Updates on CLOC and SILC Version 3

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

DIAC 2016

September 26, 2016, Nagoya, Japan

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

Outline

- Review of the schemes
  - Specification of CLOC and SILC, security, implementation
- Updates from version 2 to version 3
- Discussion
- Future plan

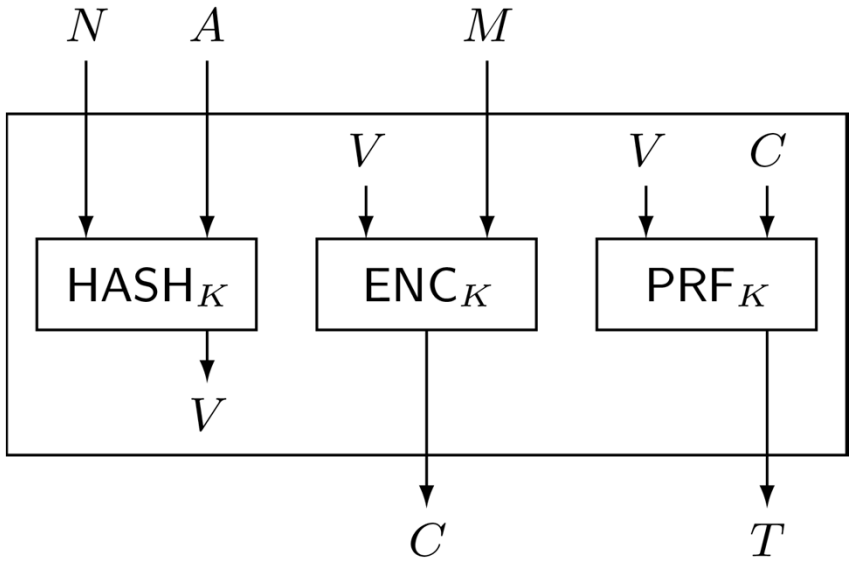
CLOC and SILC

- CLOC [1]
  - Compact Low-Overhead CFB, FSE 2014
  - Suitable for handling short input data on small microprocessors
- SILC [2]
  - SImple Lightweight CFB, DIAC 2014
  - Hardware oriented version of CLOC

[1] Iwata, Minematsu, Guo, Morioka: CLOC: Authenticated Encryption for Short Input. FSE 2014
[2] Iwata, Minematsu, Guo, Morioka, Kobayashi: SILC: SImple Lightweight CFB. DIAC 2014

