ESTATE: Hardware Benchmarking and Security Analysis

A.Chakraborti, N.Datta*, A.Jha*, C. Mancillas Lopez**, M.Nandi*, Y. Sasaki

NTT Secure Platform Laboratories, Japan
*Indian Statistical Institute, Kolkata, India
**CINVESTAV, Mexico

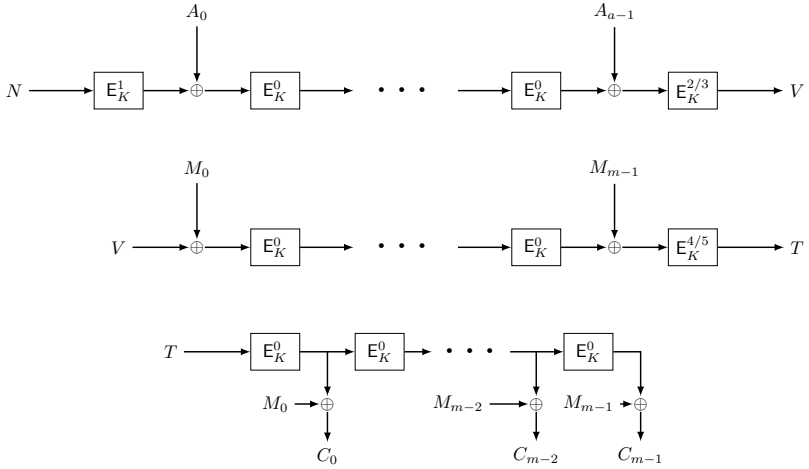
NIST Lightweight Workshop Nov 04, 2019

Motivation

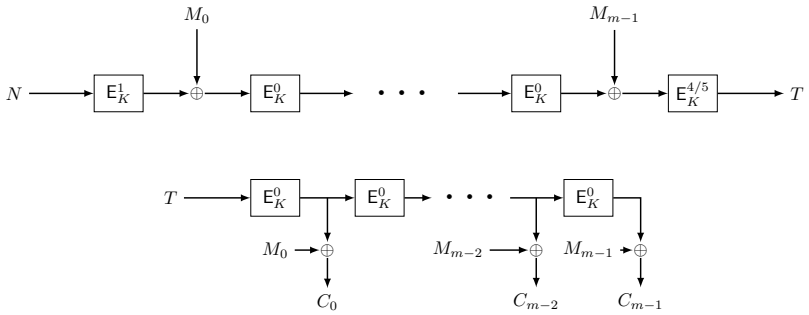
Designing Lightweight and INT-RUP Secure Authenticated Encryption with Efficiency in Short Message Processing

- Optimum state size.
- Multiplication-free.
- Optimal primitive calls.
- Nonce-Misuse resistant.
- INT-RUP secure

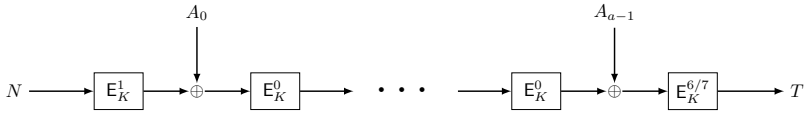
ESTATE: For *a* block AD and *m* block message



ESTATE: For empty AD and m block message



ESTATE: For *a* block AD and empty message



Design Rationale

Choice of MAC-then-Encrypt Mode

- Single-state
- Inverse free
- Nonce Misuse Resistance

Design Rationale

Why Tweakable Block Cipher?

Use short tweaks (4-bit) for domain separation:

- Type of the current data (associated data or message)
- Completeness of the final data block (partial or full)
- Emptyness of the associated data and/or message

