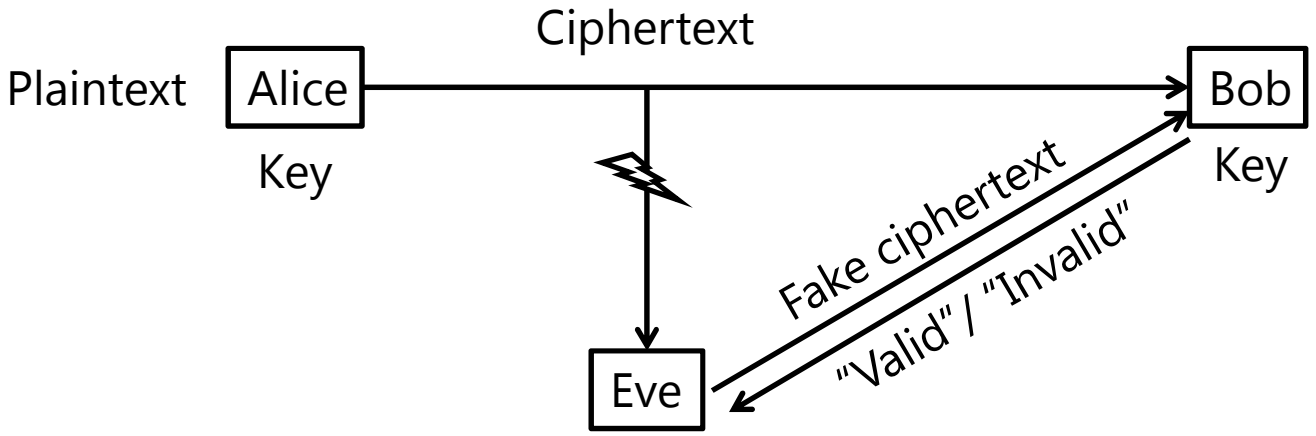
Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions

Kazuhiko Minematsu
NEC Corporation

Eurocrypt 2014, May 13 2014, Copenhagen, Denmark

Authenticated Encryption (AE)

- Symmetric-key function doing encryption & authentication
- Security goal : protect plaintext from eavesdropping and detect ciphertext tampering



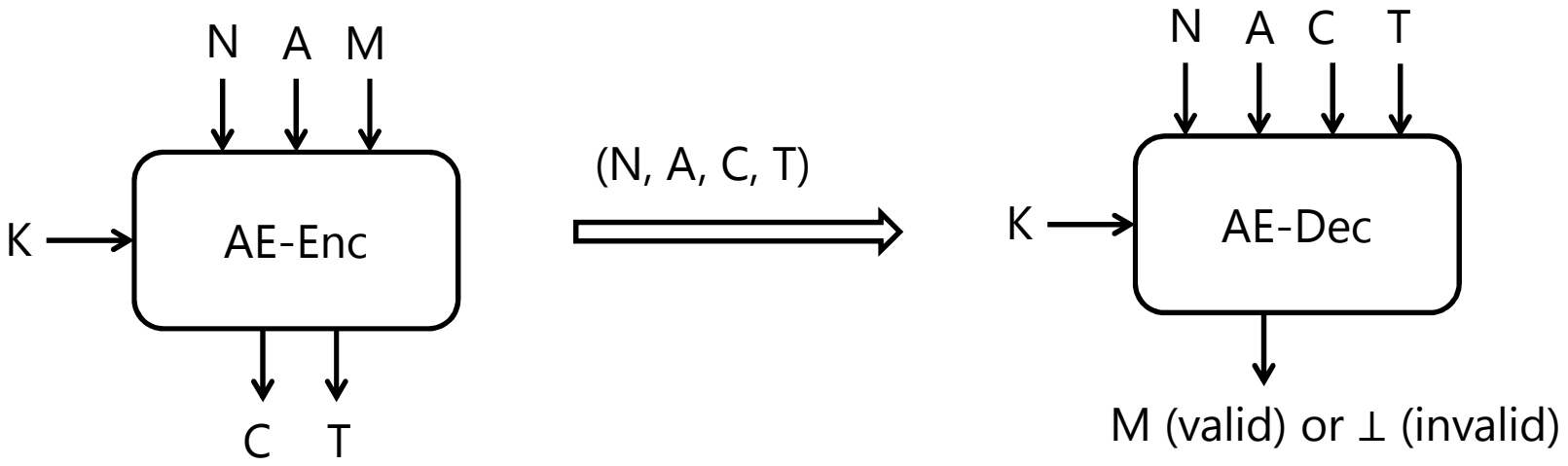
AE is (going to be) everywhere

- Internet protocols (e.g. SSL/TLS)
- Mobile
- Storage
- Satellite
- Sensors, plants, cars, …

- An old problem, still active research area
- Cryptographic competition on AE (CAESAR) started