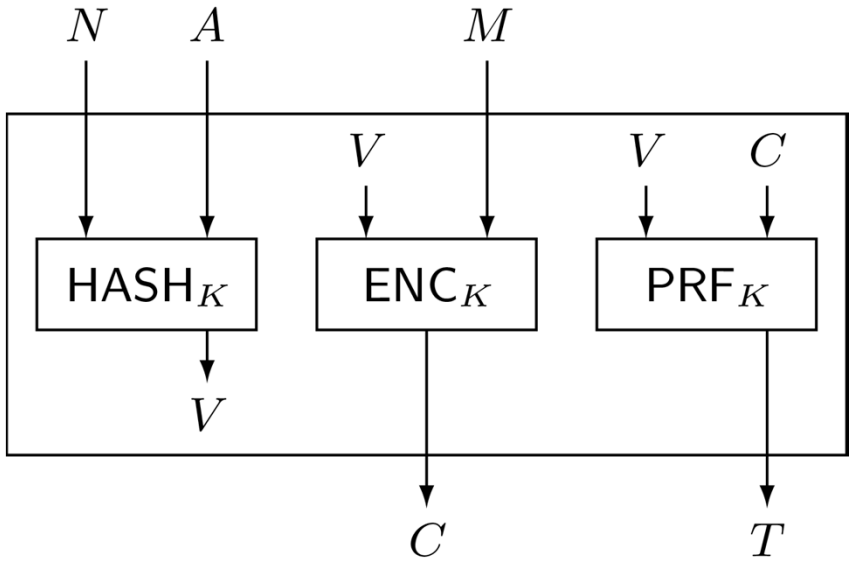
Overview of CLOC and SILC

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

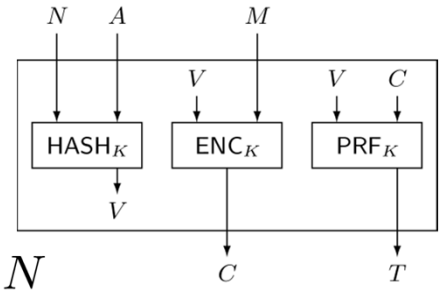


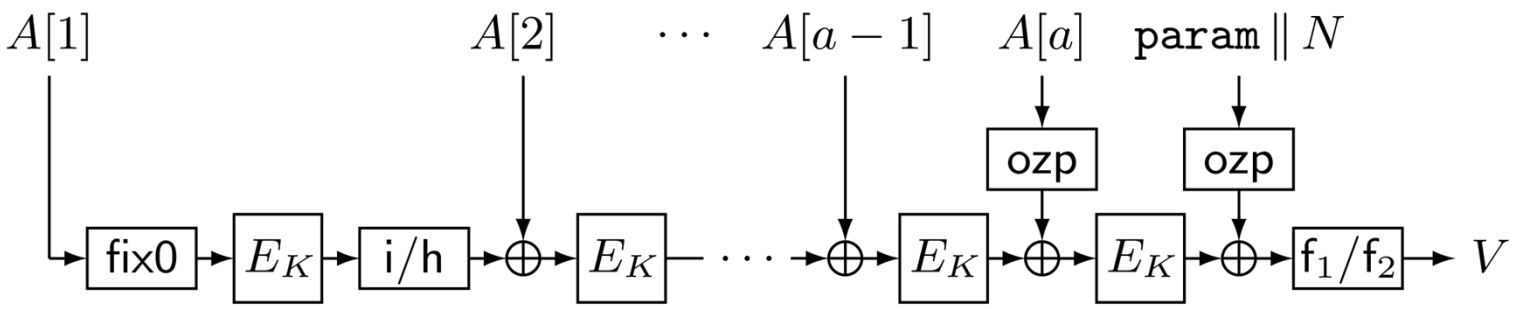
Parameters of CLOC and SILC

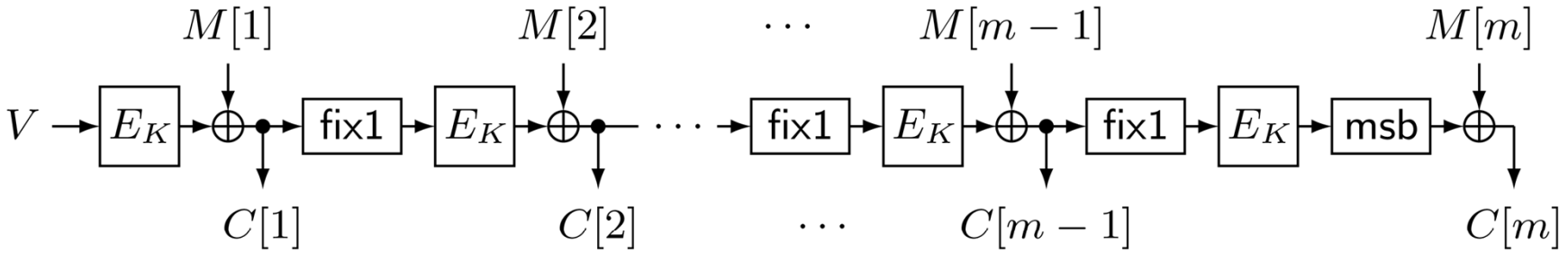
- E: blockcipher
- l_N: nonce length
- τ: tag length

