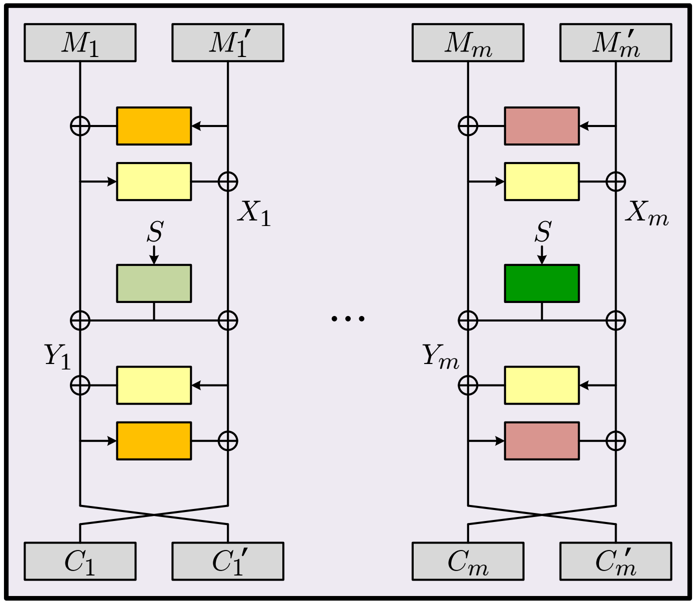


Some thoughts on **AEZ** v. 4

Viet Tung Hoang
Florida State University
USA

Ted Krovetz
Sacramento State
USA

Phillip Rogaway
Univ of California, Davis
USA

DIAC 2016
Nagoya, Japan
27 September 2016

With thanks to
Tetsu Iwata *and*
Shiho Moriai
for organizing this workshop!

Reluctant to give a talk

- **No changes** for Round-3
- **Talks** @ DIAC 2014
  EUROCRYPT 2015
  Several AE survey talks

But some reasons to do so

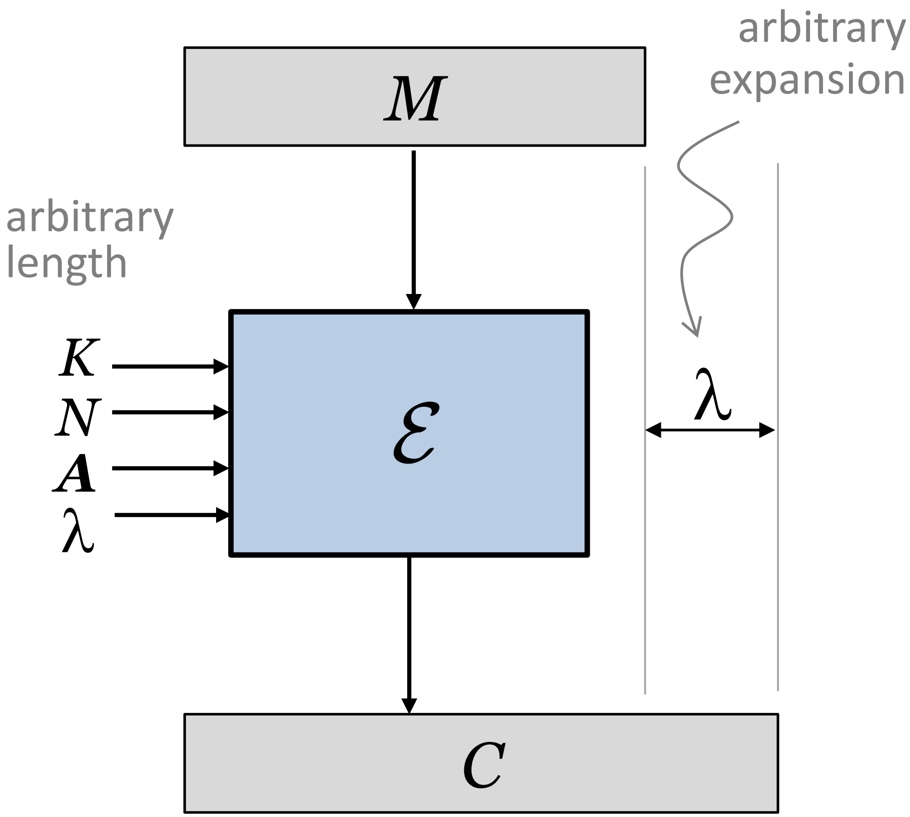
- My view of the mode has evolved
- Attacks @ ASIACRYPT 2015 and at FSE 2017
- AEZ is already in use (should it be?)

What kind of object *is* AEZ ?