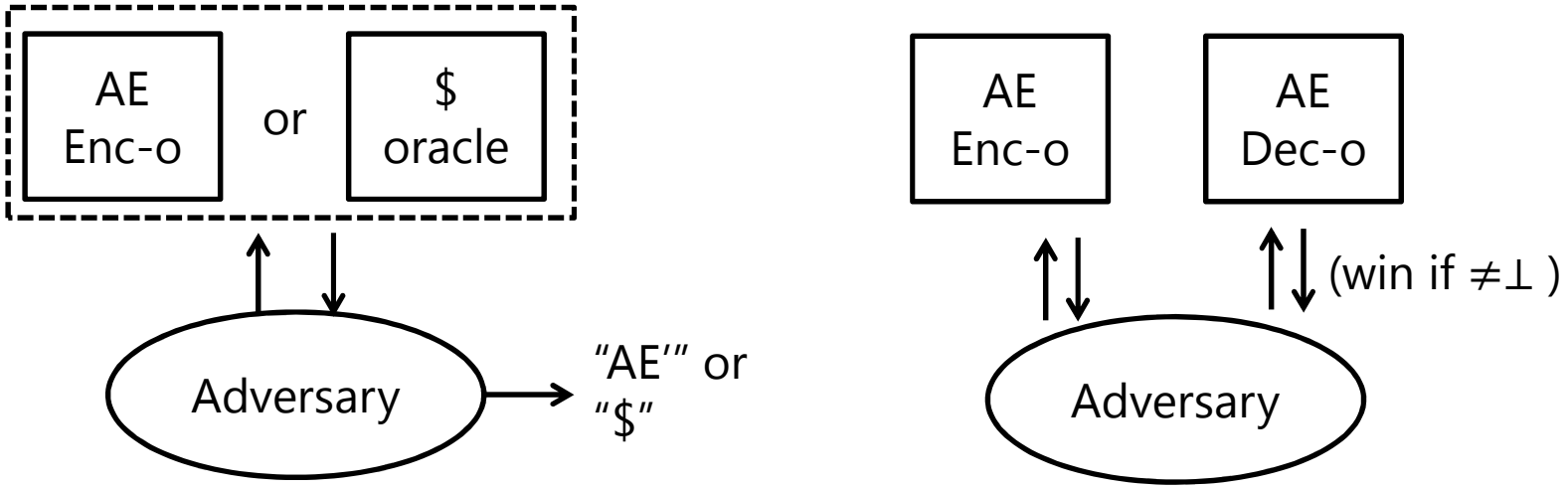
Definition

- Nonce-based AE
  - Nonce : unique for each encryption (e.g. counter)
  - Associated data (AD) : data sent w/o encryption, but authentication
    - AE w/ AD is also called AEAD
- Six variables: Key (K), Nonce (N), AD (A), Plaintext (M), Ciphertext (C), and Tag (T)
- AE-Enc takes (N,A,M) to produce (C,T) w/ |M|=|C|
- AE-Dec takes (N,A,C,T) to produce M if valid, ⊥ (default error symbol) if invalid



Two security notions

- Privacy (PRIV) : ciphertexts are hard to distinguish from random sequences
  - Distinguish two oracles, AE-Enc and random ($)
- Authenticity (AUTH) : a successful forgery of ciphertext is hard
  - Successful forgery = receiving a (non-trivial) "valid" response from Dec-oracle of AE



How can we build AE ?

- Generic composition
- Nonce-based Encryption + MAC (message authentication code) basically works
- If we focus on blockcipher (BC)-based schemes, an example is CTR encryption + CMAC, using two keys
- Security analyzed [BN00][K00] [NRS14]
- Limitation : rate is 2 (two rate-1 functions)
  - rate = # of BC calls par input block

[BN00] M. Bellare, C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. ASIACRYPT 2000.
[K00] H. Krawczyk: The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). CRYPTO 2001
[NRS14] C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering Generic Composition, Eurocrypt 2014

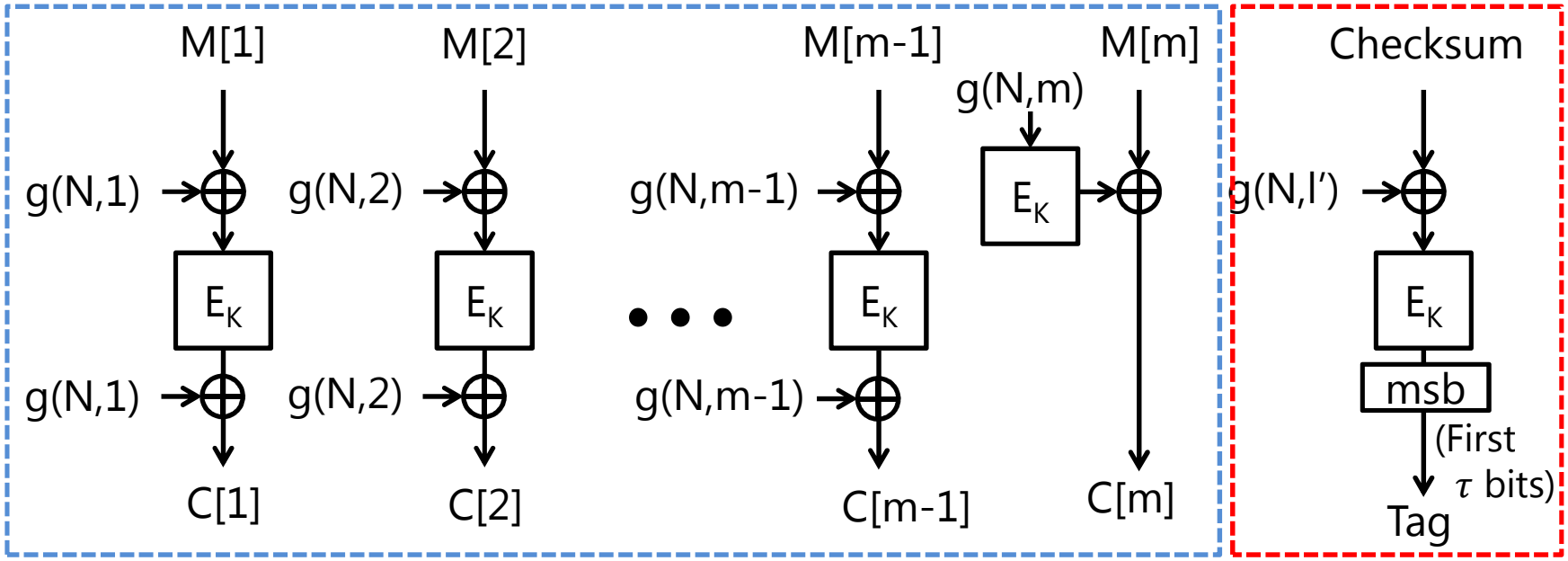
Can we go further?

- Rate-1 AE by integration of Enc and MAC
- Many early attempts broken (~'90)
- Right solutions appeared around 2000
  - IACBC, IAPM [J01], XCBC [GD01]
  - OCB [RBB03] [R04][KR11]

[GD01] V.D. Gligor and P. Donescu. Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. FSE 2001
[Ju01] C. Jutla Encryption Modes with Almost Free Message Integrity. EUROCRYPT 2001
[Ro04] Rogaway : Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. ASIACRYPT 2004
[RoBeBl03] Rogaway, Bellare, Black, : OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inf. Syst. Secur. 6(3) (2003)
[KrRo11] Krovetz, Rogaway : The Software Performance of Authenticated-Encryption Modes. FSE 2011

Structure of OCB (w/o AD)

- Enc = ECB mode with tweakable BC (TBC) [LRW02]
  - TBC = BC taking tweaks, (N,1), (N,2), …
  - Realized by BC w/ I/O masks (called XE mode [R04])
  - Mask g(*) : a function of Nonce, block index, and key
- MAC = Plaintext checksum (XOR) encryption



Checksum= M[1] ⊕ M[2] … ⊕ M[m]

[LRW02] M. Liskov, R. Rivest, D. Wagner. Tweakable Block Ciphers. CRYPTO 2002

OCB

- Many good properties
  - Rate-1
    - mask generation can be done with few BC calls (usually one)
  - Parallelizable (for E & D)
  - On-line
    - operation can start w/o knowing the input length
  - Provably secure if BC is a strong pseudorandom permutation (SPRP)*

*[AY13] showed a relaxation from SPRP

- So, can't we go further ?