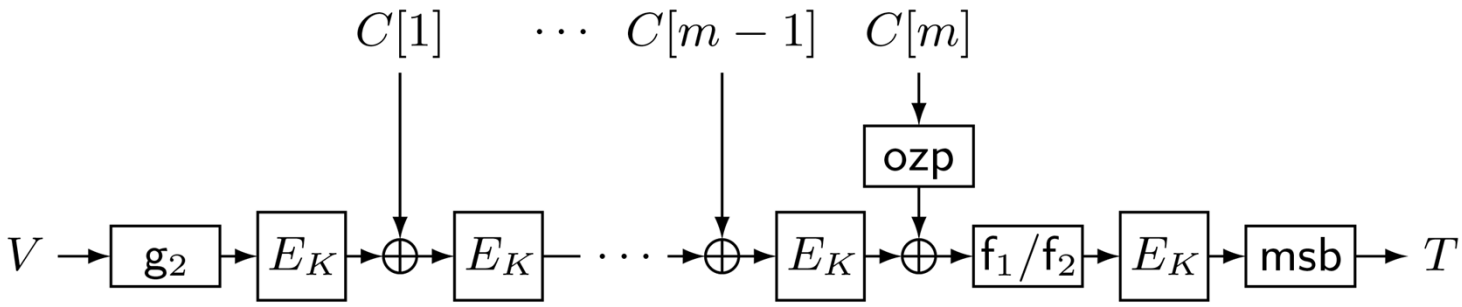


CLOC v3

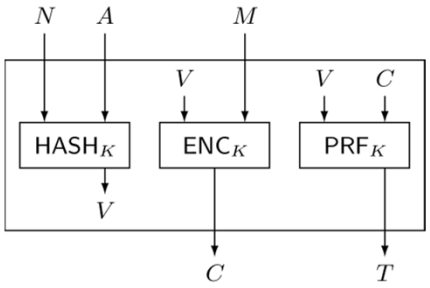

N
A
M
V
V
C
HASH$_K$
ENC$_K$
PRF$_K$
V
C
T

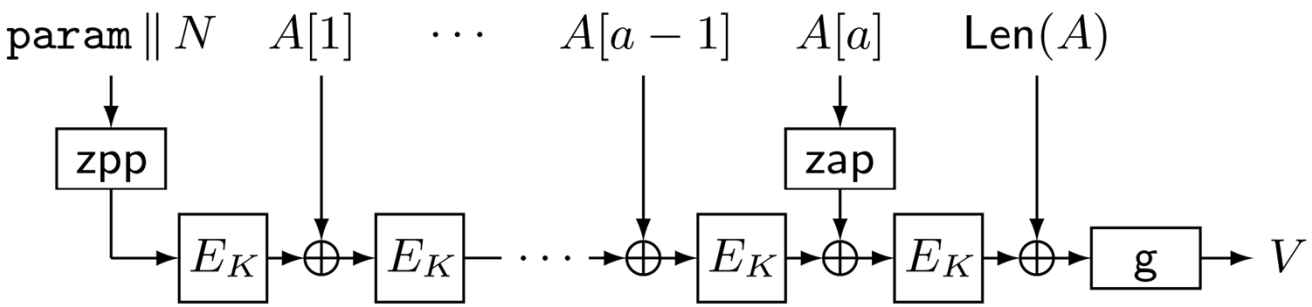

A[1]
A[2]
· · ·
A[a − 1]
A[a]
param || N
ozp
ozp
fix0
E_K
i/h
E_K
E_K
E_K
E_K
f1/f2
V

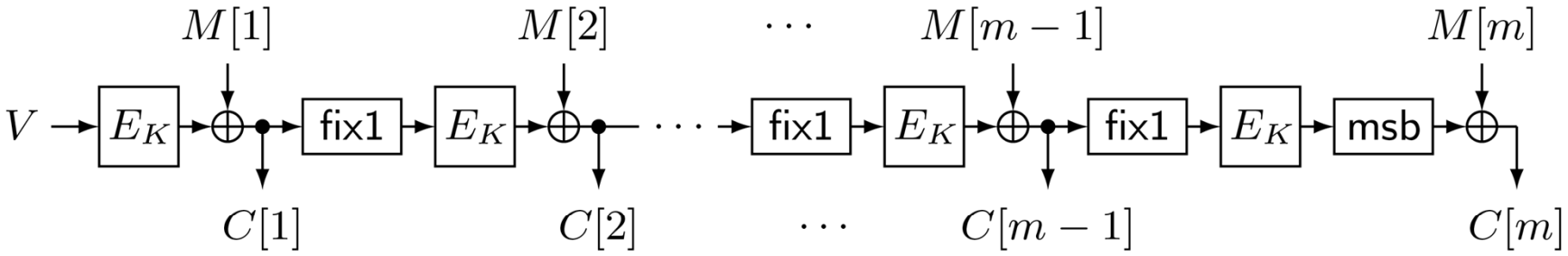

M[1]
M[2]
· · ·
M[m − 1]
M[m]
V
E_K
fix1
E_K
fix1
E_K
msb
C[1]
C[2]
· · ·
C[m − 1]
C[m]