- AIL / VIL blockcipher
- Wide-block blockcipher
- An enciphering scheme

An Robust-AE scheme

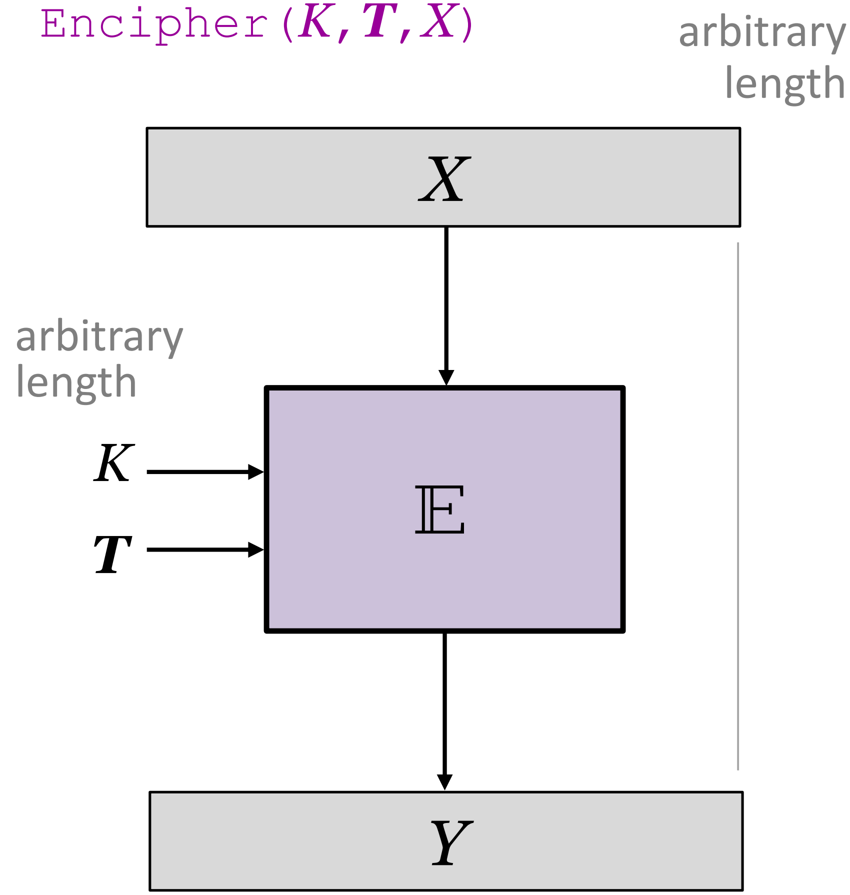
`Encrypt`$(K, N, \boldsymbol{A}, M, \lambda)$



Should look like a uniform λ-**expanding injection** (ind for N, $\boldsymbol{A}$, λ) (forward + backward oracles)

A Generalized Blockcipher

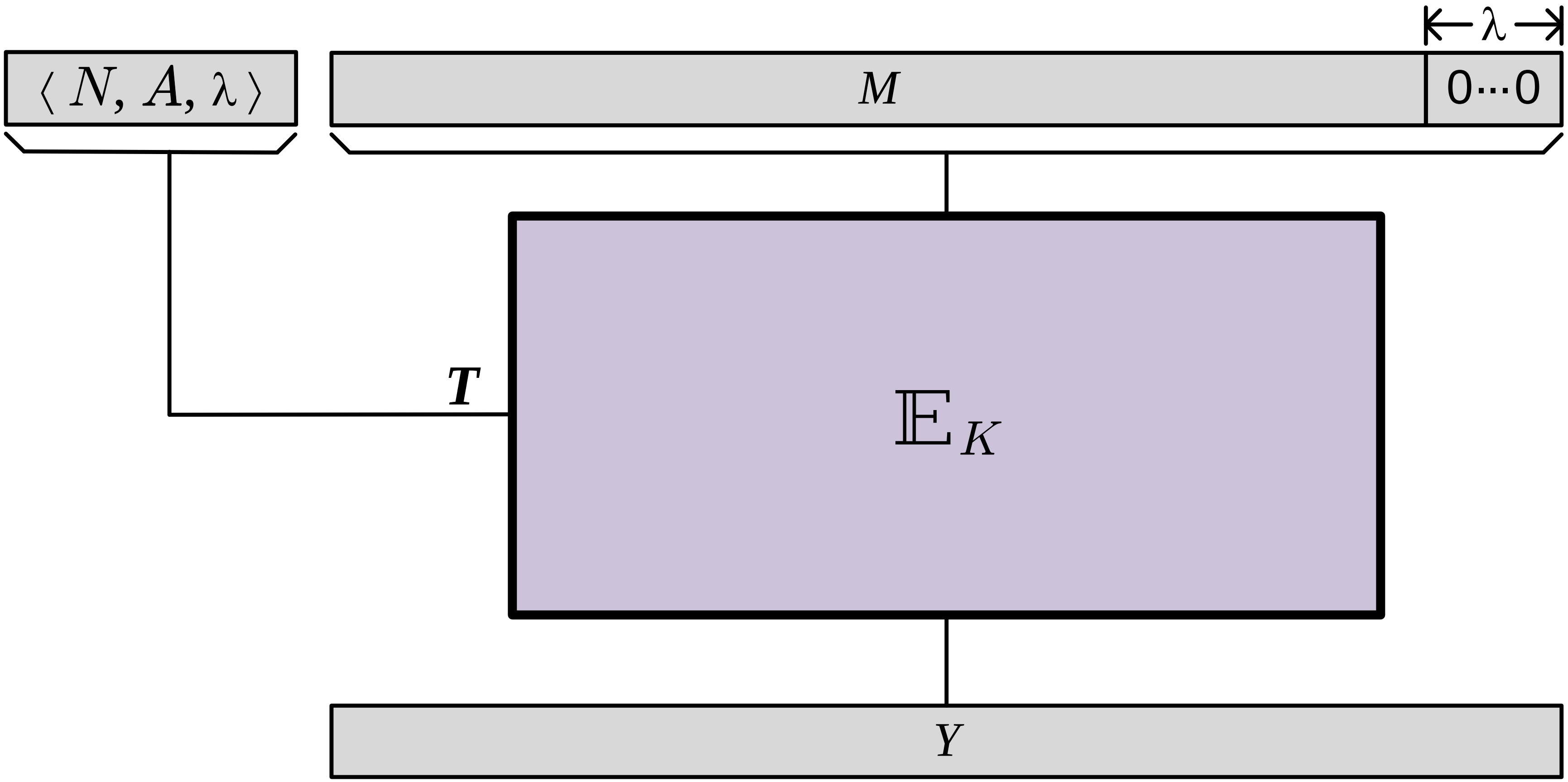
`Encipher`$(K, \boldsymbol{T}, X)$



Should look like a uniform **permutation** (ind for all $\boldsymbol{T}$) (forward and backward oracles)

