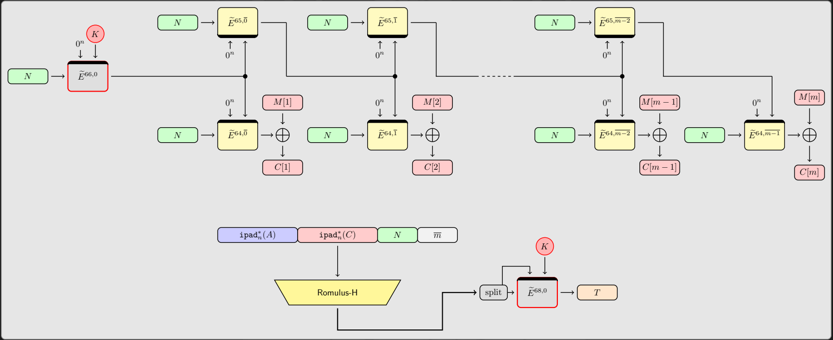
Provides **nonce-misuse resistance** (strong MRAE notion)

(in contrary to <u>all</u> remaining candidates)

Provides **Release of Unverified Plaintext** security (INT-RUP + PA1)

(in contrary to all remaining candidates except ELEPHANT)

Romulus-T : Leakage resilient AEAD

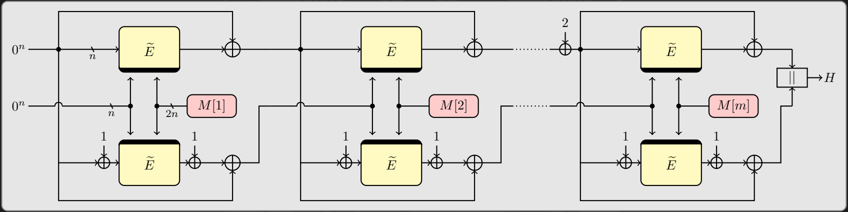


Provides **CIML2** (best for integrity) + **CCAmL2** (best for privacy)

(in contrary to all remaining candidates except `ISAP`)

Provides **nonce-misuse resilience**

Romulus-H : rate 1 Hash function



Indifferentiability up to $n - \log_2 n$

Can easily/efficiently provide **XOF** functionality

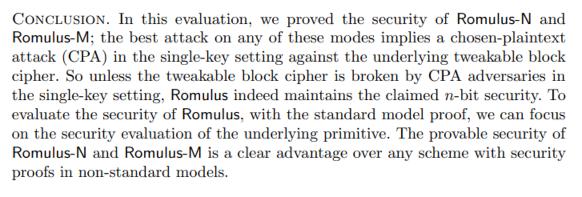
Security

Security proofs review by third-party

Confidence in a security proof correctness is very important. Our Romulus-N/Romulus-M proofs have been reviewed and published in ToSC NIST LWC and we continue verifying them, but we also adopted an approach of proof verification through a third-party review.

Third-party analysis of the Romulus-N/Romulus-M operating modes conducted by **Prof. Jooyoung Lee** (KAIST, Korea). The report confirms the correctness of the provable security result by presenting an independent proof with a different proof strategy. Full report here :

https://romulusae.github.io/romulus/docs/Security_evaluation_Romulus_Jooyoung_Lee.pdf



Conclusion. In this evaluation, we proved the security of Romulus-N and Romulus-M; the best attack on any of these modes implies a chosen-plaintext attack (CPA) in the single-key setting against the underlying tweakable block cipher. So unless the tweakable block cipher is broken by CPA adversaries in the single-key setting, Romulus indeed maintains the claimed n-bit security. To evaluate the security of Romulus, with the standard model proof, we can focus on the security evaluation of the underlying primitive. The provable security of Romulus-N and Romulus-M is a clear advantage over any scheme with security proofs in non-standard models.

New Romulus-H proof

Romulus-H is based on the Naito's MDPH construction (basically **Hirose DBL** compression function construction [FSE06] inside a **Merkle-Damgård with Permutation** (MDP) mode [JoC12]).

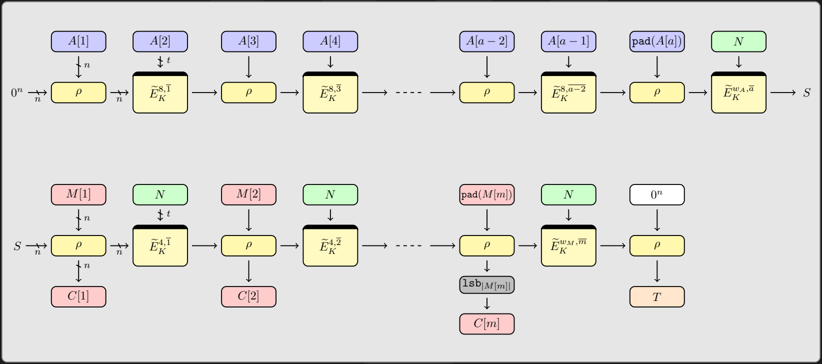
New MDPH and Romulus-H security proof

Previous analysis from Naito's contained a gap (in the definition of the simulator simulating the decryption of the underlying block cipher). We proposed a new MDPH and Romulus-H security proof, **same bounds up to constants -** published at IET Info Sec journal (2022) : `https://eprint.iacr.org/2021/1469.pdf`