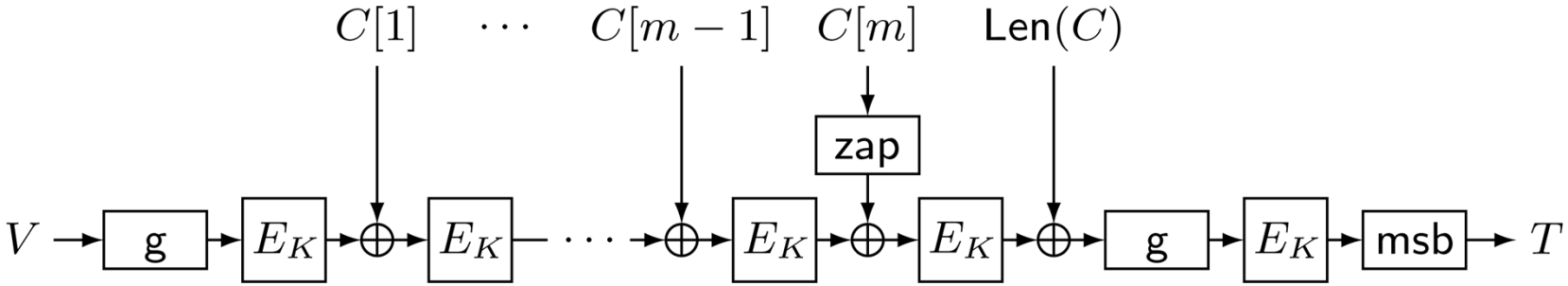

C[1]
· · ·
C[m − 1]
C[m]
ozp
V
g2
E_K
E_K
E_K
E_K
f1/f2
E_K
msb
T

SILC v3


N
A
M
V
V
C
HASH$_K$
ENC$_K$
PRF$_K$
V
C
T


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]
V
E_K
fix1
msb
C[1]
C[2]
· · ·
C[m − 1]
C[m]


C[1]
· · ·
C[m − 1]
C[m]
Len(C)
zap
V
g
E_K
msb
T

Security of CLOC and SILC

- CLOC and SILC are provably secure up to the standard birthday bound
  - Privacy: against nonce-respecting adversaries
  - Authenticity: against nonce-reusing adversaries
  - $O(\sigma^2/2^n)$, σ is the number of total blocks, n is the block length in bits

Implementation

- A: associated data of a blocks
- M: a plaintext of m blocks
- CLOC
  - 1 + a + 2m blockcipher calls when $|A| \geq 1$
  - 2 + 2m calls when $|A| = 0$
  - No precomputation beyond the blockcipher key schedule
- SILC
  - 3 + a + 2m blockcipher calls
  - No precomputation beyond the blockcipher key schedule

Software Implementation

- CLOC and SILC with AES (from DIAC 2015)
  - Intel (R) Core (TM) i5-4570 3.20GHz (Haswell family), AES-128, AES-NI
  - CLOC and SILC run at 4.56 cpb
    - seral AES runs at 4.44 cpb, very close to the speed of seral AES
- Latest performance at public benchmark (SUPERCOP)
  - Intel (R) Core (TM) i5-6600 (Skylake) :
  - 2.82 cpb for long messages, 7.81 cpb for 64-byte messages

Software Implementation

- Intel (R) Core (TM) i5-4570 3.20GHz (Haswell family), long messages (2048 bytes), vperm (vector permutation)
  - [update] CLOC with TWINE runs at 14.6 cpb
  - SILC with Present runs at about 70 cpb
  - SILC with LED runs at about 130 cpb
- CLOC Implementations on Atmel AVR ATmega128 (FSE 2014)
  - 8-bit microprocessor
  - the RAM usage is low and Init is fast, and it is fast for short input data, up to around 128 bytes

Hardware Implementation (from DIAC 2015)

- 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)

Hardware Implementation (from DIAC 2015)

- 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)

Hardware Implementation

- Area optimized ASIC implementation with dedicated API [Banik, Bogdanov, Minematsu, DIAC 2015]

