Updates on CLOC and SILC

Tetsu Iwata*, Kazuhiko Minematsu, Jian Guo,
Sumio Morioka, and Eita Kobayashi

DIAC 2015

September 28, 2015, Singapore

* Supported in part by JSPS KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 26280045

CLOC and SILC

- CLOC
  - Compact Low-Overhead CFB, FSE 2014
  - Improves CCM, EAX, and EAX-prime in terms of
    - implementation overhead beyond the blockcipher
    - precomputation complexity
    - memory requirement
  - Suitable for handling short input data on small microprocessors
- SILC
  - SImple Lightweight CFB, DIAC 2014
  - Hardware oriented version of CLOC

Agenda

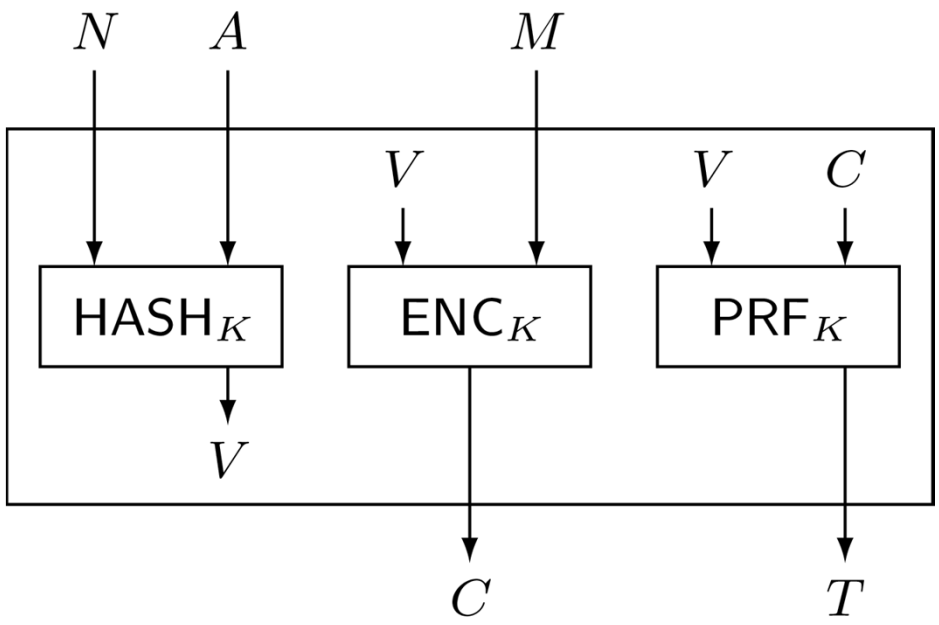
- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