New Romulus-N nonce-misuse resilience proof

New nonce-misuse resilience proof for Romulus-N

New nonce-misuse resilience proof for Romulus-N (ongoing work) : perfect for privacy, birthday for authenticity with graceful degradation (wrt nonce repetition).



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

Why Romulus-M is very well suited for lightweight

For a constrained device, it is difficult :

- ▷ to **ensure the non-repetition of a nonce** (counter requires synchronization, storing nonces requires a lot of memory, generating them randomly requires a good/non-buggy randomness source)
- ▷ to **retain the result of decryption in secure memory** until the verification result (large secure memory is difficult)

RUP security of Romulus-M

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

Nonce-misuse resistance of Romulus-M

integrity/privacy : Romulus-M is MRAE secure (up to birthday bound, with graceful degradation with number of nonce repeats).

Romulus-M is the ONLY remaining design to have RUP (except ELEPHANT) and MRAE, for a cost that is slightly more than Romulus-N and almost the same design

SKINNY family of Tweakable Block Ciphers

SKINNY:

- ▷ an ultra lightweight Tweakable Block Cipher (TBC) family
- ▷ SKINNY is with ASCON probably the most analysed primitive used in the competition (except Keccak, already standard)
- ▷ Published as ISO/IEC standard : ISO/IEC 18033-7:2022
- ▷ 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/

Current best attacks on SKINNY-128/384 and SKINNY-128/384+

Hadipour *et al.* (ePrint 2020:1317 and FSE 2022) [HBS20] :

- related-key rectangle attacks up to 30 rounds (2^{361} time, 2^{125} data)
- with one TK word fixed (TK2), up to 24 rounds (2^{209} time, 2^{125} data)
- distinguisher on 25 rounds with prob. $2^{-116.6}$ (TK2 : 21 rounds 2^{-114})

Qin *et al.* (ePrint 2021:656 and FSE 2022) [QDW+21] :

- related-key rectangle attacks up to 30 rounds (2^{341} time, 2^{122} data)
- with one TK word fixed (TK2), up to 25 rounds (2^{226} time, 2^{124} data)
- distinguisher on 22 rounds with prob. $2^{-101.5}$ (TK2 : 19 rounds 2^{-117})

Delaune *et al.* (FSE 2022 best paper) [DDV22] :

- related-key boomerang distinguisher on 24 rounds (2^{86} time/data)
- with one TK word fixed (TK2) up to 20 rounds (2^{86} time/data)

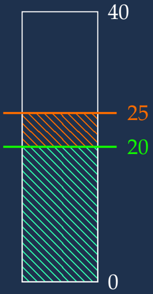
In contrary to many candidates, our internal primitive still have **no distinguisher** (by far).

The security margin of SKINNY-128/384+

A large security margin for SKINNY-128/384+

SKINNY-128/384+ has **40** rounds, proposed by the SKINNY team

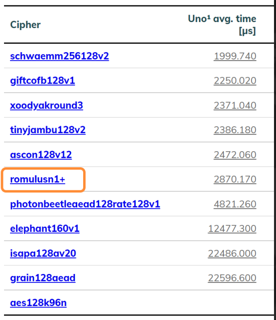
- For time/data limited to 2^{128}, current best attack reaches 25 rounds : we maintain a **37% worst case security margin**
- ... and even more if we :
  - restrict to 2^{64} data (probably 1 less round)
  - exclude related-key attacks (probably 4 less rounds)
  - consider the entire Romulus constructions
  - don't allow nonce to repeat
  - actual security margin $\gtrapprox 50\%$



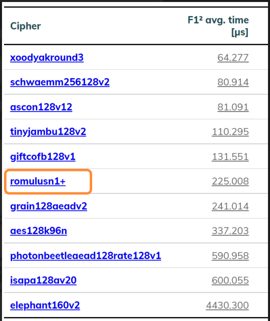
SKINNY-128/384+

Performances and Implementations

Software performances of Romulus



Cipher	Uno¹ avg. time [μs]
schwaemm256128v2	1999.740
giftcofb128v1	2250.020
xoodyakround3	2371.040
tinyjambu128v2	2386.180
ascon128v12	2472.060
romulusn1+	2870.170
photonbeetleaead128rate128v1	4821.260
elephant160v1	12477.300
isapa128av20	22486.000
grain128aead	22596.600
aes128k96n	