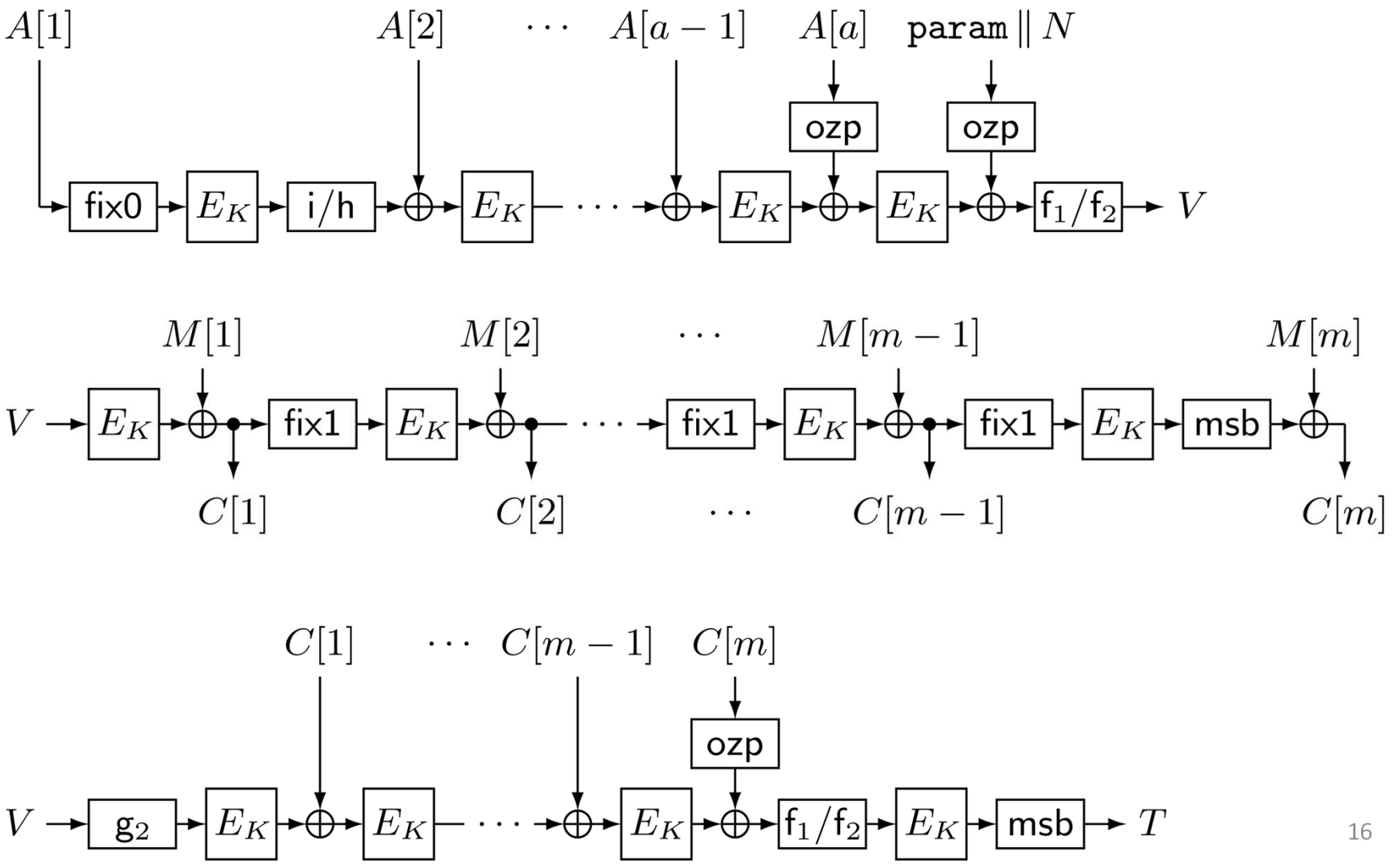
	Design	Area (μm^2)	Area (GE)
AES128_CLOC	Aggressive	13672.8	3107
AES128_CLOC	Conservative	18966.5	4311
AES128_SILC	Aggressive	13678.3	3109
AES128_SILC	Conservative	18100.0	4114

