A Survey of CBC MAC Variants

Tetsu Iwata

Ibaraki University

Tokyo, Japan, June 3, 2003.

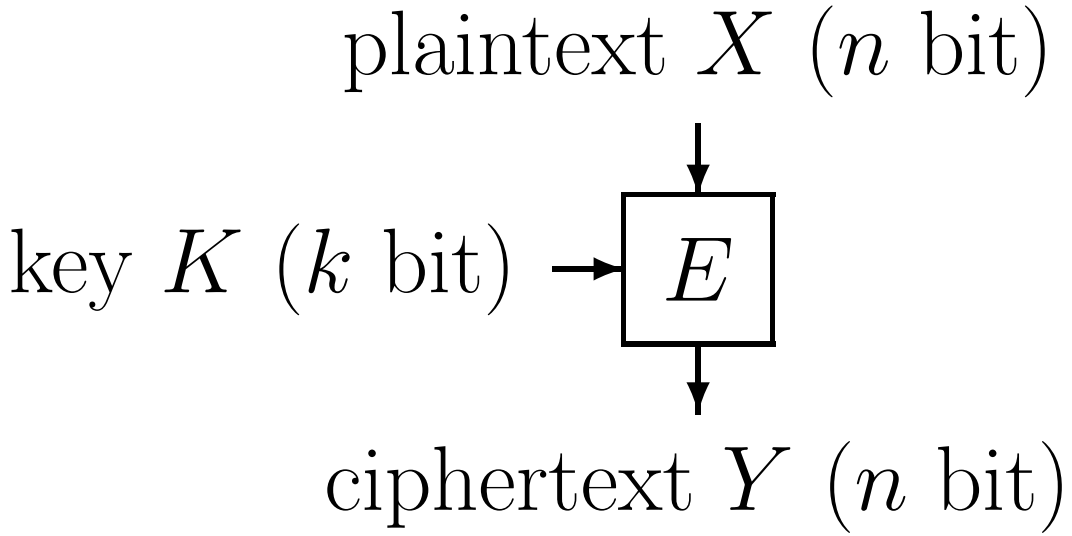
CBC MAC Variants

EMAC, RMAC, XCBC, TMAC, OMAC.

Under consideration for standardization by NIST

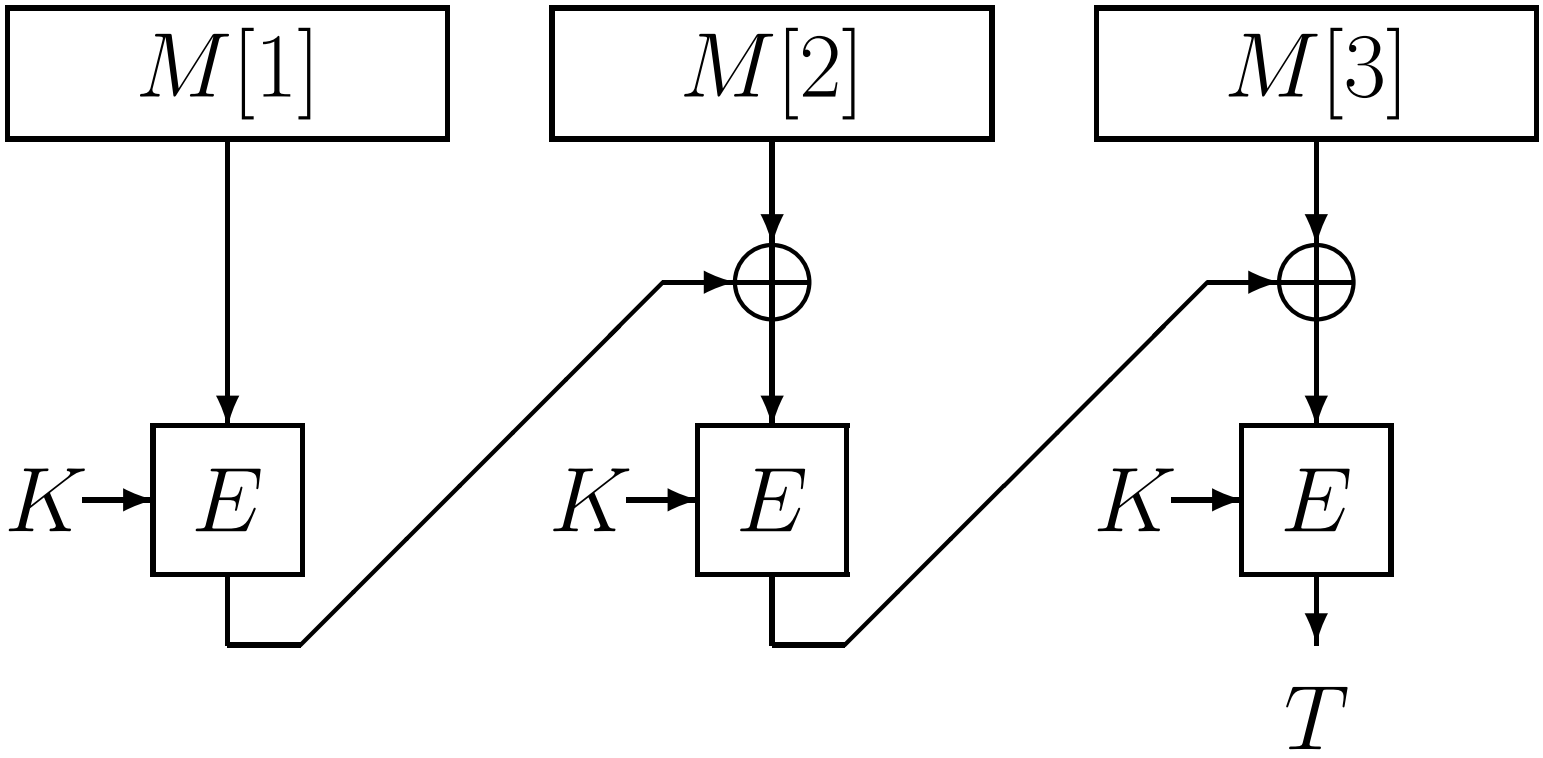
- Security
- Efficiency

Block Cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$



- Each $E_K(\cdot)$ is a permutation on $\{0,1\}^n$.
- Examples: AES, Camellia, TDES, ...