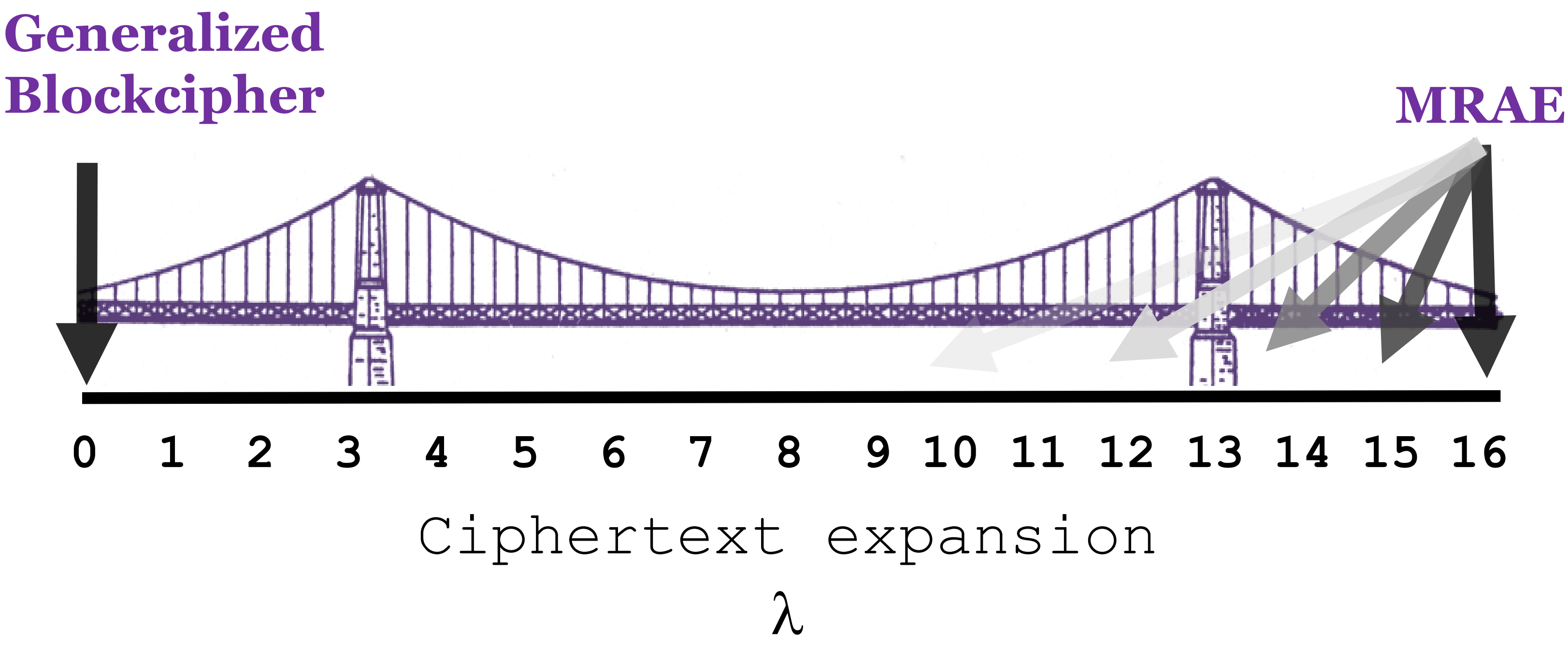
Robust-AE ⇔ Generalized Blockcipher

Following [BR00, ST13]



The natural construction, "enciphering-based AE," to make an RAE scheme from a generalized blockcipher

Unifying MRAE and Blockciphers



Claims lurking behind AEZ

(1) Enciphering-based AE is a great way to achieve AE: very strong properties – not necessarily expensive

a) If (M, $\boldsymbol{A}$) tuples are known not to repeat, no nonce is needed
b) Nonce repetitions: privacy loss is limited to revealing repetitions in (N, $\boldsymbol{A}$, M) tuples, authenticity not damaged at all.
c) Any authenticator-length can be selected, achieving best-possible authenticity for this amount of stretch.
d) If there's redundancy in plaintexts whose presence is verified on decryption, this augments authenticity
e) By last two properties: one can minimize length-expansion for bandwidth-constrained apps
f) If a decrypting party leaks some or all of a putative plaintext that was supposed to be squelched because of an authenticity-check failure, no problem.