- Reference implementations were submitted to ATHENa

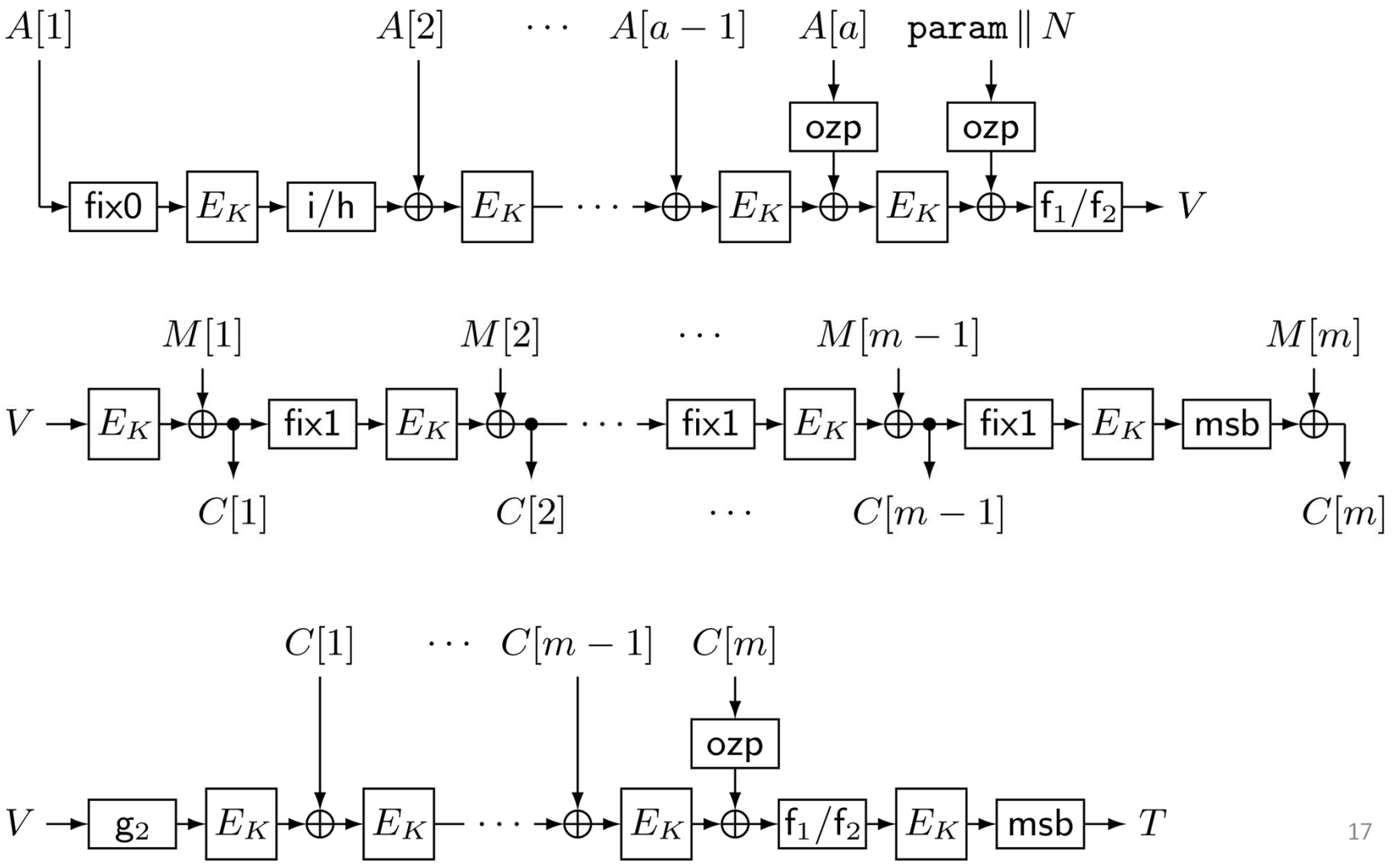
Outline

- Review of the schemes
  - Specification of CLOC and SILC, security, implementation
- Updates from version 2 to version 3
- Discussion
- Future plan

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
E_K
f1/f2
V
M[1]
M[2]
⋯
M[m − 1]
M[m]
V
E_K
fix1
E_K
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
E_K
f1/f2
E_K
msb
T

CLOC v3


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

Updates from Version 2 to 3

- No changes in specifications (both for CLOC and SILC)
- Two submission documents, “CLOC v2” and “SILC v2,” were merged into one submission “CLOC and SILC v3”
  - following the request by CAESAR
- Recommended parameter sets
  - 3 in “CLOC v2”
    aes128n12t8clocv2, aes128n8t8clocv2, twine80n6t4clocv2
  - 4 in “SILC v2”
    aes128n12t8silcv2, aes128n8t8silcv2, present80n6t4silcv2, led80n6t4silcv2
  - 3 + 4 = 7 -> 5 in “CLOC and SILC v3”