CBC MAC



CBC MAC

- Simple
- Secure (on messages of a **fixed** length) [BKR94]
- Widely standardized:

ANSI X9.19, FIPS 113, ISO 9797

Problems of CBC MAC

- does not allow messages of **varying** lengths

  (**insecure**)

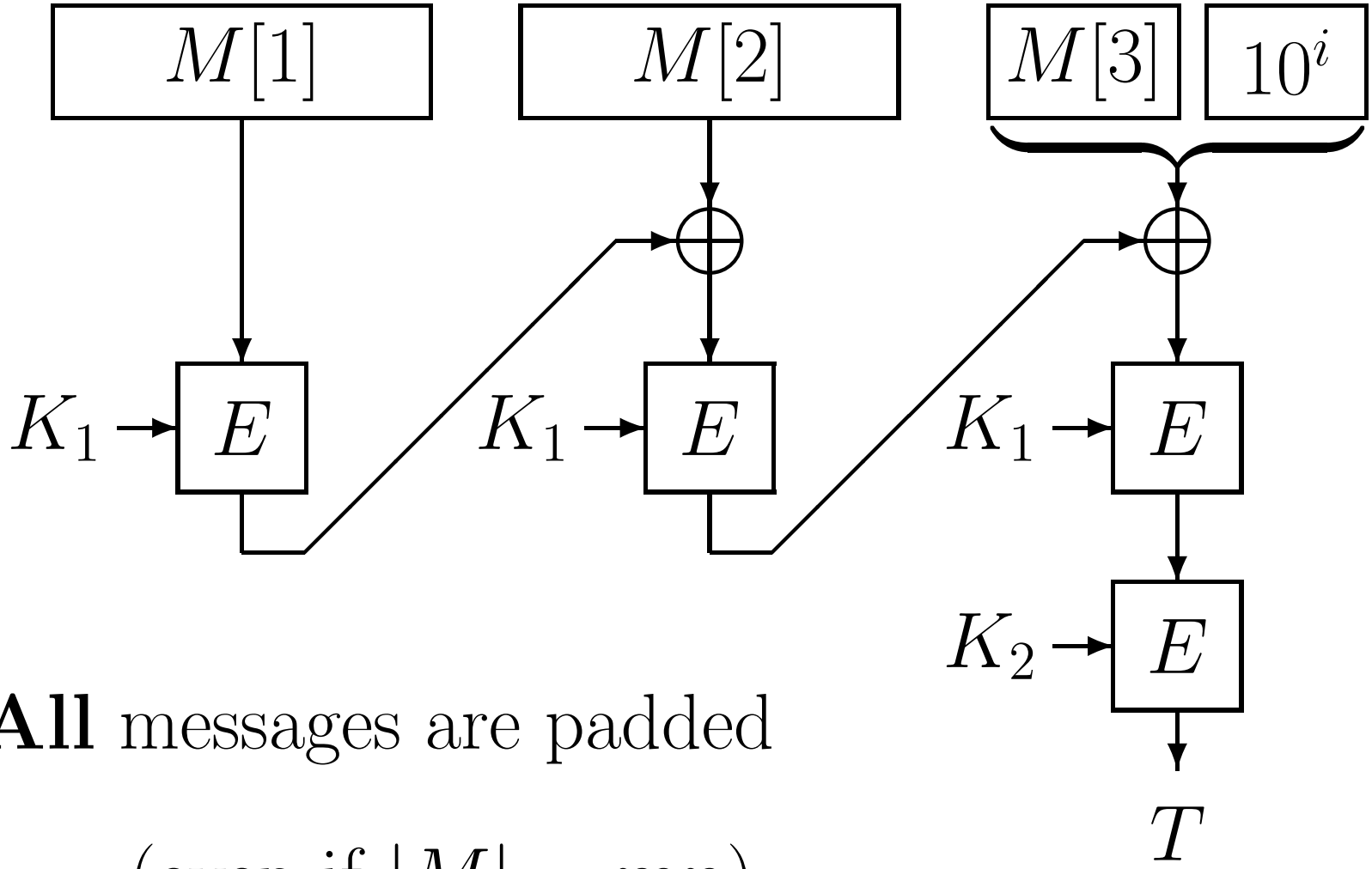
- does not allow messages of **arbitrary bit length**

  (all messages must be a multiple of n bits)

Proposals

- EMAC [Race Project, NESSIE]
  (Analysis by [Petrank, Rackoff])
- RMAC [Jaulmes, Joux, Valette]
- XCBC [Black, Rogaway]
- TMAC [Kurosawa, Iwata]
- OMAC [Iwata, Kurosawa]

EMAC (Race Project, NESSIE)



All messages are padded

(even if $|M| = mn$).

Advantages of EMAC

- Secure for messages of varying lengths.

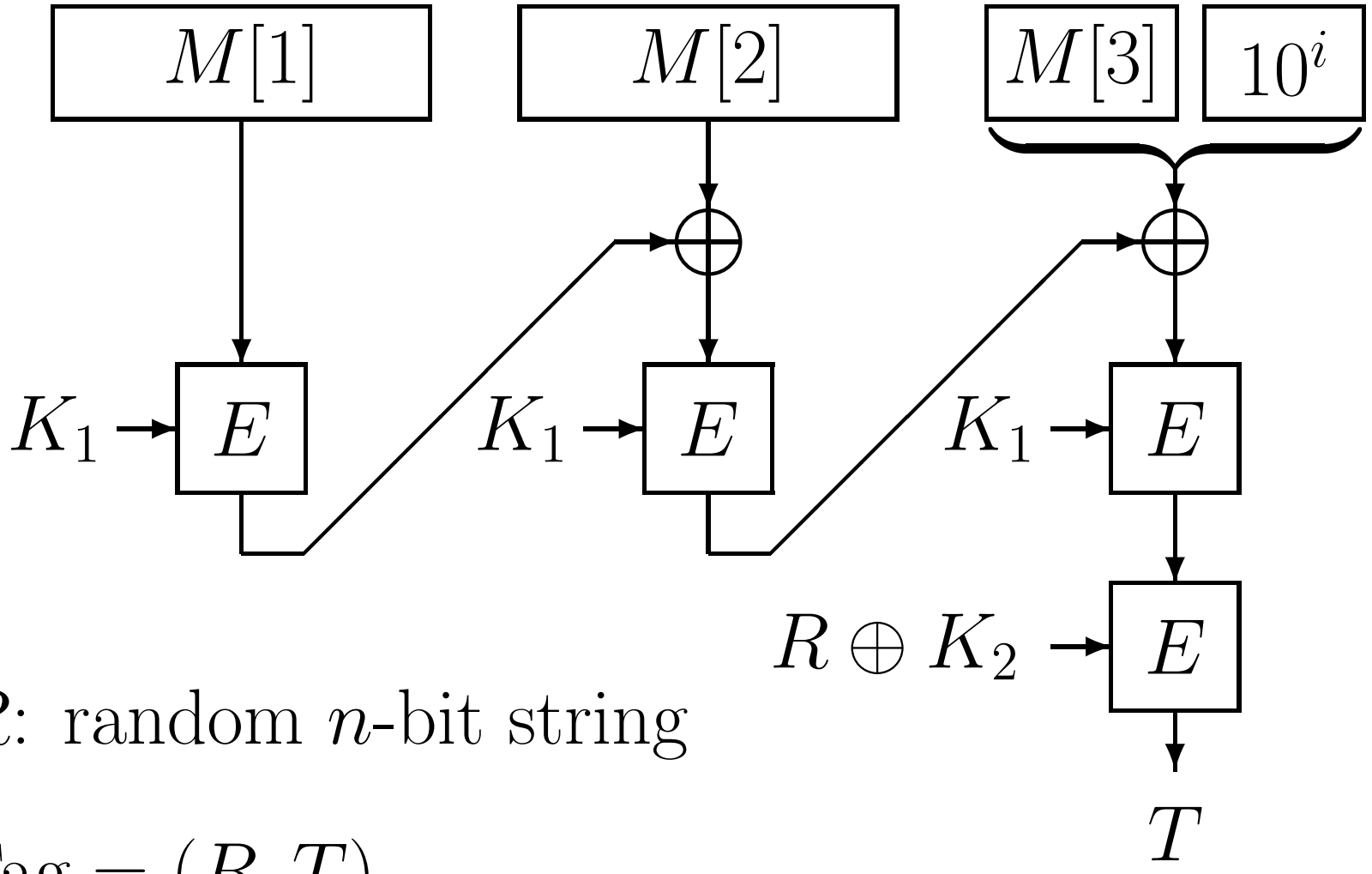
  (up to the birthday paradox limit)