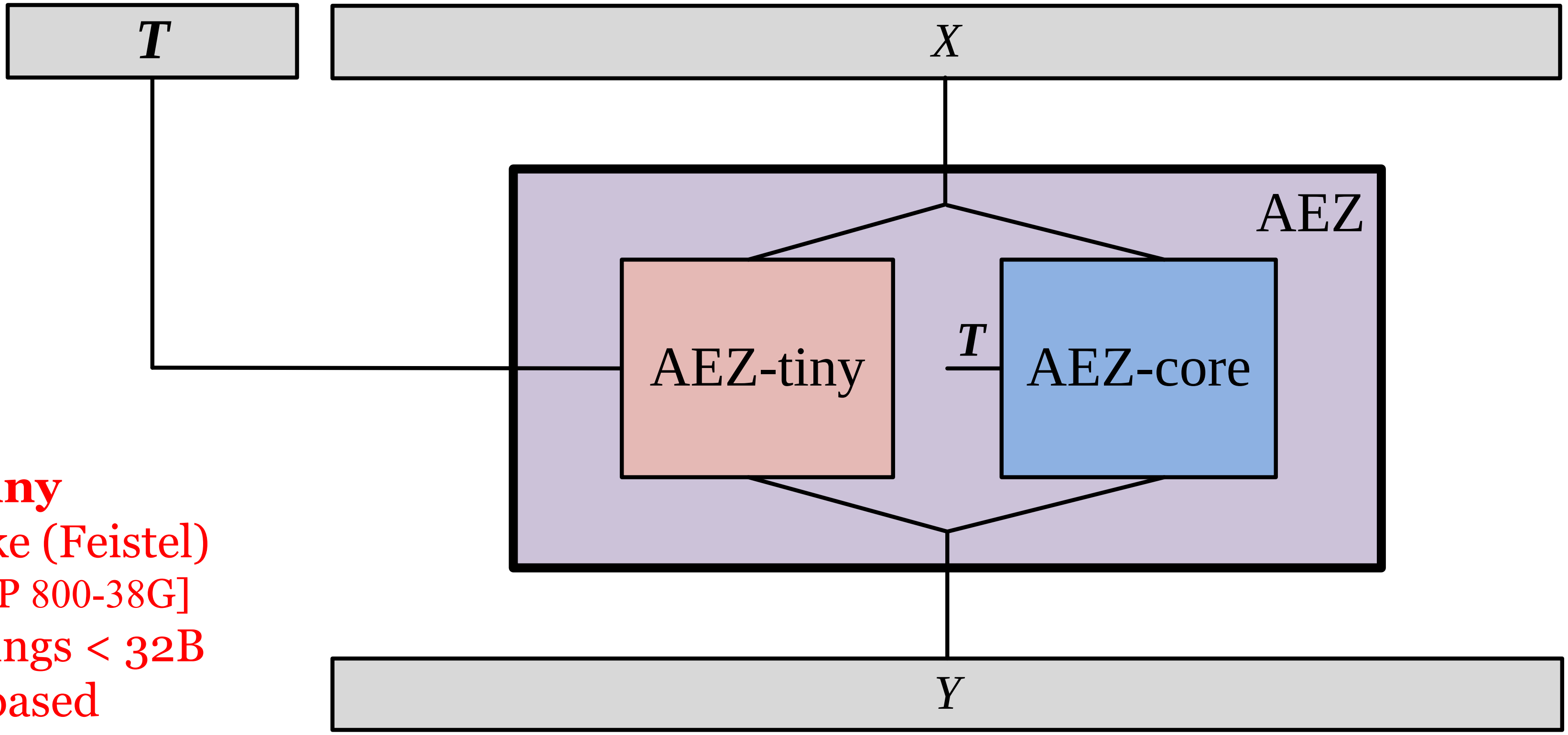
(2) A generalized blockcipher is a great tool to have around

Conceptual simplicity and versatility: it's an AE scheme, a PRG, a MAC, a PRF, a hash function, an entropy extractor, …

AEZ

The first concrete construction of a generalized blockcipher

(although VIL wide-block blockciphers like **EME2** [Halevi; Halevi-Rogaway] come very close)