Other Methods for Domain Separation

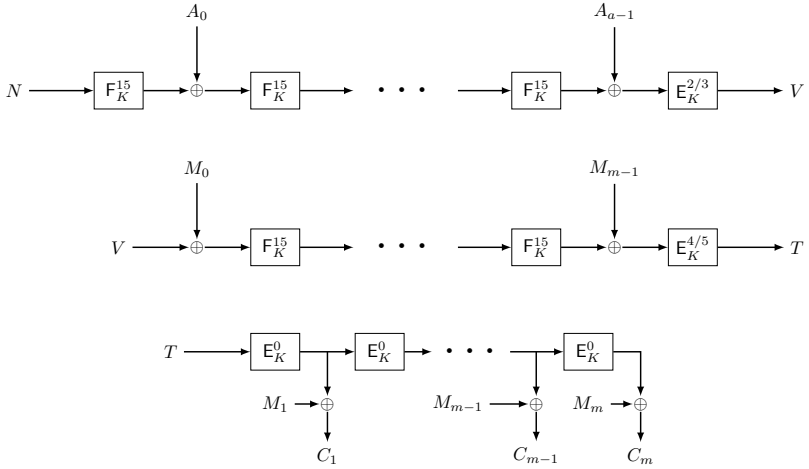
- Few constant field multiplications: increase the hardware footprint
- Additional block cipher invocations: decrease the energy efficiency and throughput for short messages.

Design Rationale

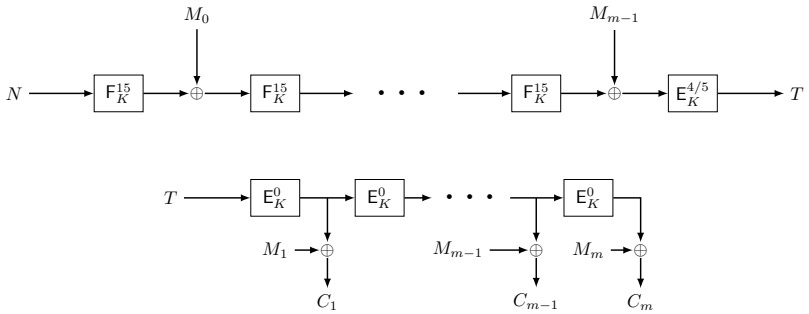
Choice of Tweaks

- Process Bulk Messages with tweak 0: Identical to block cipher.
- First Block Cipher Invocation with tweak 1: To ensures the RUP security of the mode.
- Finalize with tweaks 2-5. For the purpose of domain separation:
  - 2 and 3: full and partial final AD block processing
  - 4 and 5 full and partial final plaintext block processing.

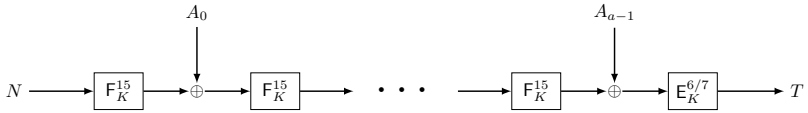
sESTATE: For *a* block AD and *m* block message



sESTATE: For empty AD and m block message



sESTATE: For *a* block AD and empty message



Tweak Choices for sESTATE

- Always use tweak 15 for F (round-reduced E) to maximize the distance with other tweaks specially tweak 0.
- Everything else are simialr as ESTATE.

Choice of E_K^t

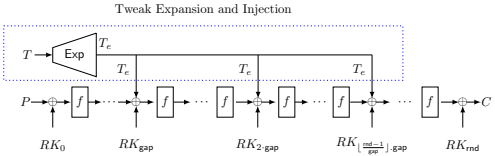


Figure: Elastic-Tweak Framework.

- BC to tBC: BC[t, t_e, tic, gap]
- Expand Tweak with high distance encoding
- Inject Tweak
- AES-128[4, 8, 8, 2] (energy efficient), GIFT-128[4, 32, 32, 5] (area efficient)

Choice of E_K^t

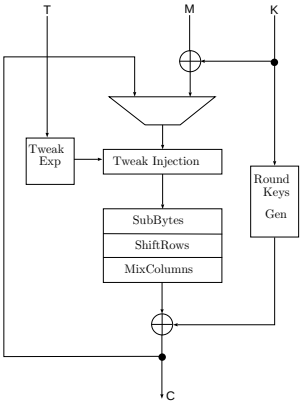


Figure: Architecture.