Updates from Version 2 to 3

- 5 recommended parameter sets for “CLOC and SILC v3”
- specified the ordering:
  1. aes128n12t8clocv3
  2. aes128n12t8silcv3
  3. twine80n6t4clocv3
  4. present80n6t4silcv3
  5. led80n6t4silcv3
- dropped aes128n8t8clocv2 and aes128n8t8silcv2

Updates from Version 2 to 3

- CAESAR Use Cases:
  - Use Case 1: Lightweight applications
  - Use Case 2: High-performance applications
  - Use Case 3: Defense in depth
- Targeted use cases of 5 recommended parameter sets
  1. aes128n12t8clocv3 Use Case 1
  2. aes128n12t8silcv3 Use Case 1
  3. twine80n6t4clocv3 Use Case 1
  4. present80n6t4silcv3 Use Case 1
  5. led80n6t4silcv3 Use Case 1
- CLOC and SILC have some features of Use Case 3, but this is not a targeted use case

Other Updates

- n=96 is listed as a possible choice of the block length of the underlying blockcipher
- added Simon and Speck as possible options

Outline

- Review of the schemes
  - Specification of CLOC and SILC, security, implementation
- Updates from version 2 to version 3
- Discussion
- Future plan

History: CCM, EAX, and EAX-prime

- CCM (NIST SP 800-38C)
  - complex specification, not online, a number of limitations were pointed out [3]
- EAX (ISO/IEC 19772)
  - simple design, AES-CMAC plus AES-CTR
  - provably secure
  - precomputation cost ($E_K(0)$, $E_K(1)$, and $E_K(2)$) can be an issue for highly constrained devices; time and memory

[3] Rogaway, Wagner: A Critique of CCM. IACR ePrint 2003/070

History: CCM, EAX, and EAX-prime

- EAX-prime (ANSI C12.22)
  - reduces precomputation complexity of EAX
  - efficiently handles short input data with small memory
    - was not proposed with a proof of security
  - standardized by ANSI for Smartgrid
  - ANSI pushed EAX-prime to NIST, and NIST requested public comments for inclusion of it into NIST SP-800 series
  - EAX-prime was seriously broken [4]
    - single-query forgery etc.
    - was not adopted in the NIST standard

[4] Minematsu, Lucks, Morita, Iwata: Attacks and Security Proofs of EAX-Prime. FSE 2013

From CCM, EAX, and EAX-prime to CLOC and SILC

- There is a public need for a scheme like CCM, EAX, and EAX-prime:
  - constrained devices, blockcipher-based, design simplicity, small footprint
  - low memory requirement, low precomputation complexity, and provable security
- CCM -> EAX -> EAX-prime -> CLOC and SILC
  - catch “the public need”

64-bit blockciphers

- 64-bit blockciphers are in use:
  - TDES, Kasumi, Present,…
- Many new designs for lightweight applications
  - LED, TWINE, Prince, Midori, Rectangle,…
- a blockcipher cannot be used by itself
  - encryption mode, MAC for authentication, AE mode
  - in modes, block length matters, not for general purpose applications
  - for low-powered, small-bandwidth, or limited-lifetime communications

64-bit blockciphers

- showing “the right way” of using 64-bit blockciphers to the outside world is important
  - Generic composition works
  - CCM and GCM are for 128-bit blockciphers
    - there is a 64-bit block version of GCM
  - CLOC and SILC give one right way

Outline

- Review of the schemes
  - Specification of CLOC and SILC, security, implementation
- Updates from version 2 to version 3
- Discussion
- Future plan

Future Plan

- Future plan (from DIAC 2015, still in progress):
  - Analysis of CLOC and SILC in terms of INT-RUP security
  - Designing a variant of SILC for empty associated data

Thank you

- Review of the schemes
  - Specification of CLOC and SILC, security, implementation
- Updates from version 2 to version 3
- Discussion
- Future plan

- Web site:
  - http://www.nuee.nagoya-u.ac.jp/labs/tiwata/AE/
  - documents, slides, test vectors