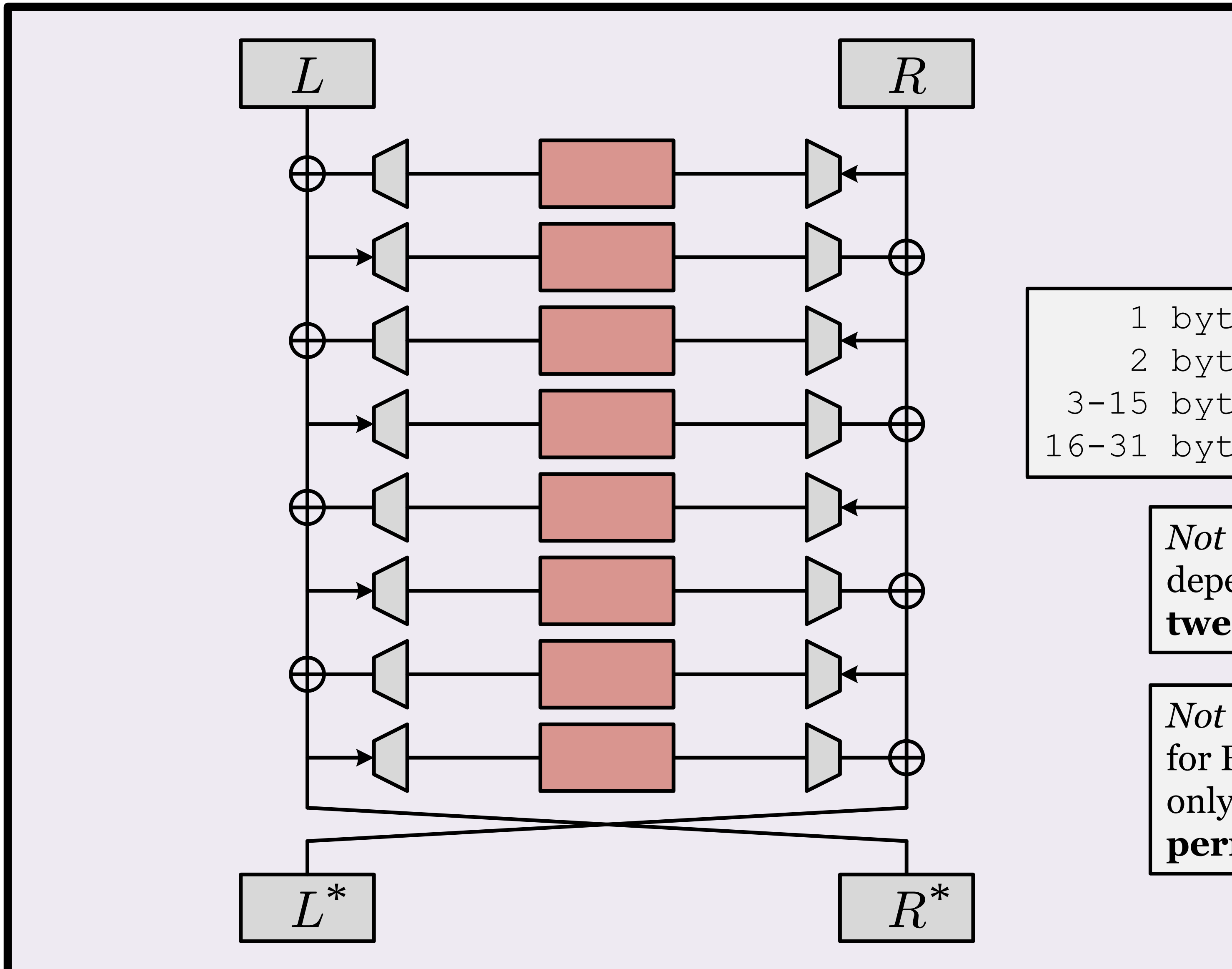


AEZ-tiny
FFX-like (Feistel)
[NIST SP 800-38G]
For strings < 32B
AES4-based

Structure of AEZ

AEZ-core
Builds on EME [HR04]
and OTR [M14]
For strings ≥ 32B
AES4 & AES based.

AEZ-tiny

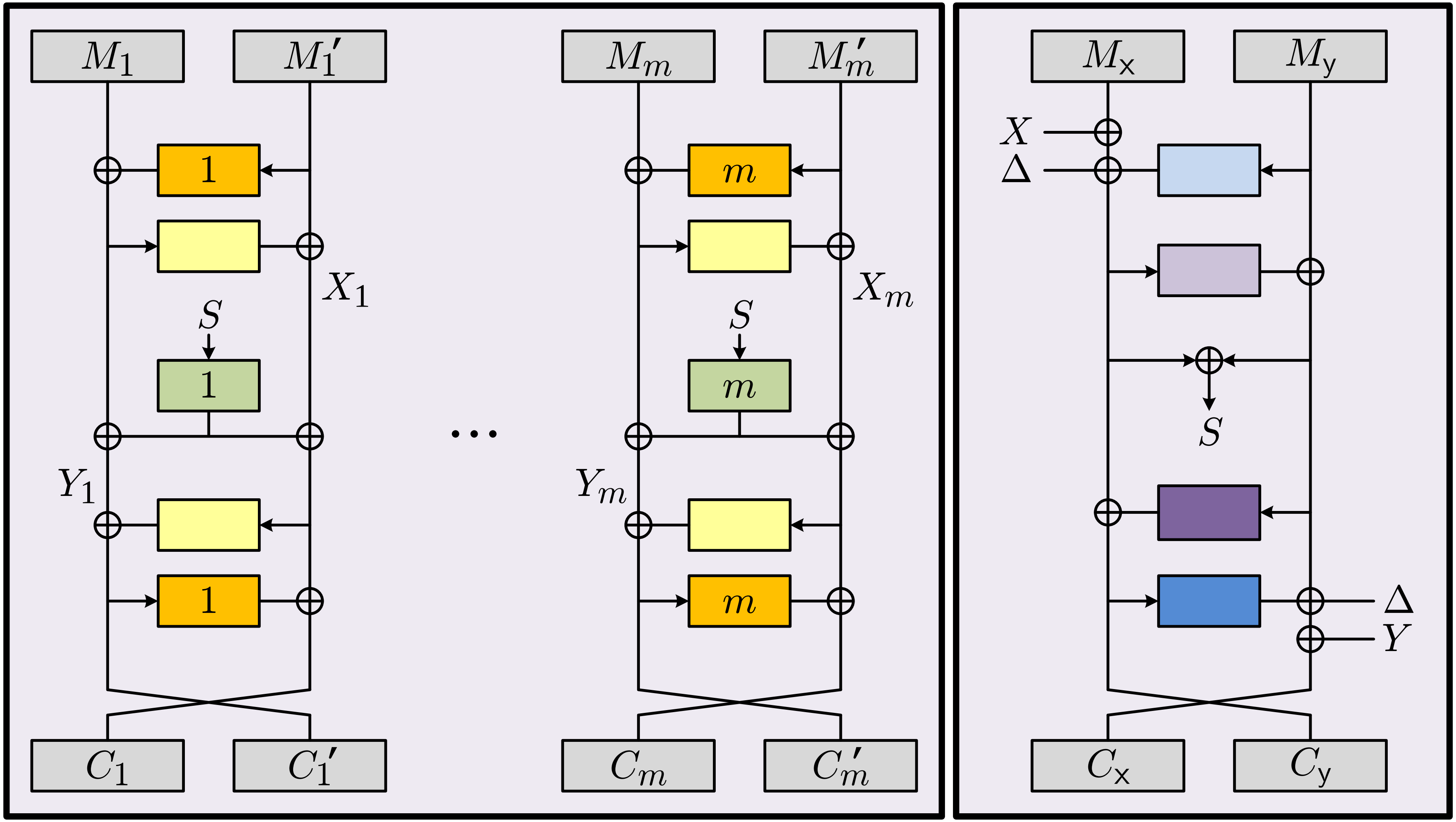


1 byte:	24 rounds
2 bytes:	16 rounds
3-15 bytes:	10 rounds
16-31 bytes:	8 rounds

Not shown: each round depends on the hashed **tweak**

Not shown: we correct for Feistel networks only generating **even permutations**

AEZ-core



M_1
M_1'
M_m
M_m'
M_x
M_y
1
m
X_1
X_m
X
Δ
S
1
m
S
Y_1
Y_m
1
m
Δ
Y
C_1
C_1'
C_m
C_m'
C_x
C_y

What's to Like?