- Allows messages of arbitrary bit length.

Disadvantages of EMAC

- Needs two block cipher key schedulings.
- $m + 2$ block cipher invocations if $|M| = mn$.

RMAC (JJV, FSE '02)



R: random n-bit string

Tag $= (R, T)$

Advantage of RMAC (JJV)

- Security beyond the birthday paradox limit.

Disadvantages of RMAC (JJV)

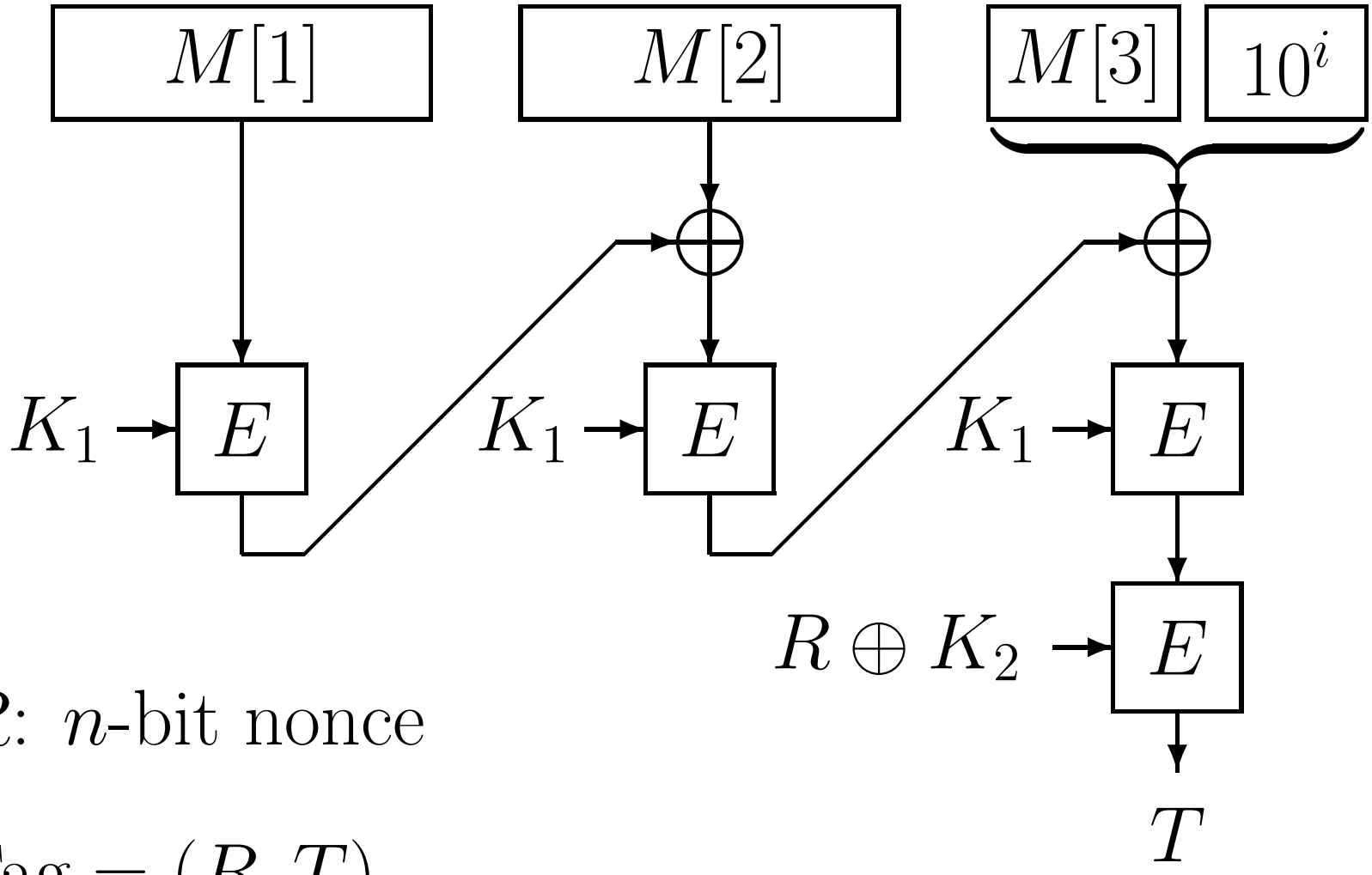
- Security proof needs ideal block cipher.

  (No reduction based security)

- Inefficiency of EMAC.

- Needs random n-bit string for each M.

RMAC (NIST '02)



R: n-bit nonce

Tag $= (R, T)$

Advantage of RMAC (NIST)

- Reduction based proof.