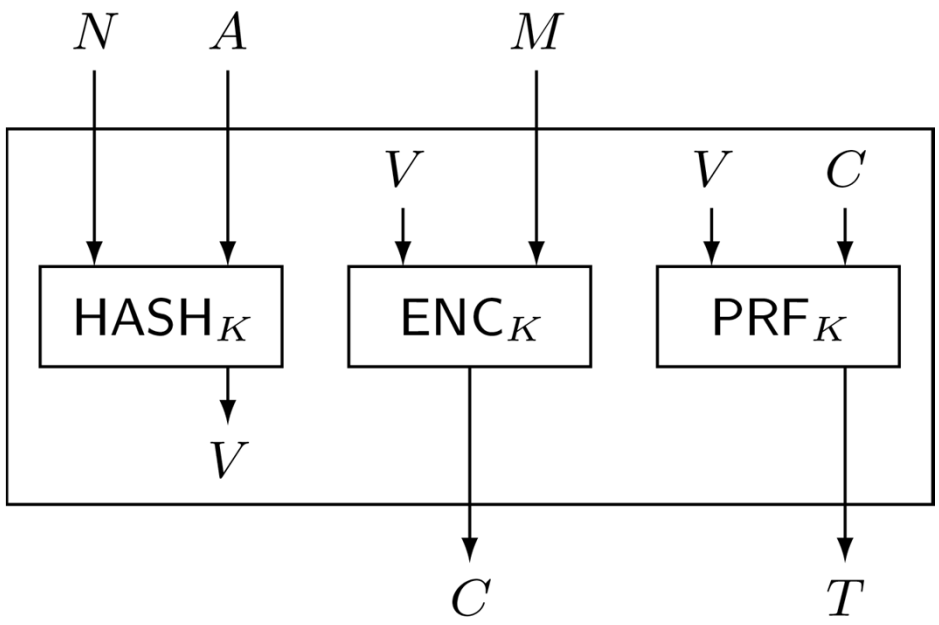
Overview of CLOC and SILC

- HASH and PRF: variants of CBC-MAC
- ENC: variant of CFB encryption mode

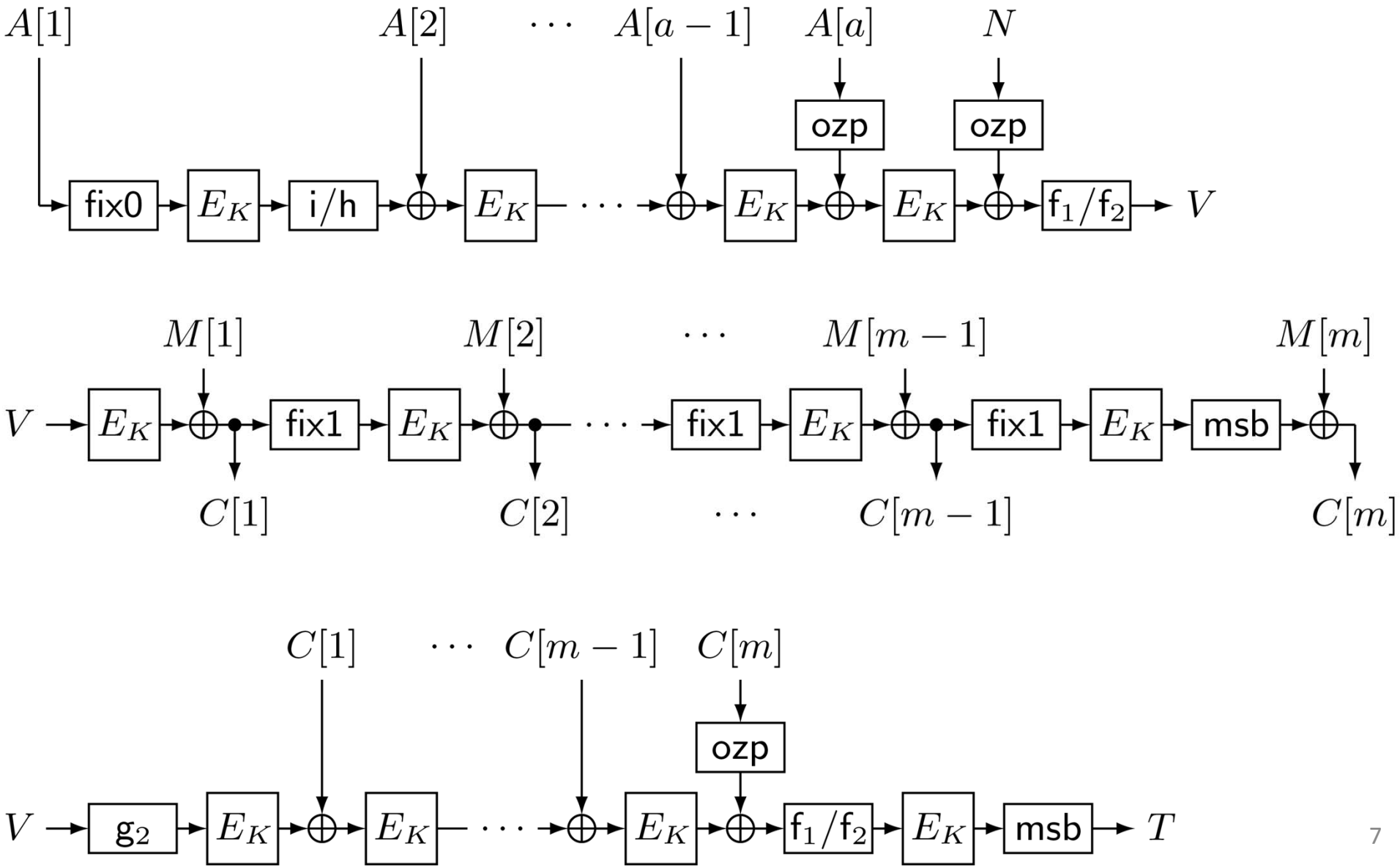


Parameters of CLOC and SILC

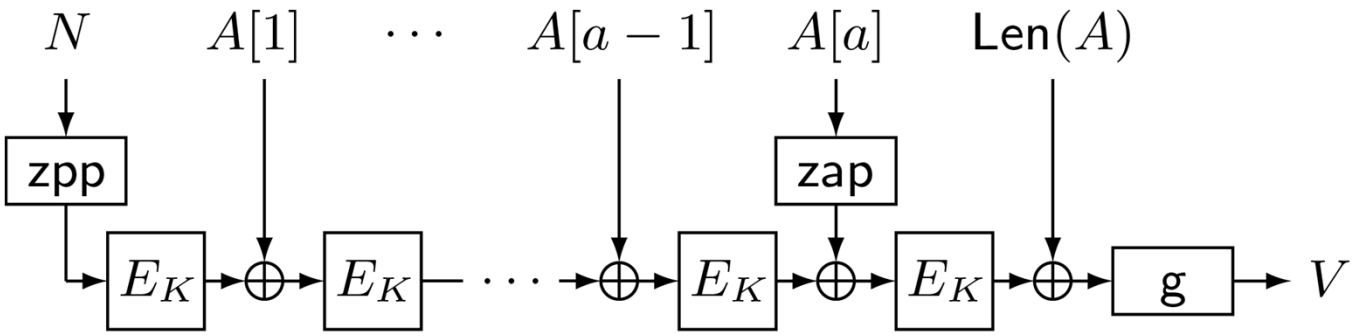
- E: blockcipher
- l_N: nonce length
- τ: tag length