Defense-in-depth *and* good speed

- The **security target**. Robust-AE is a **very strong** notion – implies almost all security properties one might hope for. **Very few** MRAE schemes remain in round-3.

- Wonderful **versatility, ease of use** – arbitrary-length keys, arbitrary ciphertext expansion, single-version scheme

- Amazing **speed** (in SW with AES-NI: peak 0.63 cpb Skylake; 1.0 AES-equivalents/block) considering the goal. Two-pass schemes are *not* inherently slow. HW performance looks respectable. **Quick-rejection** of invalid messages

- A **proof** for AEZ-core, to the birthday bound, in the **prove-then-prune** paradigm

What's <u>not</u> to Like?

- Scheme is **very complex**. *Anything-but-EZ* in HW… and not easy for the SW, either. 58 lines of dense pseudocode.

- Aggressively optimized – **not** a conservative design.

- There are **birthday key-recovery attacks**: [Chaigneau, Gilbert 2017] ($2^{66.5}$ chosen plaintexts) (v.4), following [Fuhr, Leurent, Suder 2015] (v.3). Note: 2^{48} byte usage cap.

- A **prove-then-prune** proof does not, by itself, imply security; **cryptanalysis is still needed**. Should not be treated as a proof in the same sense as assuming some primitive is a PRP.

- Are the RAE \ MRAE properties (particularly the possibility of small ciphertext expansion) **useful**?

Seduced by speed?

"*Don't worry about speed*. An RAE scheme / generalized blockcipher is very strong goal, and a scheme achieving it based on `aesenc` is going to need to be 2× – 3× slower, per block, than AES."

"No! We can **match** AES's speed in an RAE scheme. We can even get features like fast-reject and encipher-direction only processing, at the same time." → **AEZ**

"No!! We should be able to **exceed** AES speed in an aesenc-based MRAE scheme, and even an RAE scheme. What goes for AEGIS/Tioxin can be made to fly here, too."

For in the future, I'd like to see

A generalized blockcipher / RAE scheme

that's **much simpler** than AEZ, yet

Maybe a healthier alternative:

Enjoys (good old-fashioned) provable security

(DJB "boring crypto")

Is just as fast, or faster

Feels more conservative

Apparently has BBB security

Has *at least* an ideal-permutation model proof of security, with good bounds

But, for now: AEZ is the best there is for this degree of versatility and defense in depth.

AEZ (v4)

algorithm Encrypt$(K, N, \boldsymbol{A}, \tau, M)$ // AEZ authenticated encryption
$X \leftarrow M \parallel 0^\tau$; $(A_1, \ldots, A_a) \leftarrow \boldsymbol{A}$
$\boldsymbol{T} \leftarrow ([\tau]_{128}, N, A_1, \ldots, A_a)$
if $M = \varepsilon$ **then return** AEZ-prf$(K, \boldsymbol{T}, \tau)$
return Encipher$(K, \boldsymbol{T}, X)$

algorithm Decrypt$(K, N, \boldsymbol{A}, \tau, C)$ // AEZ authenticated decryption
$(A_1, \ldots, A_a) \leftarrow \boldsymbol{A}$; $\boldsymbol{T} \leftarrow ([\tau]_{128}, N, A_1, \ldots, A_a)$
if $|C| < \tau$ **then return** $\perp$
if $|C| = \tau$ **then if** $C =$ AEZ-prf$(K, \boldsymbol{T}, \tau)$ **then return** ε **else return** $\perp$ **fi fi**
$X \leftarrow$ Decipher$(K, \boldsymbol{T}, C)$
$M \parallel Z \leftarrow X$ where $|Z| = \tau$
if $(Z = 0^\tau)$ **then return** M **else return** $\perp$

algorithm Encipher$(K, \boldsymbol{T}, X)$ // AEZ enciphering
if $|X| < 256$ **then return** Encipher-AEZ-tiny$(K, \boldsymbol{T}, X)$
if $|X| \geq 256$ **then return** Encipher-AEZ-core$(K, \boldsymbol{T}, X)$

algorithm Encipher-AEZ-tiny$(K, \boldsymbol{T}, M)$ // AEZ-tiny enciphering
$\mu \leftarrow |M|$; $n \leftarrow \mu/2$; $\Delta \leftarrow$ AEZ-hash$(K, \boldsymbol{T})$
if $\mu = 8$ **then** $k \leftarrow 24$ **else if** $\mu = 16$ **then** $k \leftarrow 16$ **else if** $\mu < 128$ **then** $k \leftarrow 10$ **else** $k \leftarrow 8$ **fi**
$L \leftarrow M[1..n]$; $R \leftarrow M[n+1..\mu]$; **if** $\mu \geq 128$ **then** $i \leftarrow 6$ **else** $i \leftarrow 7$ **fi**
for $j \leftarrow 0$ **to** $k-1$ **do** $R' \leftarrow L \oplus ((\mathrm{E}_K^{0,i}(\Delta \oplus R10^* \oplus [j]_{128}))[1..n])$; $L \leftarrow R$; $R \leftarrow R'$ **od**
$C \leftarrow R \,\|\, L$; **if** $\mu < 128$ **then** $C \leftarrow C \oplus (\mathrm{E}_K^{0,3}(\Delta \oplus (C0^* \vee 10^*)) \wedge 10^*)$ **fi**
return C