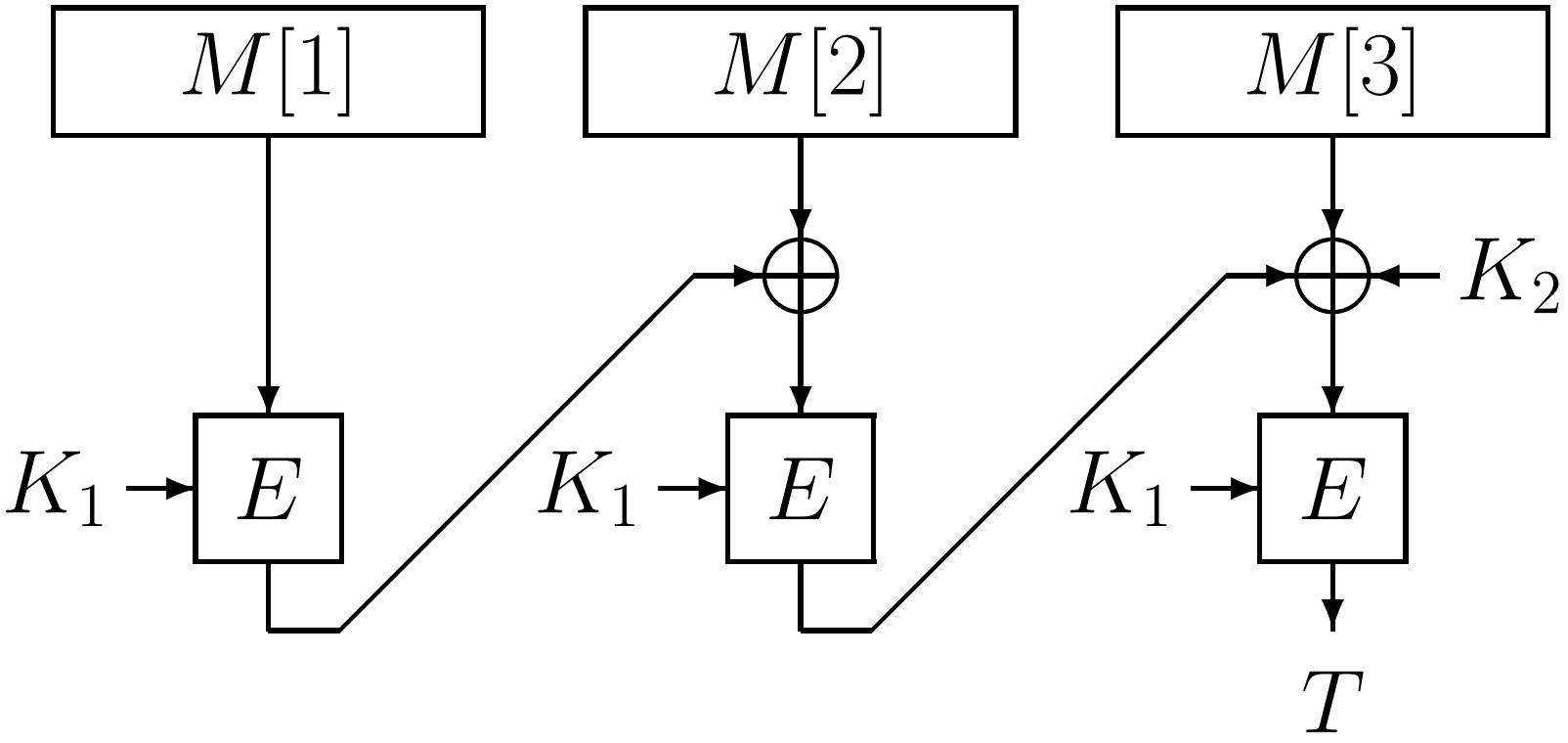
  (up to the birthday paradox limit)

Disadvantage of RMAC (NIST)

- Inefficiency of EMAC.

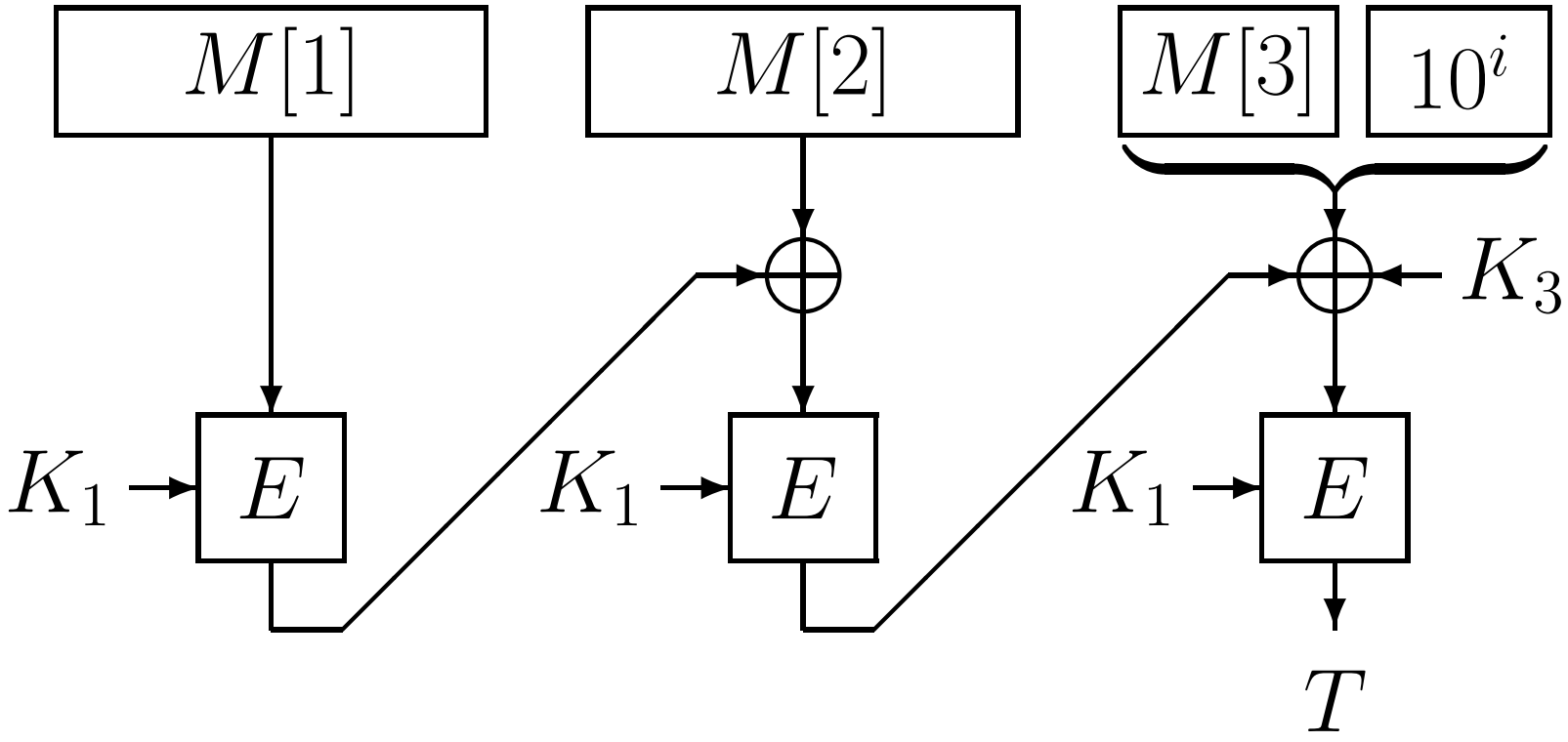
XCBC (Black and Rogaway, Crypto'00)