[AY13] K.Aoki, K. Yasuda: The Security of the OCB Mode of Operation without the SPRP Assumption, ProvSec 2013

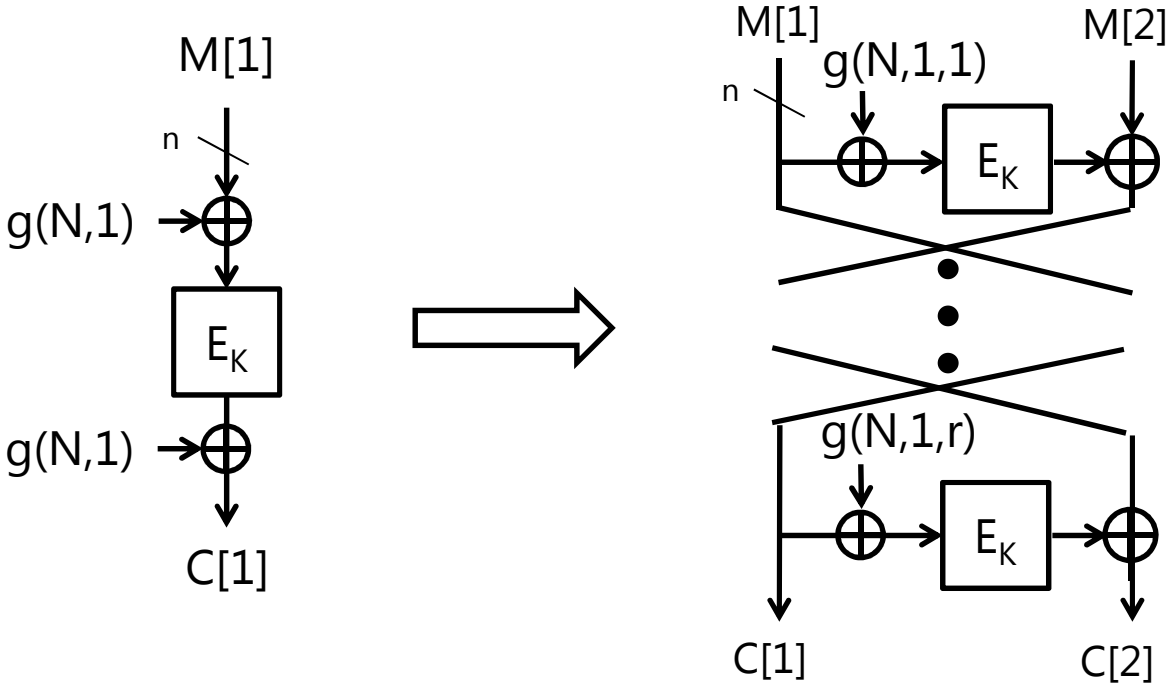
Existence of Blockcipher Inverse

- One potential disadvantage of OCB: the existence of BC inverse (decryption function)
  - Popular rate-2 modes use only the forward (encryption) function of BC, i.e. inverse-free
- Undesirable in some cases
  - Increased size (Sw, Hw)
  - BC inverse may be slower than forward (or vice versa)
    - E.g. Byte-wise Sw AES on microcontrollers
  - Stronger security assumption (SPRP rather than PRP/PRF)

- **Can we remove BC inverse ?**

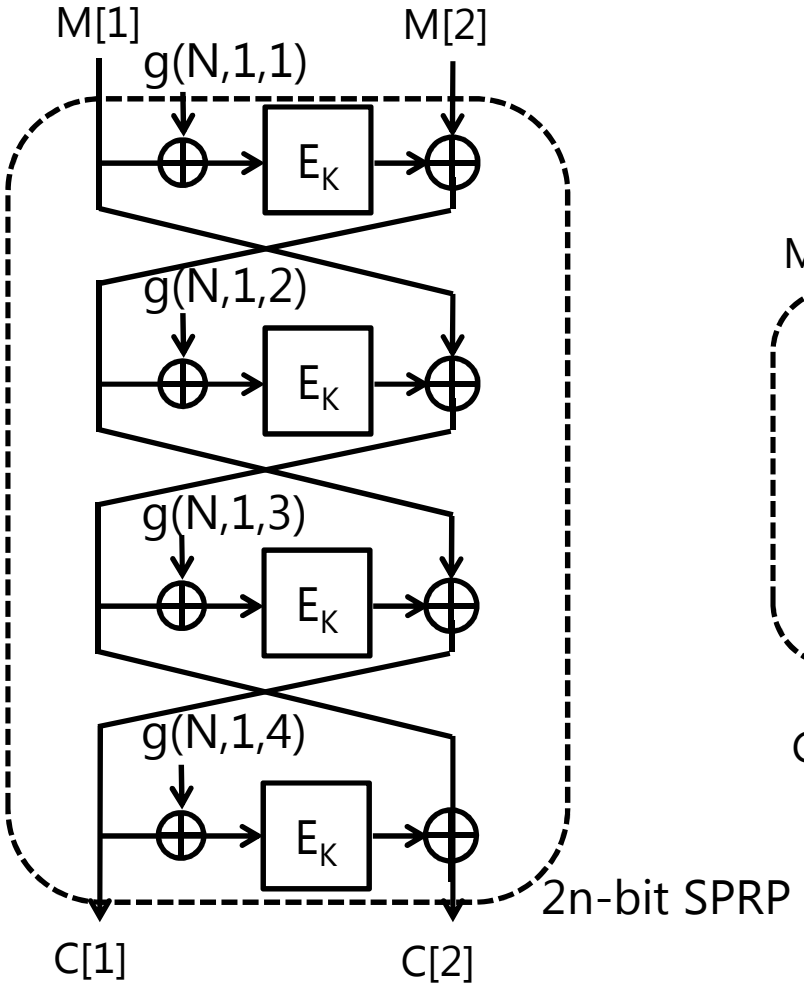
Using Feistel rounds

- Substituting n-bit TBC with 2n-bit balanced Feistel permutation
  - Round function = n-bit TBC built from n-bit BC
    - forward function, with input mask
    - Tweak consists of Nonce, block index, and round index
- How many rounds are needed?