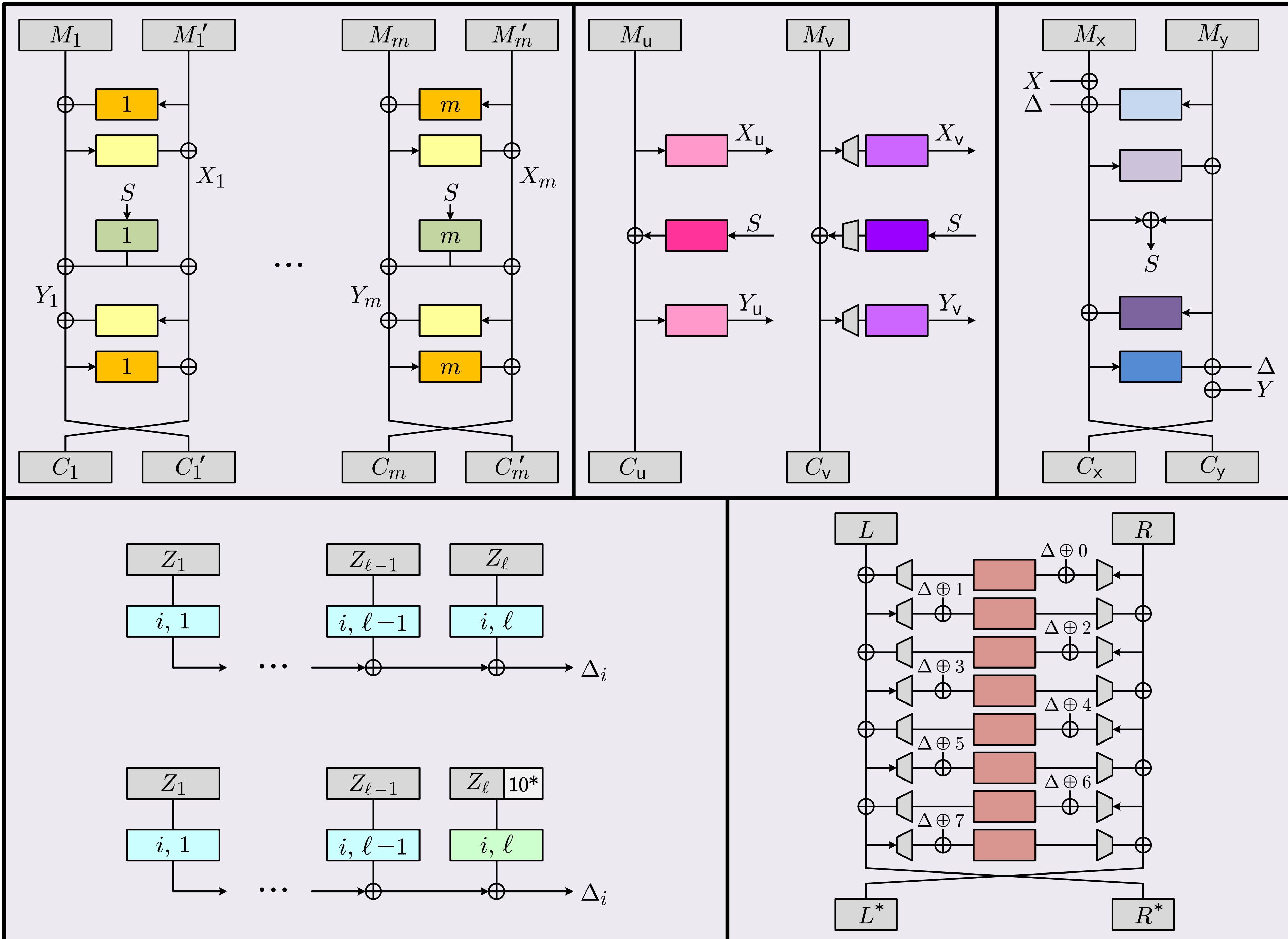
algorithm Encipher-AEZ-core$(K, \boldsymbol{T}, M)$ // AEZ-core enciphering
$\Delta \leftarrow$ AEZ-hash$(K, \boldsymbol{T})$
$M_1 M_1' \cdots M_m M_m'\ M_{\mathsf{uv}}\ M_{\mathsf{x}} M_{\mathsf{y}} \leftarrow M$ where $|M_1| = \cdots = |M_m'| = |M_{\mathsf{x}}| = |M_{\mathsf{y}}| = 128$ and $|M_{\mathsf{uv}}| < 256$
$d \leftarrow |M_{\mathsf{uv}}|$; **if** $d \leq 127$ **then** $M_{\mathsf{u}} \leftarrow M_{\mathsf{uv}}$; $M_{\mathsf{v}} \leftarrow \varepsilon$ **else** $M_{\mathsf{u}} \leftarrow M_{\mathsf{uv}}[1..128]$; $M_{\mathsf{v}} \leftarrow M_{\mathsf{uv}}[129..|M_{\mathsf{uv}}|]$ **fi**
for $i \leftarrow 1$ **to** m **do** $W_i \leftarrow M_i \oplus \mathrm{E}_K^{1,i}(M_i')$; $X_i \leftarrow M_i' \oplus \mathrm{E}_K^{0,0}(W_i)$ **od**
if $d = 0$ **then** $X \leftarrow X_1 \oplus \cdots \oplus X_m \oplus \mathbf{0}$ **else if** $d \leq 127$ **then** $X \leftarrow X_1 \oplus \cdots \oplus X_m \oplus \mathrm{E}_K^{0,4}(M_{\mathsf{u}}10^*)$
else $X \leftarrow X_1 \oplus \cdots \oplus X_m \oplus \mathrm{E}_K^{0,4}(M_{\mathsf{u}}) \oplus \mathrm{E}_K^{0,5}(M_{\mathsf{v}}10^*)$ **fi**
$S_{\mathsf{x}} \leftarrow M_{\mathsf{x}} \oplus \Delta \oplus X \oplus \mathrm{E}_K^{0,1}(M_{\mathsf{y}})$; $S_{\mathsf{y}} \leftarrow M_{\mathsf{y}} \oplus \mathrm{E}_K^{-1,1}(S_{\mathsf{x}})$; $S \leftarrow S_{\mathsf{x}} \oplus S_{\mathsf{y}}$
for $i \leftarrow 1$ **to** m **do** $S' \leftarrow \mathrm{E}_K^{2,i}(S)$; $Y_i \leftarrow W_i \oplus S'$; $Z_i \leftarrow X_i \oplus S'$; $C_i' \leftarrow Y_i \oplus \mathrm{E}_K^{0,0}(Z_i)$; $C_i \leftarrow Z_i \oplus \mathrm{E}_K^{1,i}(C_i')$ **od**
if $d = 0$ **then** $C_{\mathsf{u}} \leftarrow C_{\mathsf{v}} \leftarrow \varepsilon$; $Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus \mathbf{0}$
else if $d \leq 127$ **then** $C_{\mathsf{u}} \leftarrow M_{\mathsf{u}} \oplus \mathrm{E}_K^{-1,4}(S)$; $C_{\mathsf{v}} \leftarrow \varepsilon$; $Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus \mathrm{E}_K^{0,4}(C_{\mathsf{u}}10^*)$
else $C_{\mathsf{u}} \leftarrow M_{\mathsf{u}} \oplus \mathrm{E}_K^{-1,4}(S)$; $C_{\mathsf{v}} \leftarrow M_{\mathsf{v}} \oplus \mathrm{E}_K^{-1,5}(S)$; $Y \leftarrow Y_1 \oplus \cdots \oplus Y_m \oplus \mathrm{E}_K^{0,4}(C_{\mathsf{u}}) \oplus \mathrm{E}_K^{0,5}(C_{\mathsf{v}}10^*)$ **fi**
$C_{\mathsf{y}} \leftarrow S_{\mathsf{x}} \oplus \mathrm{E}_K^{-1,2}(S_{\mathsf{y}})$; $C_{\mathsf{x}} \leftarrow S_{\mathsf{y}} \oplus \Delta \oplus Y \oplus \mathrm{E}_K^{0,2}(C_{\mathsf{y}})$
return $C_1 C_1' \cdots C_m C_m'\ C_{\mathsf{u}} C_{\mathsf{v}}\ C_{\mathsf{x}} C_{\mathsf{y}}$