Case $|M| = mn$ $(m \geq 1)$



XCBC (Black and Rogaway, Crypto'00)

Case $|M| \neq mn$



Advantages of XCBC

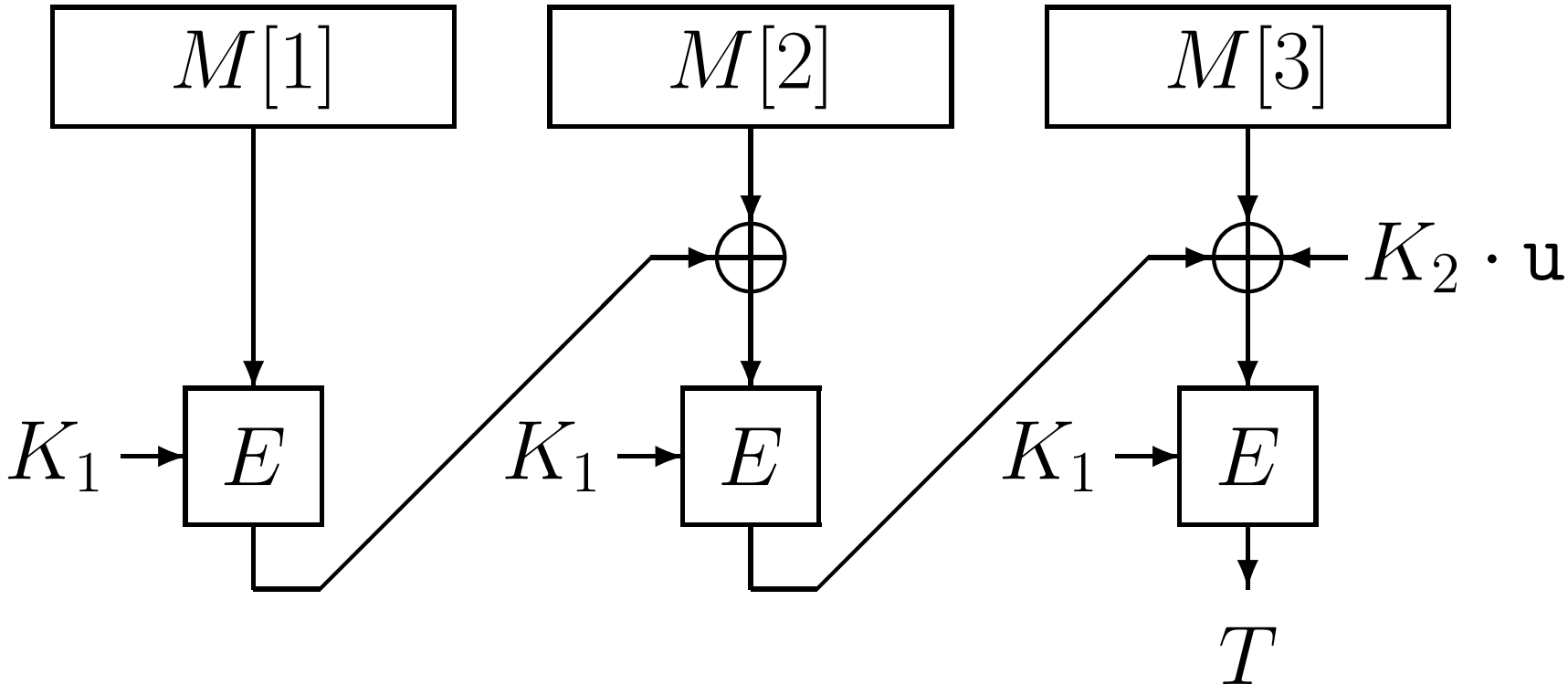
- 1 block cipher key scheduling.
- $\lceil |M|/n \rceil$ block cipher invocations.
- Same security as EMAC.

Disadvantage of XCBC

- Three keys, K_1, K_2, K_3, $(k + 2n)$ bits in total.