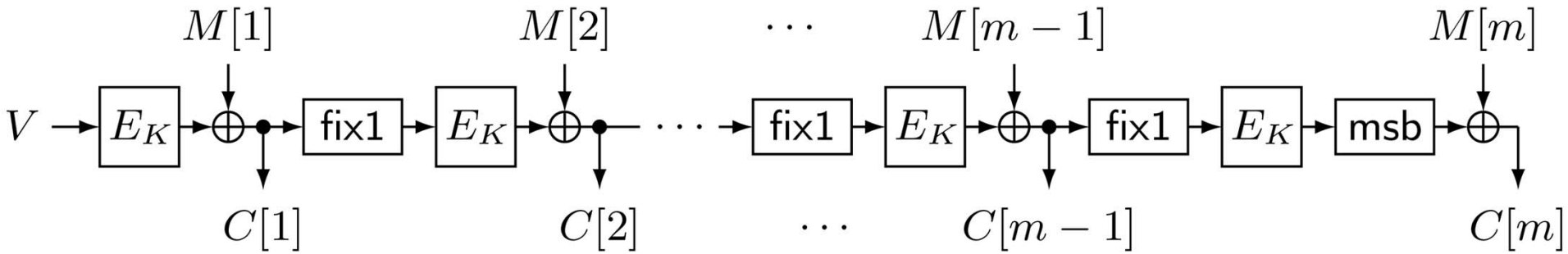


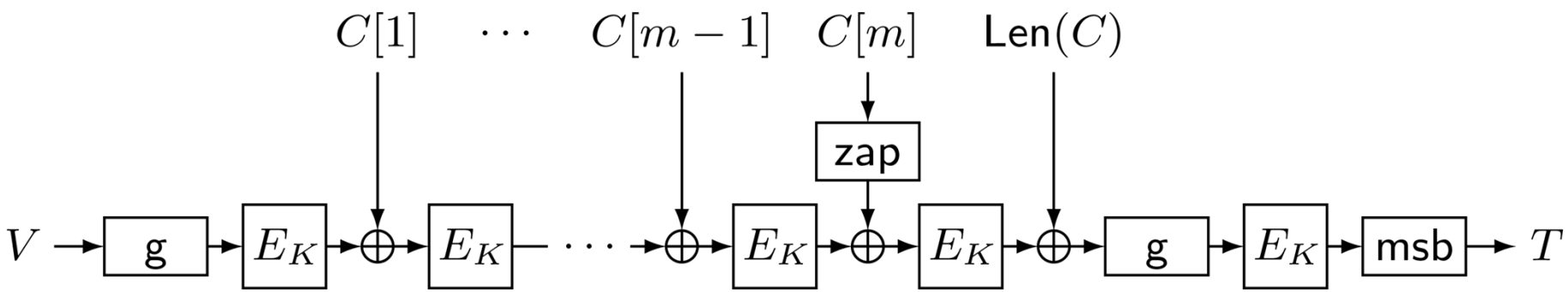
CLOC


A[1]
A[2]
· · ·
A[a − 1]
A[a]
N
ozp
ozp
fix0
E_K
i/h
E_K
· · ·
E_K
E_K
f1/f2
V
M[1]
M[2]
· · ·
M[m − 1]
M[m]
V
E_K
fix1
E_K
· · ·
fix1
E_K
fix1
E_K
msb
C[1]
C[2]
· · ·
C[m − 1]
C[m]
C[1]
· · ·
C[m − 1]
C[m]
ozp
V
g2
E_K
E_K
· · ·
E_K
f1/f2
E_K
msb
T