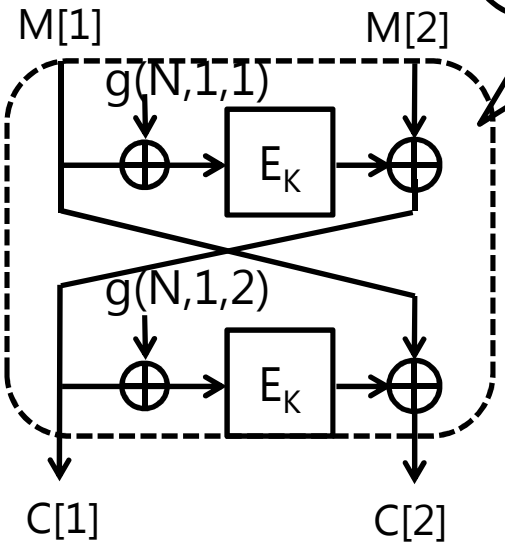


Using Feistel rounds (Contd.)

- 4 rounds are sufficient, as it is 2n-bit SPRP (Luby-Rackoff), but rate-2, no gain
- To keep rate-1, we have to use 2 rounds



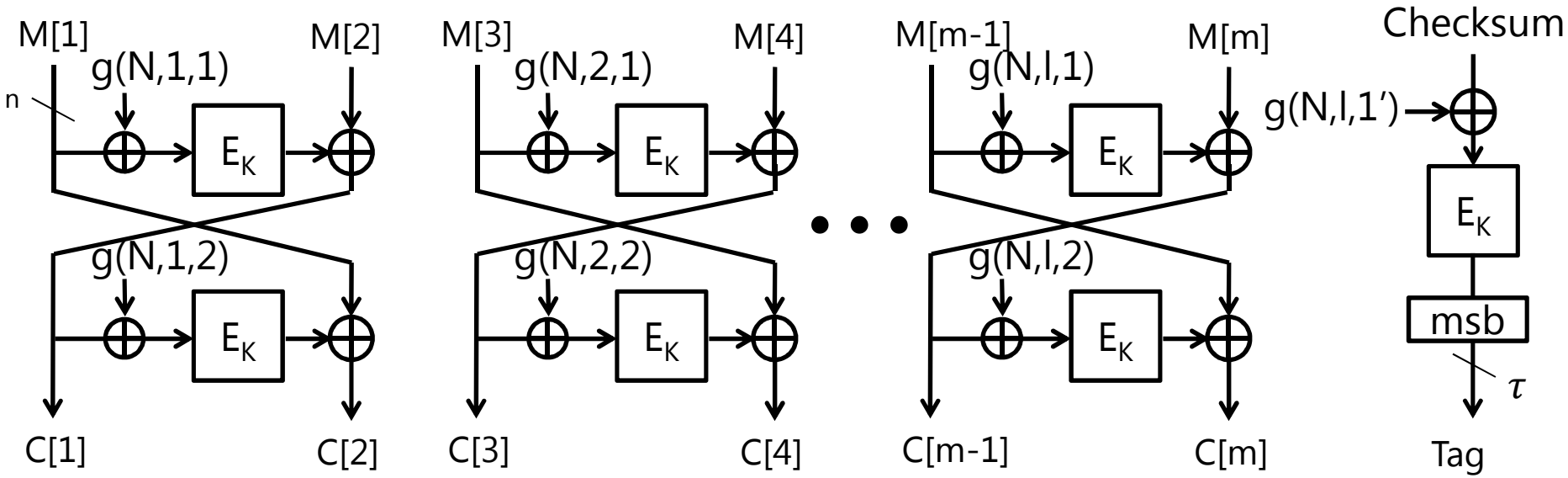
2n-bit SPRP



2-R is not even PRP, so we can not directly follow the proof of OCB

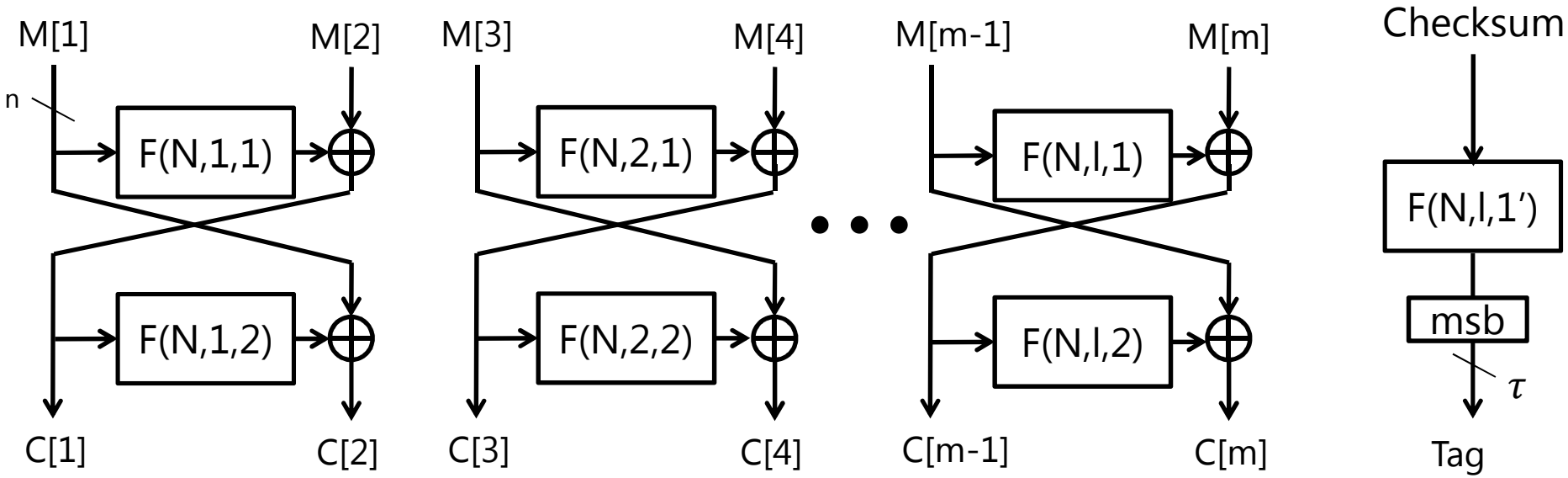
2-round AE construction

- We use 2n-bit 2-R Feistel permutation instead of OCB's n-bit TBC
- n-bit checksum needs to be defined (later)
- Inverse-free, rate-1