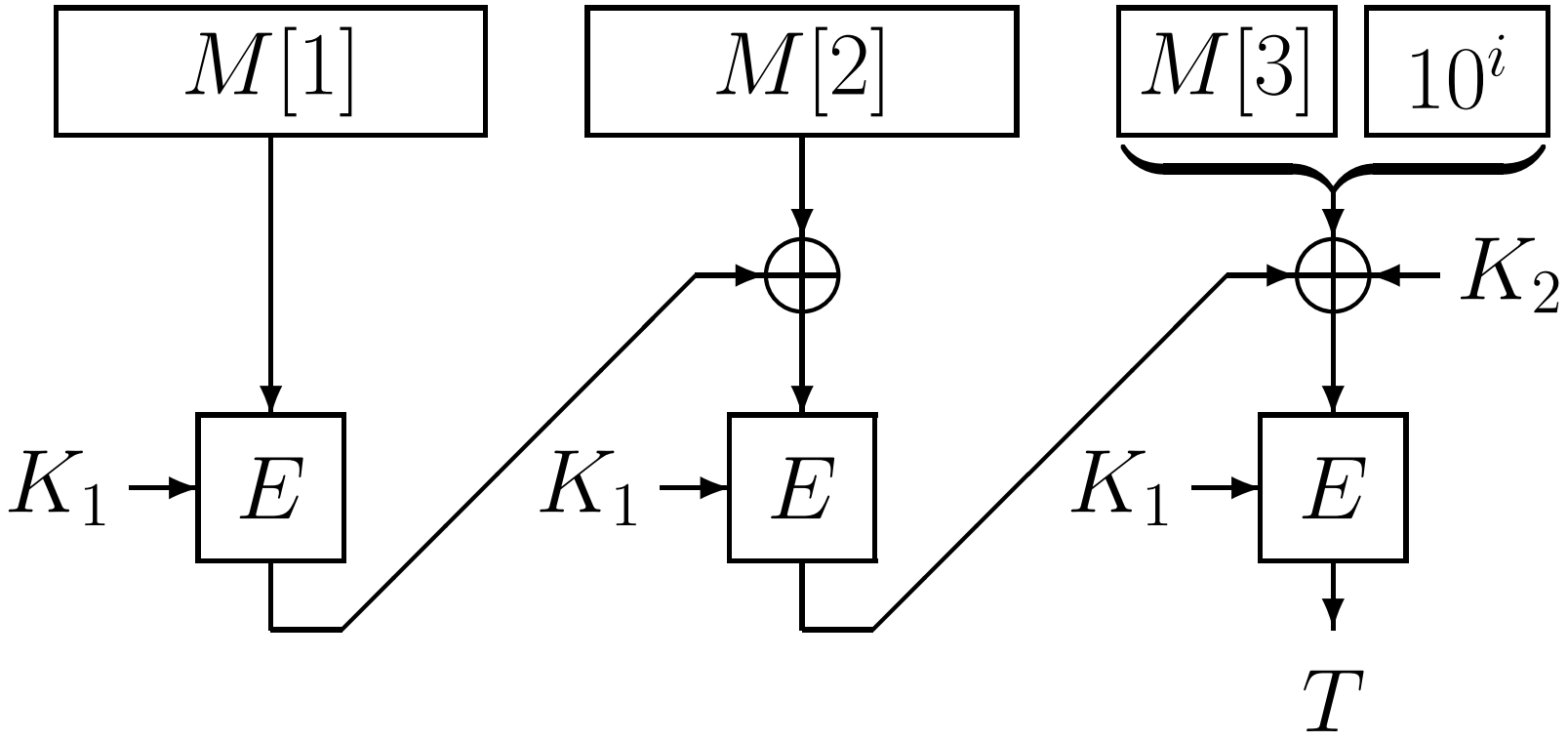
TMAC (Kurosawa and Iwata, RSA '03)

Case $|M| = mn$ $(m \geq 1)$



TMAC (Kuroaswa and Iwata, RSA '03)

Case $|M| \neq mn$



Computation of $K_2 \cdot \mathtt{u}$ ($\mathbf{GF}(2^n)$)

$n = 128$

$a = a_{127}\mathtt{u}^{127} + \cdots + a_1\mathtt{u} + a_0 = (a_{127}, \ldots, a_1, a_0)$

$\mathtt{u} = (0, \ldots, 0, 1, 0)$

$$K_2 \cdot \mathtt{u} = \begin{cases} K_2 \ll 1 & \text{if } \mathtt{msb}(K_2) = 0, \\ (K_2 \ll 1) \oplus \mathtt{Cst} & \text{otherwise.} \end{cases}$$

$\mathtt{Cst} = 0^{120}10000111$

Advantages of TMAC

- Two keys, K_1, K_2, $(k + n)$ bits in total.
- No security loss.
- Negligible cost.

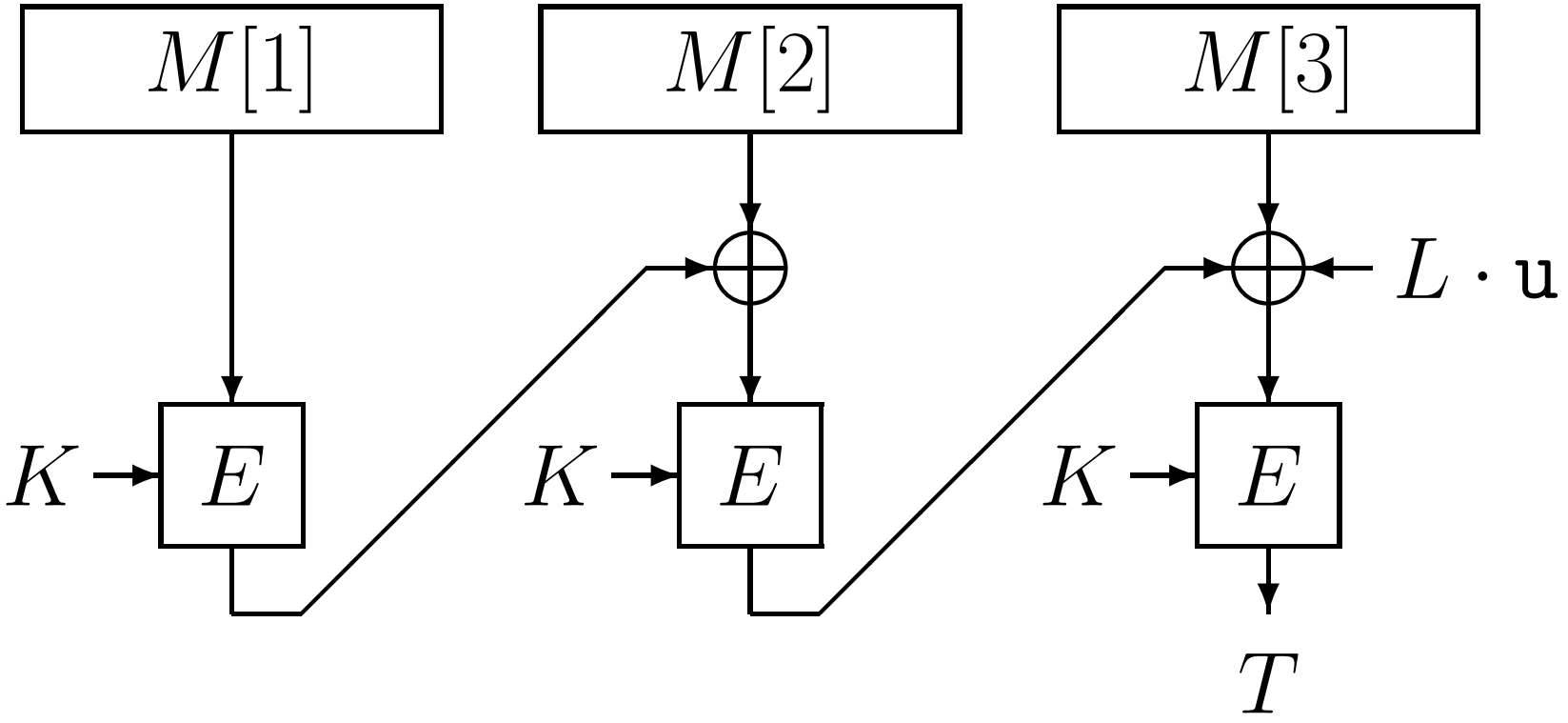
Disadvantage of TMAC

- Two keys, K_1, K_2, $(k + n)$ bits in total.