SILC


N
$A[1]$
$\cdots$
$A[a-1]$
$A[a]$
$\mathsf{Len}(A)$
zpp
zap
E_K
g
V


$M[1]$
$M[2]$
$\cdots$
$M[m-1]$
$M[m]$
V
E_K
fix1
msb
$C[1]$
$C[2]$
$\cdots$
$C[m-1]$
$C[m]$


$C[1]$
$\cdots$
$C[m-1]$
$C[m]$
$\mathsf{Len}(C)$
zap
V
g
E_K
msb
T

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

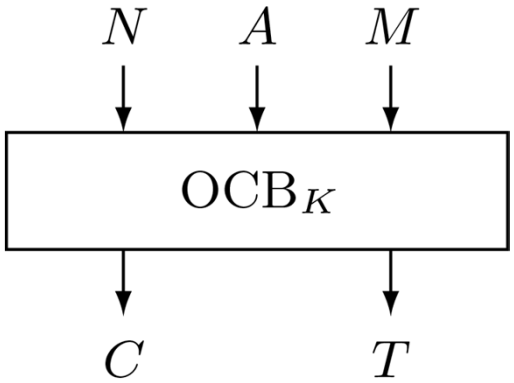
Comment at CFRG

- [Cfrg] Attacker changing tag length in OCB
  - Comment for http://tools.ietf.org/html/draft-irtf-cfrg-ocb-02 by James Manger on May 29, 2013
  - "OCB with tag lengths of 64, 96, and 128 bits are defined. 64-bit and 96-bit tags are simply truncated 128-bit tags. The tag length is not mixed into the ciphertext. It never affects any input to an AES operation. Consequently, given a valid output from the AEAD_AES_128_OCB_TAGLEN128 algorithm it is trivial to produce a valid output from the AEAD_AES_128_OCB_TAGLEN64 algorithm -- just drop the last 8 bytes. Is this ok?"
- Similar issues pointed out for CCM [1] and OMD [2]

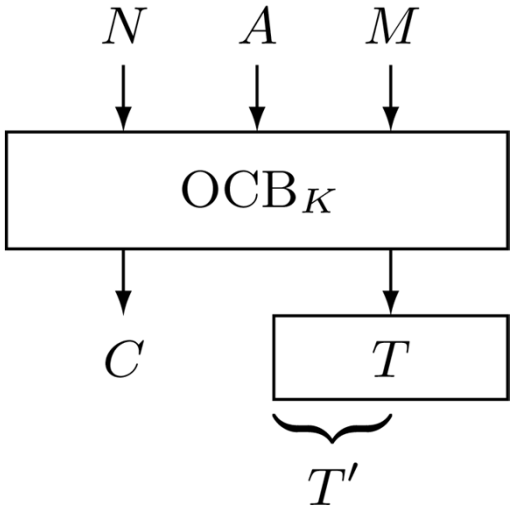
[1] Rogaway, Wagner: A Critique of CCM. Cryptology ePrint Archive, Report 2003/070 (2003)
[2] Eichlseder: Remark on variable tag lengths and OMD. CAESAR mailing list (2014)

Parameter Change in OCB

- (N, A, M, C, T) for AEAD_AES_128_OCB_TAGLEN128



- (N, A, C, T’) is valid for AEAD_AES_128_OCB_TAGLEN64



Parameter Change in OCB

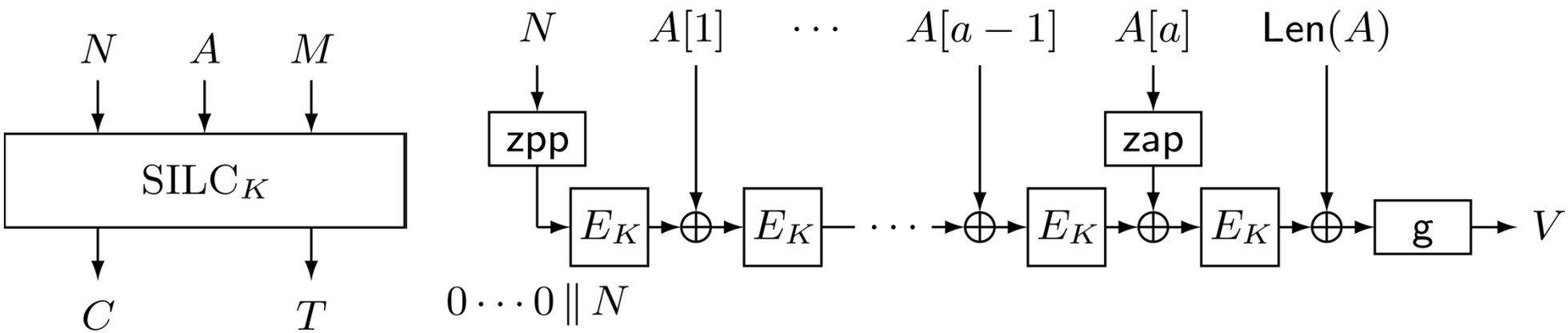
- This does not contradict the provable security result
  - assumes that parameters are fixed
- Designers usually expect that keys are independent if parameters are changed
  - Not an attack, but a kind of “parameter misuse,” related to the robustness
    - CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness
- Easy to address
  - Nonce = “0…01 || N” -> Nonce = “TAGLEN || 0…01 || N”
  - Provable security result is maintained