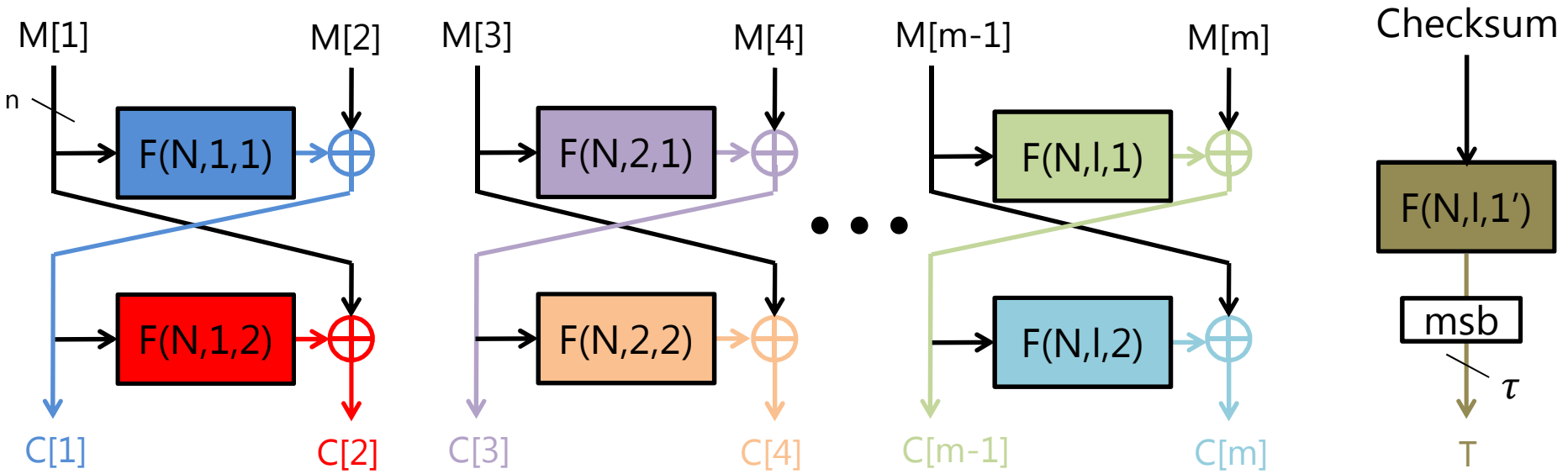
2-round AE skeleton

- We can safely assume internal TBCs are independent random functions indexed by tweak
  - if masks are properly chosen (differentially uniform [LRW02])
- The scheme is called 2-R AE *skeleton*
- We analyze PRIV and AUTH of 2-R AE skeleton



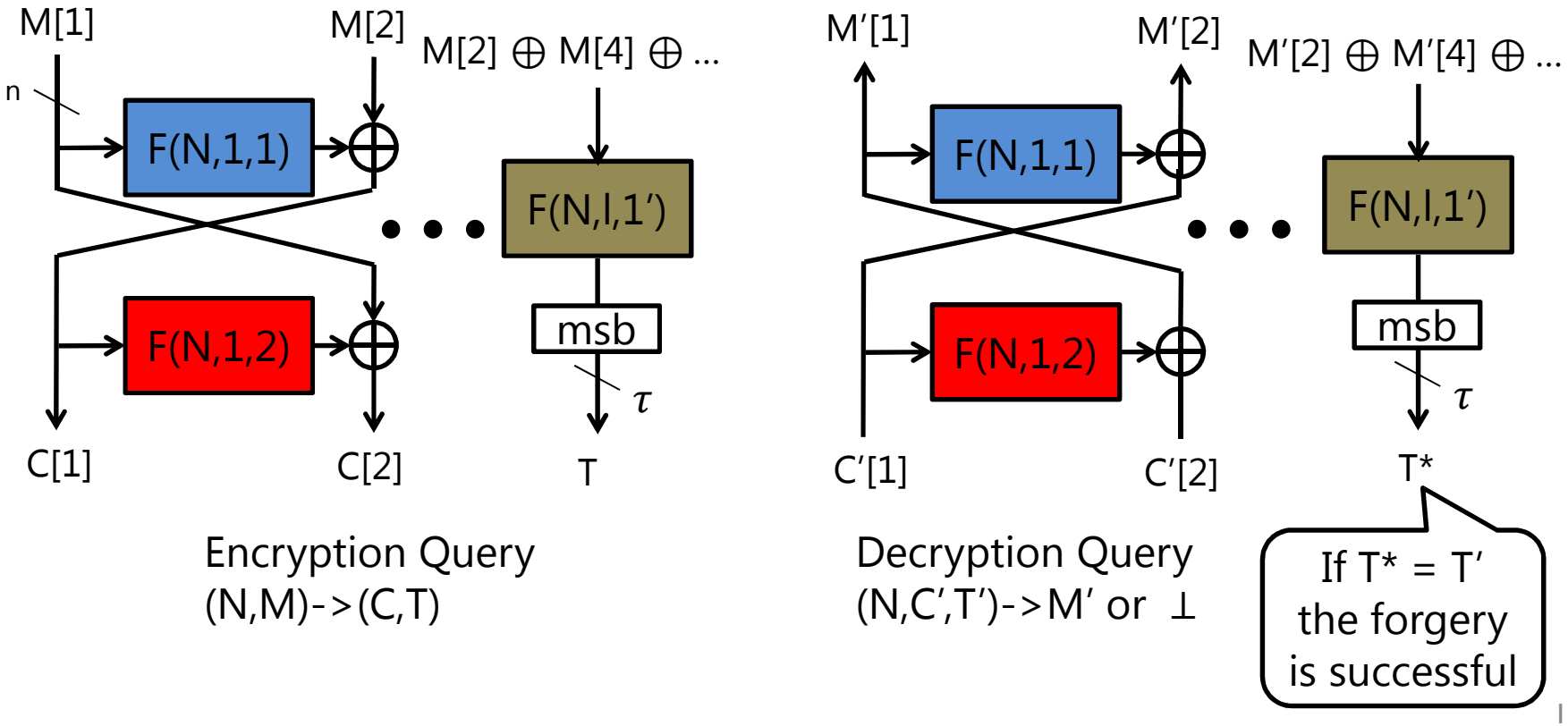
Privacy of 2-round AE skeleton

- Each C[i] contains an output of RF invoked only once (as Nonce is unique)
- Ciphertext and tag are uniformly random
- **PRIV bound is zero**