(longer than CBC MAC)

OMAC1 (Iwata and Kurosawa, FSE '03)

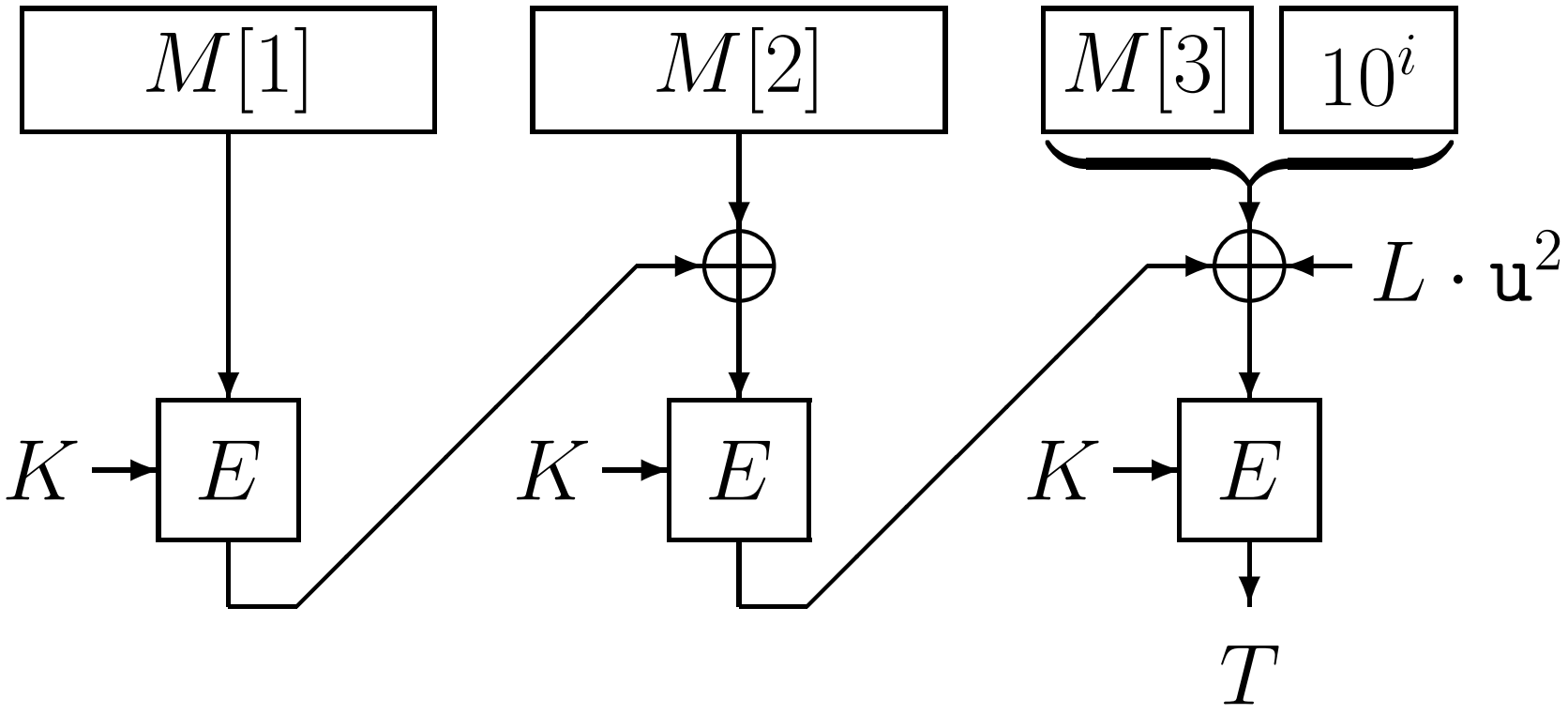
Case $|M| = mn$ $(m \geq 1)$



$$L = E_K(0^n)$$

OMAC1 (Iwata and Kurosawa, FSE '03)

Case $|M| \neq mn$



$$L \cdot \mathrm{u}^2 = (L \cdot \mathrm{u}) \cdot \mathrm{u}$$

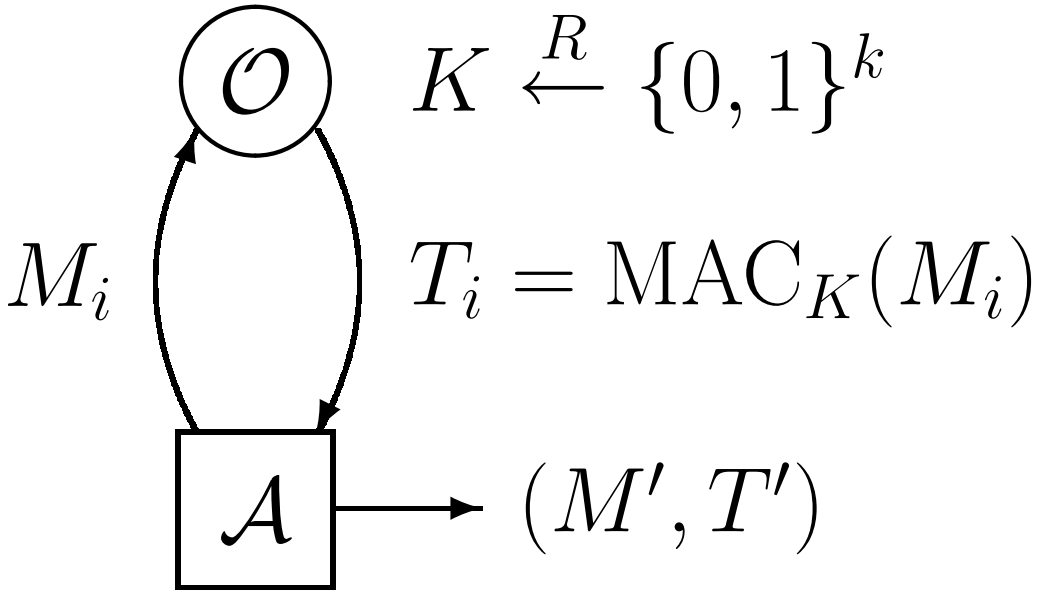
Advantages of OMAC1

- One k-bit key K.
- No security loss.