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

CLOC and SILC

- (N, A, M, C, T) for SILC with $l_N = 96$ and $\tau = 128$
  - assume that $\mathrm{msb}_{32}(N)=0\ldots0$



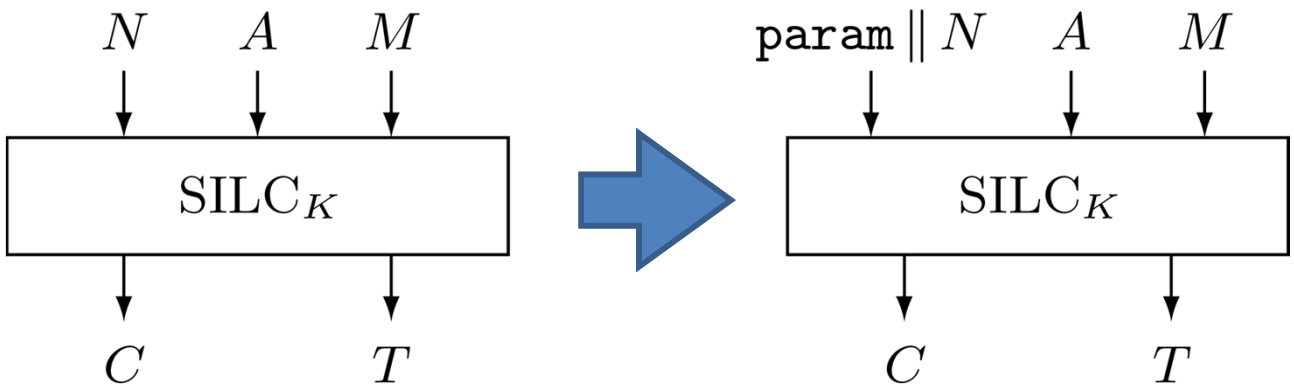
- (N', A, C, T') is valid for the parameters $l_N = 64$ and $\tau = 64$ with N' = $\mathrm{lsb}_{64}(N)$ and T' = $\mathrm{msb}_{64}(T)$
- A similar observation of changing the tag length holds for CLOC

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

Introduce param

- param: an 8-bit constant that depends on the parameters
- param is hardcoded into encryption and decryption algorithms
- use “param || N” instead of N



Definition of param for CLOC

	E	ℓ_N	τ	param
*	AES-128	12	8	0xc0
	AES-128	12	12	0xc1
	AES-128	12	16	0xc2
	AES-128	12	4	0xc3
*	AES-128	8	8	0xd0
	AES-128	8	12	0xd1
	AES-128	8	16	0xd2
	AES-128	8	4	0xd3
	AES-128	14	8	0xe0
	AES-128	14	12	0xe1
	AES-128	14	16	0xe2
	AES-128	14	4	0xe3

	E	ℓ_N	τ	param
*	TWINE-80	6	4	0xcc
	TWINE-80	6	6	0xcd
	TWINE-80	6	8	0xce
	TWINE-80	4	4	0xdc
	TWINE-80	4	6	0xdd
	TWINE-80	4	8	0xde