Authenticity of 2-round AE skeleton

- Now checksum is defined as a sum of **even** plaintext blocks
- Consider simple attack using one encryption query and one decryption query
- Forgery is successful iff T* (true tag for dec query) = T′ (fake tag)
- Suppose (C[1],C[2]) was changed to (C′[1], C′[2]) and N was not changed

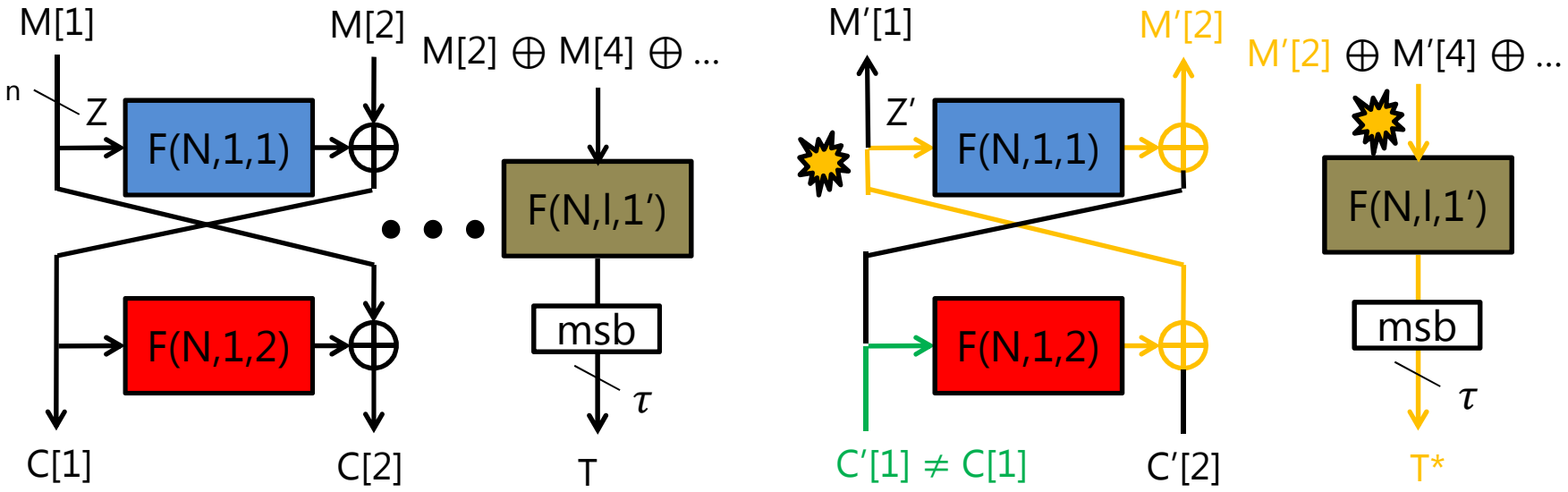


Encryption Query
(N,M)->(C,T)

Decryption Query
(N,C′,T′)->M′ or ⊥

Authenticity of 2-R AE skeleton (Contd.)

- Case C'[1] ≠ C[1] :
- Then the first round input (Z') is random -> M'[2] is random, **unless the collision between Z and Z'**
- If M'[2] is random, then checksum is random -> T* is random, **unless the checksum collision**
- Two collision events of prob. $1/2^n$
- If T* is random, the chance of guessing T* is $1/2^\tau$, for τ-bit T*
- **-> AUTH bound is $2/2^n + 1/2^\tau$**

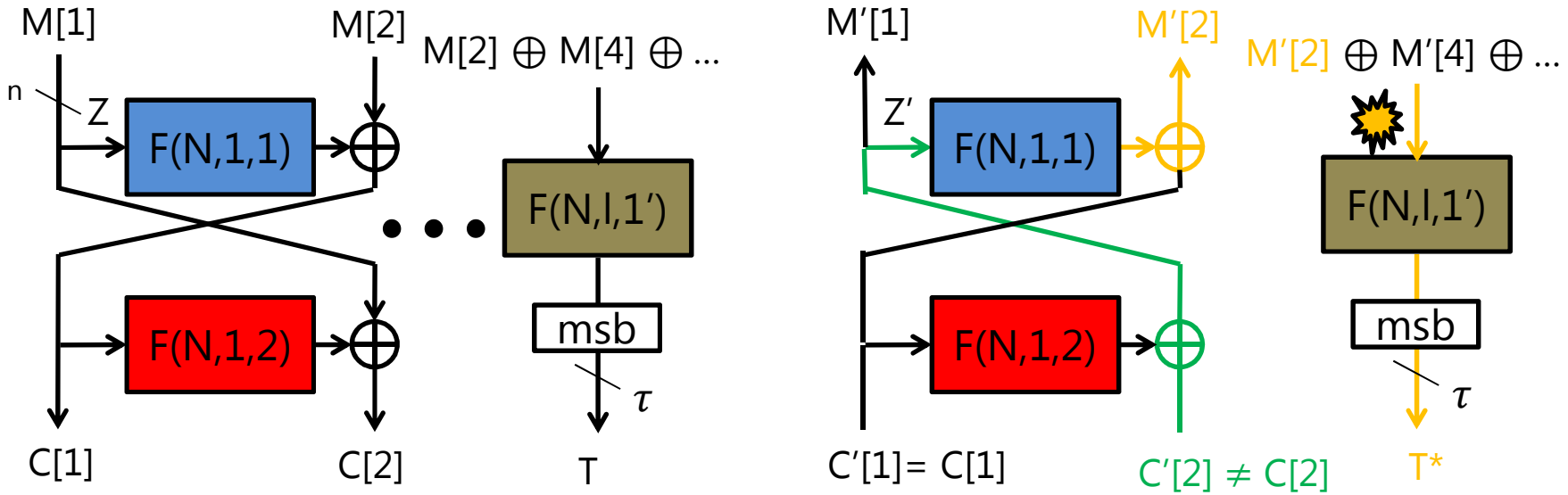


Encryption Query
(N,M)->(C,T)

Decryption Query
(N,C',T')->M' or ⊥

Authenticity of 2-R AE skeleton (Contd.)