Choice of E_K^t

Table: Benchmark for several AES-128 based E_K^ts

tBC or TBC	LUTs	FF	Slices	Frequency (MHz)	Clock cycles	Throughput (Mbps)
AES-ED	2945	533	943	297.88	11	3466.24
AES-ED[4,8,8,2]	2960	534	1044	295.97	11	3444.01
AES-ED[8,16,8,2]	2976	534	1129	295.81	11	3442.15
AES-ED[16,32,8,2]	3006	534	1134	292.87	11	3407.94
AES-E	1605	524	559	330.52	11	3846.05
AES-E[4,8,8,2]	1617	524	574	328.27	11	3819.87
AES-E[8,16,8,2]	1632	524	593	325.17	11	3783.79
AES-E[16,32,8,2]	1659	524	592	326.56	11	3799.97

Choice of E_K^t

Table: Benchmark for several GIFT-128 based E_K^ts

tBC or TBC	LUTs	FF	Slices	Frequency	Clock	Throughput
GIFT-64-ED	615	277	236	455.17	29	1004.51
GIFT-64-ED[4,16,16,4]	617	277	234	430.29	29	946.60
GIFT-64-E	449	275	153	596.66	29	1316.77
GIFT-64-E[4,16,16,4]	479	275	179	595.09	29	1313.30
GIFT-128-ED	1113	408	432	447.83	41	1398.10
GIFT-128-ED[4,32,32,5]	1158	408	419	416.50	41	1300.29
GIFT-128-ED[16,32,32,4]	1223	408	428	429.32	41	1340.31
GIFT-128-E	763	403	330	596.30	41	1861.62
GIFT-128-E[4,32,32,5]	796	403	332	597.59	41	1865.65
GIFT-128-E[16,32,32,4]	805	403	377	598.78	41	1869.36

Comparative Study of SIV based Submissions

Submission	Primitive	State size (bits)	Optimality	INT-RUP	Mult-free
ESTATE	tBC-128/128/4	260	✓	✓	✓
SUNDAE-GIFT	BC-128/128	256	×	×	×
Limdolen	BC-128/128	384	×	×	×
SIV-Rijndael256	tBC-256/128/4	388	✓	✓	✓
SIV-TEM-PHOTON	TBC-256/128/132	516	✓	✓	✓
TRIFLE	BC-128/128	384	×	×	×

- Only SUNDAE and ESTATE are surviving in the competition
- ESTATE has advantage over SUNDAE with negligible (4-bit) increase in the state size

Comparative Study with SUNDAE

Number of Primitive Call

- SUNDAE makes one additional primitive call with constant value for domain separation of emptyness of data.
- ESTATE controls them using tweaks in tBC.
- Number of primitive calls: ESTATE - a+2m (optimal), SUNDAE - a+2m+1.
- Efficiency in short message processing.

Table: Throughput Comparison for Short Message Processing

	AES-SUNDAE					ESTATE-AES				
Msg Len (bytes)	16	32	64	128	2048	16	32	64	128	2048
Cycles	41	61	101	181	2581	31	51	91	171	2571
Mbps	945.36	1270.81	1535.04	1713.13	1922.21	1251.10	1520.94	1704.79	1814.46	1930.90

Comparative Study with SUNDAE

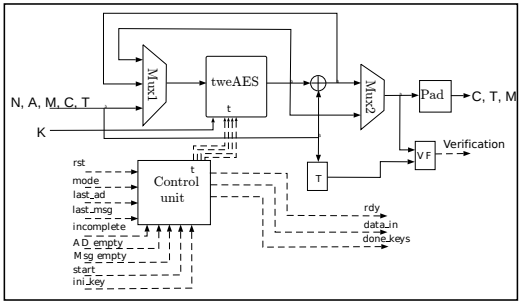
Constant Multiplication

- SUNDAE requires multip[lications by 2 and/or 2^2 for domain separation of partial/full data blocks.
- ESTATE controls them using tweaks in tBC.
- Efficiency in hardware footprint.

INT-RUP Security

- SUNDAE is insecure against INT-RUP attacks.
- Clever choice of tweaks ensure INT-RUP security of ESTATE.