- lengths are in byes, param is in hex, * denotes the recommended parameter

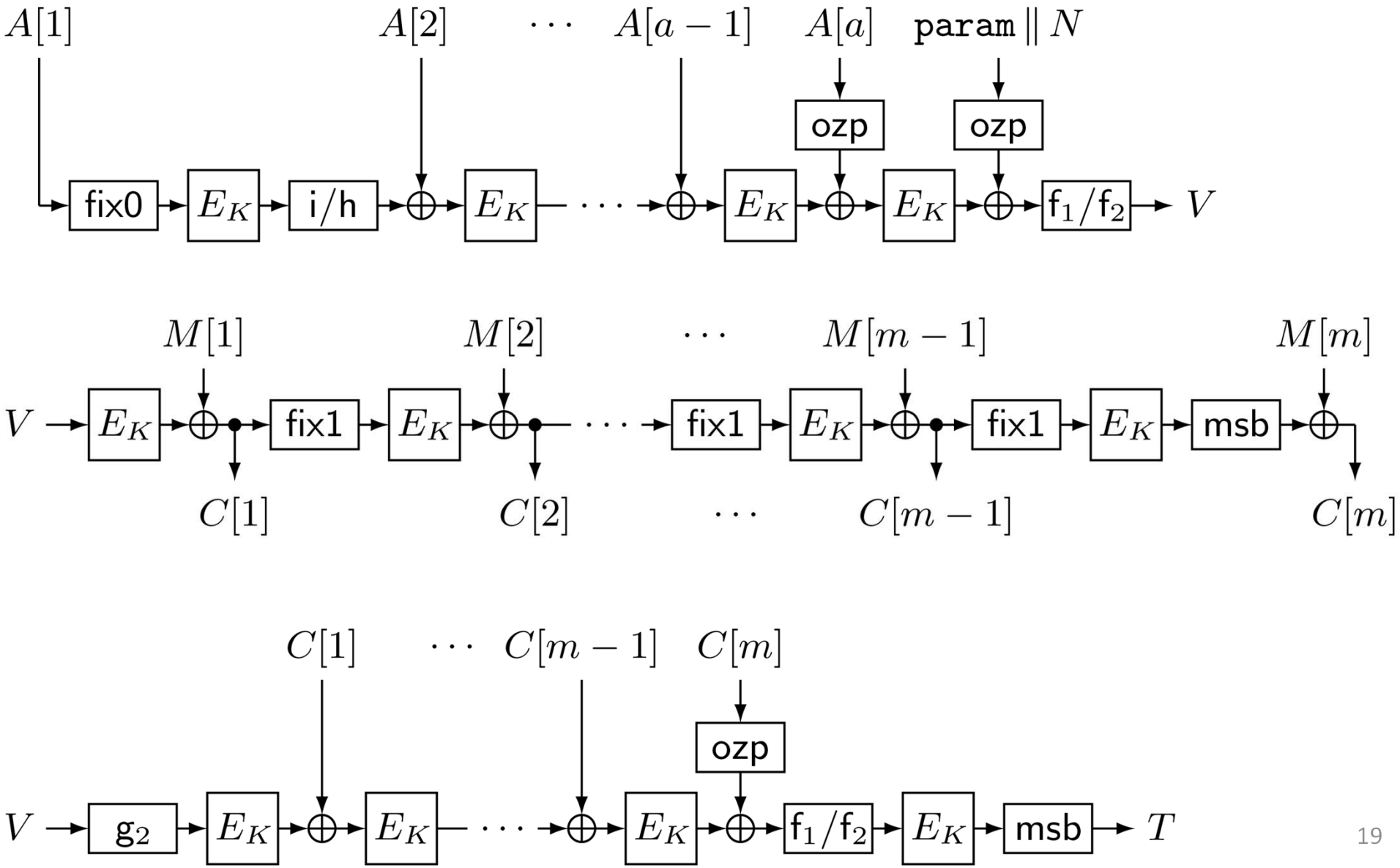
Definition of param for SILC

	E	ℓ_N	τ	param
*	AES-128	12	8	0xc0
	AES-128	12	12	0xc1
	AES-128	12	16	0xc2
	AES-128	12	4	0xc3
*	AES-128	8	8	0xd0
	AES-128	8	12	0xd1
	AES-128	8	16	0xd2
	AES-128	8	4	0xd3
	AES-128	14	8	0xe0
	AES-128	14	12	0xe1
	AES-128	14	16	0xe2
	AES-128	14	4	0xe3