- Case C′[1] = C[1], C′[2] ≠ C[2] can be handled similarly, yielding a smaller probability
- AUTH is bounded by $\mathbf{2/2^n + 1/2^\tau}$, for single dec query
  - The bound for multiple dec queries is derived using [BGM04]
- 2-R Feistel actually works



Encryption Query
(N,M)->(C,T)

Decryption Query
(N,C′,T′)->M′ or ⊥

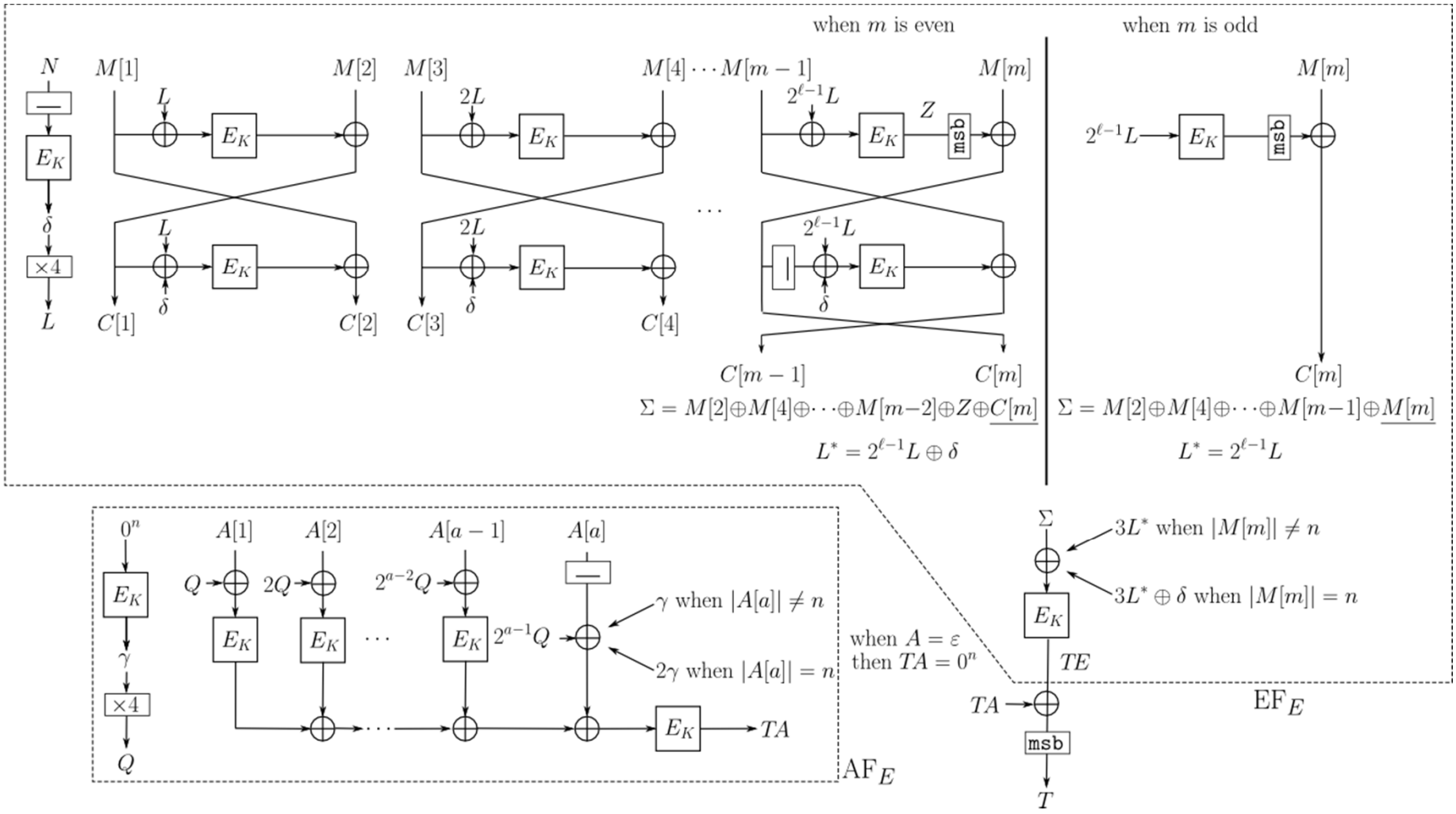
[BGM04] M. Bellare, O. Goldreich, A. Mityagin. The Power of Verification Queries in Message Authentication and Authenticated Encryption. ePrint 2004

OTR

- OTR (Offset Two-Round) : a concrete instantiation of 2-R AE skeleton using a BC
- A mode like OCB but without BC inverse
- Some details:
  - Mask generation is based on constant-multiplication over GF(2^n) (GF doubling)
    - Similar to many BC modes
  - AD is processed by a PRF like PMAC [R04]
- Surprisingly simple idea
  - The idea of using Feistel rounds was described at ManTiCore papers [ABDST04-1][ABDST04-2], while OTR is an independent work
- AES-OTR submitted to CAESAR

[ABDST04-1] E. Anderson, C. Beaver, T. Draelos, R. Schroeppel, M. Torgerson. ManTiCore: Encryption with Joint Cipher-State Authentication. ACISP 2004
[ABDST04-2]. Anderson, C. Beaver, T. Draelos, R. Schroeppel, M. Torgerson. Manticore and CS mode: parallelizable encryption with joint Cipher-State authentication (2004)

Encryption of OTR



Properties of OTR

- Mostly keeping OCB's good properties
  - Rate-1
  - Parallelizable (for E & D)
  - On-line
    - under two-block partition, more restrictive than OCB
  - Provably secure if BC is a PRP (or PRF)
- And inverse-free

Table 1. A comparison of AE modes. Calls denotes the number of calls for m-block message and a-block header and one-block nonce, without constants.

Mode	Calls	On-line	Parallel	Primitive
CCM [3]	$a + 2m$	no	no	E
GCM [5]	m [E] and $a + m$ [Mul]	yes	yes	$E, \text{Mul}^{\dagger}$
EAX [16]	$a + 2m$	yes	no	E
OCB [32, 43, 46]	$a + m$	yes	yes	E, E^{-1}
CCFB [35]	$a + cm$ for some $1 < c^{\ddagger}$	yes	no	E
OTR	$a + m$	yes¶	yes¶	E

† $\text{GF}(2^n)$ multiplication
‡ Security degrades as c approaches 1
¶ two-block partition

Comparison of AE modes

Security bounds

- Combine the bounds of 2-R skeleton w/ TBC's security bounds [R04]
- Standard birthday-type bounds
  - We need about $2^{n/2}$ data blocks to break OTR

Privacy

Theorem 1. *Fix $\tau \in \{1, \ldots, n\}$. For any PRIV-adversary $\mathcal{A}$ with parameter (q, σ_A, σ_M),*

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) \leq \frac{6\sigma^2_{\mathtt{priv}}}{2^n}$$

holds for $\sigma_{\mathtt{priv}} = q + \sigma_A + \sigma_M$.