Cipher	F1² avg. time [μs]
xoodyakround3	64.277
schwaemm256128v2	80.914
ascon128v12	81.091
tinyjambu128v2	110.295
giftcofb128v1	131.551
romulusn1+	225.008
grain128aeadv2	241.014
aes128k96n	337.203
photonbeetleaead128rate128v1	590.958
isapa128av20	600.055
elephant160v2	4430.300

Software performance rankings
on AVR (8-bit - left) and ARM Cortex M3 (32-bit - right)
from OTH (Germany) : `lwc.las3.de/table.php`

Hardware performances of Romulus : FPGA

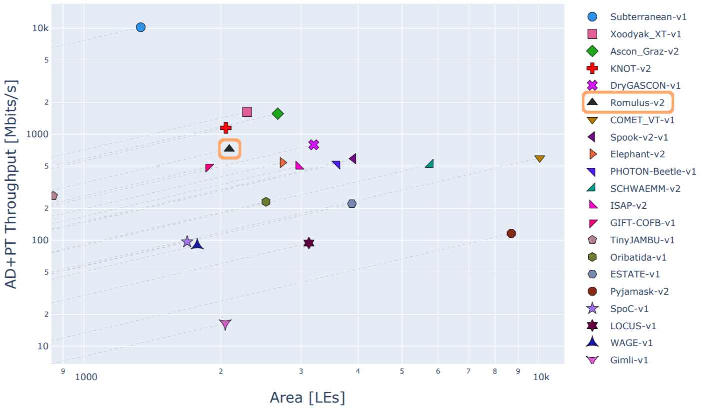
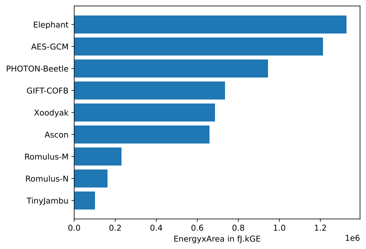
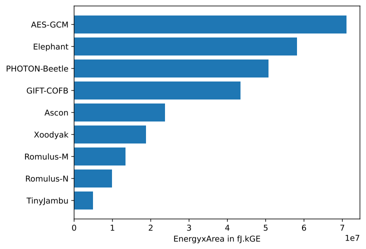


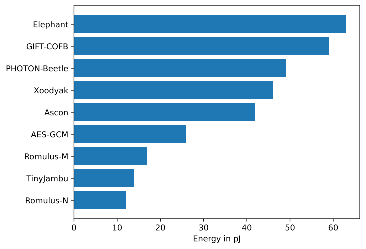
Figure 8: Cyclone-10-LP Encryption AD+PT Throughput for Long Messages vs LEs

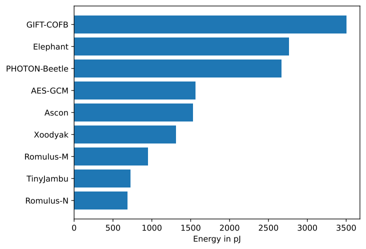
FPGA performance from GMU, USA

Hardware performances of Romulus : ASIC









ASIC performance ranking from

`https://github.com/mustafam001/lwc-aead-rtl/`

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)

Enc. of 1600 bytes of A and M using Romulus-N in different implementations.

- stands for unprotected, P for probing, NI, SNI, and C for coupling resistance

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

Features

Romulus **features :**

- provably secure in standard model (unlike most LWC candidates)
- full 128-bit security time/data (unlike some LWC 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), ISO
- easy nonce-misuse resistance mode (unlike **all** 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
- excellent hardware profile, good software profile (good for 4 or 8-bit)
- side-channel protection : efficient masking (small protected state) + Romulus-T mode protection

No TBC currently appears in NIST cryptography standards yet.

NIST Lightweight cryptography competition

The 10 finalists of the ongoing NIST competition

		SECURITY			CLAIMED FEATURES				
name	type	internal	distinguisher internal	data. sec. claims	nonce-misuse	RUP	hash	side-chan. resistance	other
ASCON	perm.	ASCON-p	yes	birthday			✓	some	CAESAR
ELEPHANT	perm.	SPONGENT	no	birthday	integrity	✓			parallel
GIFT-COFB	BC	GIFT	no	birthday					
Grain-128AEAD	SC	Grain	no	full					eSTREAM
ISAP	perm.	ASCON-p	yes	full				yes	
PHOTON-Beetle	perm.	PHOTON	no	full			✓		ISO/IEC
Romulus	TBC	SKINNY	no	full	Romulus-M/T	Romulus-M/T	✓	Romulus-T	ISO/IEC
SPARKLE	perm.	ad-hoc	no	full			✓		
TinyJambu	perm.	ad-hoc	yes	birthday					
Xoodyak	perm.	Xoodoo	yes	full			✓		

Thank you !