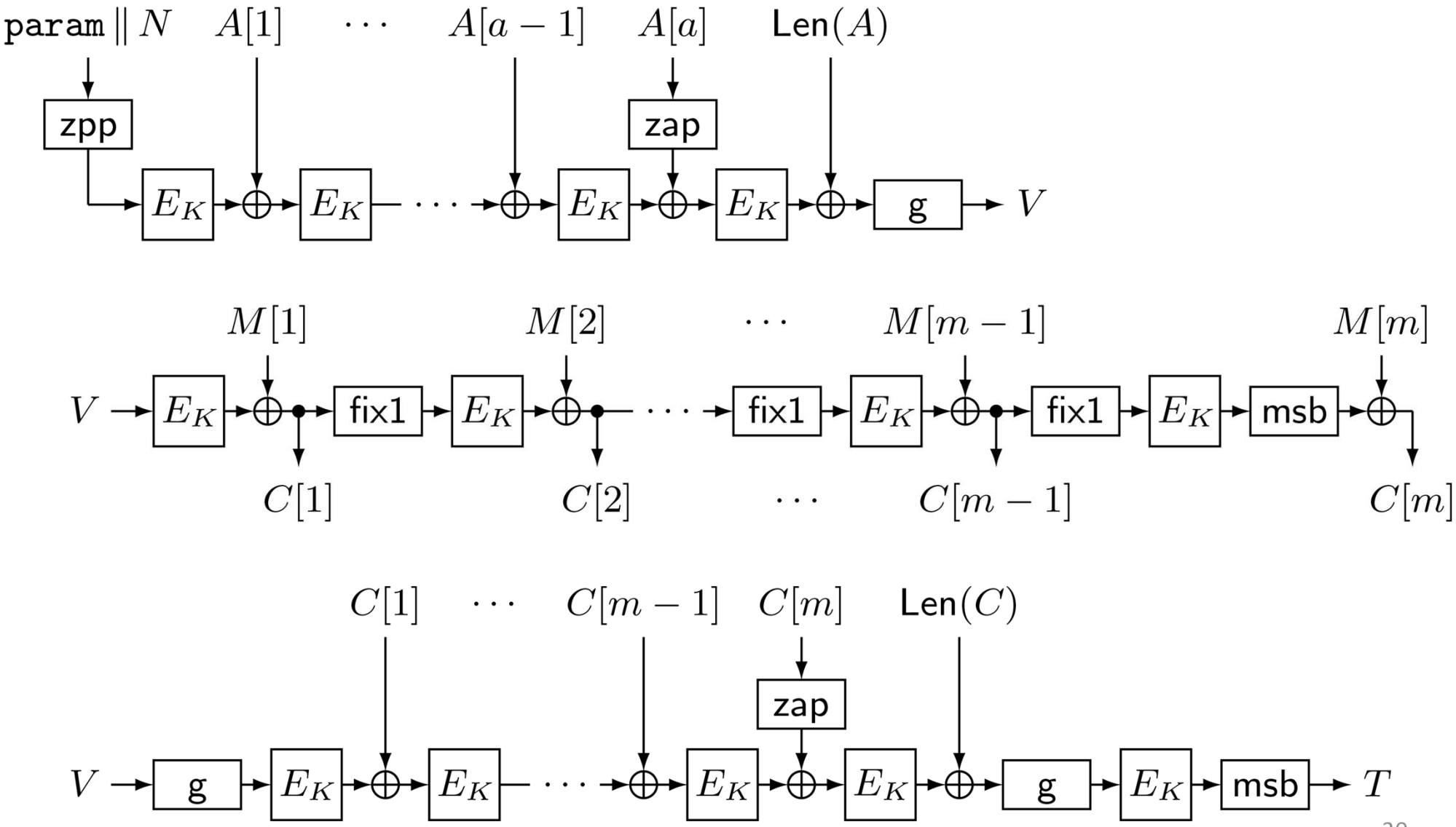
	E	ℓ_N	τ	param
*	PRESENT-80	6	4	0xc4
	PRESENT-80	6	6	0xc5
	PRESENT-80	6	8	0xc6
	PRESENT-80	4	4	0xd4
	PRESENT-80	4	6	0xd5
	PRESENT-80	4	8	0xd6
*	LED-80	6	4	0xc8
	LED-80	6	6	0xc9
	LED-80	6	8	0xca
	LED-80	4	4	0xd8
	LED-80	4	6	0xd9
	LED-80	4	8	0xda

- lengths are in byes, param is in hex, * denotes the recommended parameter

CLOC v2


A[1]
A[2]
A[a − 1]
A[a]
param || N
ozp
ozp
fix0
E_K
i/h
E_K
E_K
E_K
f1/f2
V
M[1]
M[2]
M[m − 1]
M[m]
V
E_K
fix1
E_K
fix1
E_K
fix1
E_K
msb
C[1]
C[2]
C[m − 1]
C[m]
C[1]
C[m − 1]
C[m]
ozp
V
g2
E_K
E_K
E_K
f1/f2
E_K
msb
T