Authenticity

Theorem 2. *Fix $\tau \in \{1, \ldots, n\}$. For any AUTH-adversary $\mathcal{A}$ with parameter $(q, q_v, \sigma_A, \sigma_M, \sigma_{A'}, \sigma_{C'})$,*

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathrm{OTR}[\mathsf{P},\tau]}(\mathcal{A}) \leq \frac{6\sigma^2_{\mathtt{auth}}}{2^n} + \frac{q_v}{2^\tau}$$

holds for $\sigma_{\mathtt{auth}} = q + q_v + \sigma_A + \sigma_M + \sigma_{A'} + \sigma_{C'}$.

(Toy) Software Implementations

1. Naïve C-code of OTR and OCB(2), using AES w/ 4Kb table (called T-table), run on x86 PC

- Both have similar speed (20~25 cycles/byte), but OTR has a smaller binary object than OCB (50~60 %)
  - Due to the absence of AES inverse

2. Simple substitution of T-table AES w/ AESNI (single block) resulted in ~2 cycles/byte for long inputs for OTR and OCB2

  - OTR is slight slower, as expected (2-R Feistel is more complex than ECB)
- Optimized AESNI codes? Not yet, see [BLT14] instead
  - (third-party implementations are always welcome!)

[BLT14] A. Bogdanov, M. Lauridsen, E. Tischhauser. AES-Based Authenticated Encryption Modes in Parallel High-Performance Software. ePrint 2014

Conclusions

- OTR : parallelizable, rate-1 AE w/o BC inverse
- An alternative to OCB if using BC inverse is undesirable
  - E.g. when space is precious (constrained devices, hardware)
  - Not a ultimate substitute
- Limitations (as OCB):
  - No protection against nonce-reusing (for encryption)
    - ask other functions for such cases
  - Birthday-bound security
- Future topics
  - Optimized implementations (Sw, Hw)
  - Explore the power of (2 or more) Feistel rounds in other applications

Thank you !

Toy Sw Implementation 1

- Naïve C-code of OTR, with AES using 4Kb table (T-table) , on a standard x86 PC
- OCB2 is also implemented using the same AES and components (doublings etc.)
- Expectation : OTR/OCB have similar speed, OTR has a smaller size (binary object) than OCB
- The results are mostly as expected (40~50 % size reduction)

Table 2. Reference implementation results of OTR and OCB2. (Upper) Speed in cycles per byte. (Lower) Object size in Kbyte.

	x86			ARM
Algorithm	VC12(32-bit)	VC12(64-bit)	gcc 4.7.1(32-bit)	gcc 4.7.3
OTR Enc	27.59	18.94	22.02	69.88
OTR Dec	27.56	18.99	22.2	69.78
OCB2 Enc	27.38	19.93	22.69	71.22
OCB2 Dec	30.86	25.43	34.29	76.16
AES Enc	18.29	12.98	15.9	54.38
AES Dec	22.28	18.36	26.64	58.14

	x86			ARM
Object	VC12(32-bit)	VC12(64-bit)	gcc 4.7.1(32-bit)	gcc 4.7.3
OTR.o	19.9	21.3	5.4	5.9
OCB2.o	20.5	21.7	4.6	5.3
AES_Enc.o	20.2	20.7	6.7	7.1
AES_EncDec.o	45.4	46.2	17.3	17.9
OTR Total	40.1	42.0	12.1	13.0
OCB2 Total	65.9	67.9	21.9	23.2

Note : our AES dec was slightly slower than enc, resulting in slower OCB-dec than others.
Accidental, not always true to T-table AES.

Toy Sw Implementation 2

- We then simply substituted T-table AES with AES instruction (AESNI)
  - with SIMD codes for some subroutines
- Results: OTR and OCB achieve ~2 cycles/byte (cpb) for long messages
  - Something unexpected (at least to me) : AESNI in single block has ~4.5 cpb
    - The power of AESNI parallelism
  - OTR is slight slower, as expected (2-R Feistel is more complex than ECB)

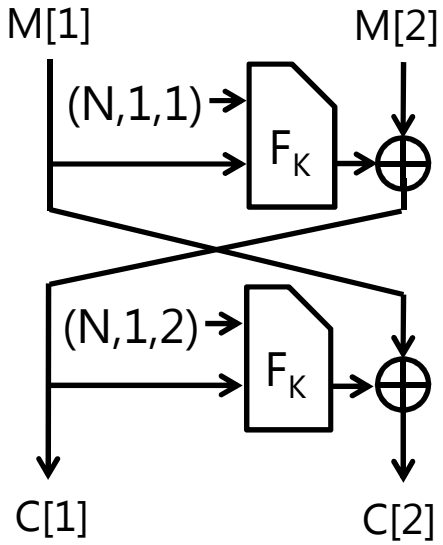
Table 3. Performance of codes with single-block AES routine using AES-NI. Data `x` denotes the plaintext length in bytes, and `a`/`b` denotes `a` (`b`) cycles per byte in 32-bit (64-bit) VC12 compilation.

Data (byte)	128	512	1024	2048	4096
OTR Enc	6.01/5.43	3.32/3.16	2.85/2.74	2.66/2.51	2.49/2.40
OTR Dec	7.22/5.60	3.81/3.15	3.06/2.72	2.79/2.51	2.59/2.39
OCB2 Enc	6.39/5.60	3.26/2.76	2.81/2.26	2.53/2.02	2.37/1.90
OCB2 Dec	6.36/5.86	3.04/2.80	2.59/2.26	2.28/2.03	2.11/1.91

[BLT14] A. Bogdanov, M. Lauridsen, E. Tischhauser. AES-Based Authenticated Encryption Modes in Parallel High-Performance Software. ePrint 2014

Other instantiations

- We can also use non-invertible primitives
  - Compression function of SHA-1/2
  - Full-scratch PRF (e.g. SipHash [AB12])
- If output is n-bit and input is something longer than n (to take N and index), skeleton is directly instantiated by prepending, no need to use input masks
  - Resulting security bounds will be those of skeleton
  - Roughly, perfect privacy & n-bit authenticity



[AB12]J.P. Aumasson, D.J. Bernstein. SipHash: A Fast Short-Input PRF. INDOCRYPT 2012