Disadvantage of OMAC1

- One block cipher invocation to compute L.

Security of a MAC



- $\mathcal{A}$ forges if $T' = \mathcal{O}(M')$, $M' \neq M_i$
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{MAC}_K}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \Pr_K(\mathcal{A} \text{ forges})$

Security of a MAC (Cont.)

$$\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{MAC}}(q,\sigma) \stackrel{\mathrm{def}}{=} \max_{\mathcal{A}} \left\{ \mathtt{Adv}^{\mathsf{mac}}_{\mathrm{MAC}}(\mathcal{A}) \right\},$$

where $\mathcal{A}$ asks at most q queries,

which are at most σ blocks in total.

- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{EMAC}}(q,\sigma) \leq \dfrac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma) + \mathtt{Adv}^{\mathsf{prp}}_{E}(q)$
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{RMAC(NIST)}}(q,\sigma) \leq \dfrac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma) + \mathtt{Adv}^{\mathsf{prp-rka}}_{\Phi_k{}^{\oplus},E}(q)$
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{XCBC}}(q,\sigma) \leq \dfrac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma)$
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{TMAC}}(q,\sigma) \leq \dfrac{3\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma)$
- $\mathtt{Adv}^{\mathsf{mac}}_{\mathrm{OMAC}}(q,\sigma) \leq \dfrac{4\sigma^2+1}{2^n} + \mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma+1)$

- $\mathtt{Adv}^{\mathsf{prp}}_{E}(q)$ is the maximum distinguishing probability between E and a random permutation,
- $\mathtt{Adv}^{\mathsf{prp-rka}}_{\Phi_k{}^{\oplus},E}(q)$ is the maximum distinguishing probability between E and a family of random permutations (ideal block cipher) in the related key attack scenario (BK, Eurocrypt '03).
- The maximum is over all adversaries who make at most q queries.

Discussion

Suppose that $\mathtt{Adv}^{\mathsf{prp}}_{E}(\sigma)$ is small.

- EMAC, XCBC, TMAC and OMAC are secure.
- RMAC may not ($\mathtt{Adv}^{\mathsf{prp-rka}}_{\Phi_k{}^{\oplus},E}(q)$ may be large).

Discussion

Suppose that TDES is used.

- $\mathtt{Adv}^{\mathsf{prp}}_{\mathrm{TDES}}(\sigma)$ is small.
- $\mathtt{Adv}^{\mathsf{prp-rka}}_{{\Phi_k}^{\oplus},\mathrm{TDES}}(q)$ is small.

Security Comparison

- EMAC $\approx$ XCBC $\approx$ TMAC $\approx$ OMAC
- RMAC

Efficiency Comparison

- K len. $\cdots$ key length.
- $\#K$ $\cdots$ Number of block cipher key schedulings.
- $\#E$ invo. $\cdots$ Number of block cipher invocations to compute a tag.
- $\#E$ pre. $\cdots$ Number of block cipher invocations in the pre-processing time.

Efficiency Comparison

Name	K len.	$\#K$	$\#E$ invo.	$\#E$ pre.
CBC	k	1	$\lvert M\rvert/n$	0
EMAC	$2k$	2	$1+\lceil(\lvert M\rvert+1)/n\rceil$	0
RMAC	$2k$	$1+\#M$	$1+\lceil(\lvert M\rvert+1)/n\rceil$	0
XCBC	$k+2n$	1	$\lceil\lvert M\rvert/n\rceil$	0
TMAC	$k+n$	1	$\lceil\lvert M\rvert/n\rceil$	0
OMAC	k	1	$\lceil\lvert M\rvert/n\rceil$	1

OMAC needs one E pre.

Not very significant:

- it can be done in an idle time.
- it is performed infrequently

(compared to MAC generation).

One or two E invo. in EMAC/RMAC is significant.

Key Separation Technique (KST)

EMAC with TDES168.

K: 168 bits $\longrightarrow$ K_1, K_2: 168 bits $\times 2$

$$K_1 \stackrel{\text{def}}{=} \begin{cases} \text{first 168 bits of} \\ \text{TDES}_K(\mathtt{C}_1) \circ \text{TDES}_K(\mathtt{C}_2) \circ \text{TDES}_K(\mathtt{C}_3) \end{cases}$$

$$K_2 \stackrel{\text{def}}{=} \begin{cases} \text{first 168 bits of} \\ \text{TDES}_K(\mathtt{C}_4) \circ \text{TDES}_K(\mathtt{C}_5) \circ \text{TDES}_K(\mathtt{C}_6) \end{cases}$$

for some 64 bit constants $\mathtt{C}_1, \ldots, \mathtt{C}_6$

Efficiency Comparison (TDES & KST)

Name	K len.	$\#K$	$\#E$ invo.	$\#E$ pre.
EMAC	k	3	$1 + \lceil (\lvert M \rvert + 1)/n \rceil$	4 or 6
RMAC	k	$2 + \#M$	$1 + \lceil (\lvert M \rvert + 1)/n \rceil$	4 or 6
XCBC	k	2	$\lceil \lvert M \rvert /n \rceil$	4 or 5
TMAC	k	2	$\lceil \lvert M \rvert /n \rceil$	3 or 4
OMAC	k	1	$\lceil \lvert M \rvert /n \rceil$	1

Efficiency Comparison (AES & KST)

Name	K len.	$\#K$	$\#E$ invo.	$\#E$ pre.
EMAC	k	3	$1 + \lceil (\lvert M \rvert + 1)/n \rceil$	2, 3 or 4
RMAC	k	$2 + \#M$	$1 + \lceil (\lvert M \rvert + 1)/n \rceil$	2, 3 or 4
XCBC	k	2	$\lceil \lvert M \rvert /n \rceil$	3 or 4
TMAC	k	2	$\lceil \lvert M \rvert /n \rceil$	2 or 3
OMAC	k	1	$\lceil \lvert M \rvert /n \rceil$	1

OMAC needs one E pre.

The gain for this cost is its optimal key length, which *completely eliminates* the need for the KST.

- KST is a very error prone process in practice,
- Once KST is used, EMAC, RMAC, XCBC, TMAC have significant costs ($\#K$, $\#E$ pre.).

Conclusion

Use OMAC!

- Good security bound.
- Good efficiency.

Note

- OMAC paper is available at

  `http://eprint.iacr.org/2002/180/`

- NIST's web page is

  `http://csrc.nist.gov/encryption/modes/`

Questions?

Tetsu Iwata

iwata@cis.ibaraki.ac.jp