SILC v2


param || N
A[1]
A[a − 1]
A[a]
Len(A)
zpp
zap
E_K
g
V
M[1]
M[2]
M[m − 1]
M[m]
fix1
msb
C[1]
C[2]
C[m − 1]
C[m]
Len(C)
T

Notes

- param does not mean that CLOC and SILC handle variable length nonces nor variable length tags
  - The parameters have to be fixed during the lifetime of the secret key
- param does not affect the provable security results
  - “param || N” can be considered as the nonce
- param does not remove the dependency to other blockcipher modes of operation
  - the concurrent use (with the same secret key) of CLOC and ECB mode is insecure
  - Similarly, CLOC and SILC cannot be used concurrently

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

Updates on Software Implementation

- CLOC at FSE 2014
  - Intel (R) Core (TM) i5-3427U 1.80GHz (Ivy Bridge family), AES-128, AES-NI, about 4.9 cpb
    - for a long plaintext (more than 2^{20} blocks) and empty associated data
- SILC at DIAC 2014
  - about the same speed with the same processor and the same input

Updates on Software Implementation

- Updates
  - Intel (R) Core (TM) i5-4570 3.20GHz (Haswell family), AES-128, AES-NI
  - Seral AES runs at 4.44 cpb
  - CLOC v2 and SILC v2 run at 4.56 cpb
    - very close to the speed of seral AES

Updates on Hardware Implementation

- ASIC implementation
  - reference implementation (non-optimized, encryption-and-decryption implemented)
  - Environment: 90nm standard cell library with logic synthesis done by Synopsys DC Version D-2010.03-SP1-1
- CLOC

AES		TWINE	
AES128_CLOC	18991.5	TWINE80_CLOC	5917.8
AES Core	10207.8	TWINE Core	1459.5
ratio	1.9	ratio	4.1

in GE (Gate Equivalent)

Updates on Hardware Implementation

- ASIC implementation
  - reference implementation (non-optimized, encryption-and-decryption implemented)
  - Environment: 90nm standard cell library with logic synthesis done by Synopsys DC Version D-2010.03-SP1-1
- SILC

AES		TWINE		PRESENT	
AES128_SILC	17466.0	TWINE80_SILC	5178.0	PRESENT80_CLOC	5532.3
AES Core	10207.8	TWINE Core	1459.5	PRESENT Core	1817.3
ratio	1.7	ratio	3.5	ratio	3.0

in GE (Gate Equivalent)

Other Updates and Future Plan

- Parameter space has been adjusted to handle param
- Intellectual Property statement of CLOC has been updated
- Full security proof of SILC
- Web site:
  - http://www.nuee.nagoya-u.ac.jp/labs/tiwata/AE/
  - documents, slides, test vectors

- Future plan:
  - Analysis of CLOC and SILC in terms of INT-RUP security
  - Unify the documents of CLOC and SILC into one document
  - Designing a variant of SILC for empty associated data

Agenda

- Brief review of CLOC and SILC
- An issue discussed at CFRG related to OCB
- How the issue affects CLOC v1 and SILC v1
- How this is addressed in CLOC v2 and SILC v2
- Updates on implementation results

Thank you

Details of param for CLOC

- n=128, param = (p1, p2,…, p8)
  - (p1, p2) = (1, 1)
  - (p3, p4) = (0, 0) if $l_N = 12$, (0, 1) if $l_N = 8$, (1, 0) if $l_N = 14$
  - (p5, p6) = (0, 0) (reserved for AES)
  - (p7, p8) = (0, 0) if $\tau = 8$, (0, 1) if $\tau = 12$, (1, 0) if $\tau = 16$, (1, 1) if $\tau = 4$

- n=64, param = (p1, p2,…, p8)
  - (p1, p2) = (1, 1)
  - (p3, p4) = (0, 0) if $l_N = 6$, (0, 1) if $l_N = 4$
  - (p5, p6) = (1, 1) (reserved for Twine)
  - (p7, p8) = (0, 0) if $\tau = 4$, (0, 1) if $\tau = 6$, (1, 0) if $\tau = 8$

Details of param for SILC

- n=128, param = (p1, p2,…, p8)
  - same as CLOC

- n=64, param = (p1, p2,…, p8)
  - (p1, p2) = (1, 1)
  - (p3, p4) = (0, 0) if $l_N = 6$, (0, 1) if $l_N = 4$
  - (p5, p6) = (0, 1) if Present, (1, 0) if LED
  - (p7, p8) = (0, 0) if $\tau = 4$, (0, 1) if $\tau = 6$, (1, 0) if $\tau = 8$