Hardware Architecture for ESTATE



Hardware Comparison with SUNDAE

Scheme	# LUTs	# Slices	Frequency (MHZ)	Throughput (Gbps)	Mbps/ LUT	Mbps/ Slice
ESTATE-AES	1901	602	303.00	1.94	1.02	3.22
sESTATE-AES	1903	602	302.20	2.42	1.27	4.02
ESTATE-GIFT	681	263	526.00	0.84	1.23	3.20
AES-SUNDAE	1922	614	302.81	1.93	1.01	3.16
GIFT-SUNDAE	931	310	526.03	0.84	0.90	2.71

Benchmarking ESTATE

Scheme	Underlying Primitive	# LUTs	# Slices	Gbps	Mbps/ LUT	Mbps/ Slice
ESTATE-AES (32-bit datapath Implementation)	tBC	376	124	0.393	1.05	3.17
ESTATE-AES	tBC	1901	602	1.94	1.02	3.22
sESTATE-AES	tBC	1903	602	2.42	1.27	4.02
ESTATE-GIFT	tBC (non AES)	681	263	0.84	1.23	3.20
AES-OTR	BC	4263	1204	3.187	0.748	2.647
AES-OCB	BC	4269	1228	3.608	0.845	2.889
AES-COPA	BC	7795	2221	2.770	0.355	1.247
AES-GCM	BC	3478	949	3.837	1.103	4.043
CLOC-AES	BC	3552	1087	3.252	0.478	1.561
CLOC-TWINE	BC (non AES)	1552	439	0.432	0.278	0.984
SILC-AES	BC	3040	910	4.365	1.436	4.796
SILC-LED	BC (non AES)	1682	524	0.267	0.159	0.510
SILC-PRESENT	BC (non AES)	1514	484	0.479	0.316	0.990
ELmD	BC	4490	1306	4.025	0.896	3.082
JAMBU-AES	BC	1595	457	1.824	1.144	3.991
JAMBU-SIMON	BC (non AES)	1200	419	0.368	0.307	0.878
COFB-AES	BC	1456	555	2.820	2.220	5.080
SAEB	BC	348	–	–	–	–
AEGIS	BC-RF	7504	1983	94.208	12.554	47.508
DEOXYS	TBC	3234	954	1.472	0.455	2.981
Beetle[Light+]	Sponge	608	312	2.095	3.445	6.715
Beetle[Secure+]	Sponge	1101	512	2.993	2.718	5.846
ASCON-128	Sponge	1373	401	3.852	2.806	9.606
Ketje-Jr	Sponge	1567	518	4.080	2.604	7.876
NORX	Sponge	2881	857	10.328	3.585	12.051
PRIMATES-HANUMAN	Sponge	1148	370	1.072	0.934	2.897
ACORN	Stream cipher	499	155	3.437	6.888	22.174
TriviA-ck	Stream cipher	2221	684	14.852	6.687	21.713

Security

Security Statement for ESTATE

$$\mathbf{Adv}^{AE}_{ESTATE[\widetilde{E}]}(t, q, \ell, \sigma) \leq \mathbf{tprp}_{\widetilde{E}}(t', \sigma) + O\left(\frac{\sigma^2}{2^n} + \frac{q_d}{2^n}\right), \tag{1}$$

where t, q, ℓ, σ denote the computational time, query bound, maximum query length, and the total number of tweakable block cipher calls across all encryption and decryption queries, respectively.

On the Security (RUP) of ESTATE

- Tweak values for the first block cipher call in tag generation and encryption phases are always distinct.
- This ensures that release of internal state information in the encryption phase gives no information of any internal state of tag generation phase.
- For any forgery, adversary has to guess the output of a PRF, which is possible with at most $O(1/2^n)$ probability.
- This gives an INT-RUP bound of the form $O(\sigma^2/2^n + q_d/2^n)$, where
  - $O(\sigma^2/2^n)$ is due to the PRF security of the tag generation phase, and
  - $O(q_d/2^n)$ is due to the forgery attempt where q_d denotes the number of forgery attempts.

Security of the Recommended Instantiations

- We consider nonce-misuse adversaries.
- We claim integrity security even under the INT-RUP model.

Table: Summary of security claims for recommended instantiations. The data and time limits indicate the amount of data or time required to make the attack advantage close to 1.

Submissions	**Privacy**		**Integrity**	
	Time	Data (in bytes)	Time	Data (in bytes)
ESTATE-AES	2^{128}	2^{64}	2^{128}	2^{64}
ESTATE-GIFT	2^{128}	2^{64}	2^{128}	2^{64}

Thank you