algorithm AEZ-hash$(K, \boldsymbol{T})$ // AXU hash. T is a vector of strings
$(T_1, \ldots, T_t) \leftarrow \boldsymbol{T}$
for $i \leftarrow 1$ **to** t **do**
  $\ell \leftarrow \max(1, \lceil |T_i|/128 \rceil)$; $j \leftarrow i + 2$; $Z_1 \cdots Z_\ell \leftarrow T_i$ where $|Z_1| = \cdots = |Z_{\ell-1}| = 128$
  if $|Z_\ell| = 128$ **then** $\Delta_i \leftarrow \mathrm{E}_K^{j,1}(Z_1) \oplus \cdots \oplus \mathrm{E}_K^{j,\ell}(Z_\ell)$ **fi**
  if $|Z_\ell| < 128$ **then** $\Delta_i \leftarrow \mathrm{E}_K^{j,1}(Z_1) \oplus \cdots \oplus \mathrm{E}_K^{j,\ell-1}(Z_{\ell-1}) \oplus \mathrm{E}_K^{j,0}(Z_\ell 10^*)$ **fi**
return $\Delta_1 \oplus \cdots \oplus \Delta_t \oplus \mathbf{0}$

algorithm AEZ-prf$(K, \boldsymbol{T}, \tau)$ // PRF used when $M = \varepsilon$
$\Delta \leftarrow$ AEZ-hash$(K, \boldsymbol{T})$
return $(\mathrm{E}_K^{-1,3}(\Delta) \,\|\, \mathrm{E}_K^{-1,3}(\Delta \oplus [1]_{128}) \,\|\, \mathrm{E}_K^{-1,3}(\Delta \oplus [2]_{128}) \,\|\, \cdots)[1..\tau]$

algorithm $\mathrm{E}_K^{j,i}(X)$ // Scaled-down TBC
$I \,\|\, J \,\|\, L \leftarrow \mathrm{Extract}(K)$ where $|I| = |J| = |L| = 128$
$\boldsymbol{K} \leftarrow (\mathbf{0}, I, J, L, I, J, L, I, J, L, I)$
if $j = -1$ **then** $\Delta \leftarrow iJ$; **return** $\mathrm{AES10}_{\boldsymbol{K}}(X \oplus \Delta)$ **fi**
$\boldsymbol{k} \leftarrow \boldsymbol{k}_1 \leftarrow (\mathbf{0}, J, I, L, \mathbf{0})$; $\boldsymbol{k}_2 \leftarrow (\mathbf{0}, L, I, J, L)$
if $j = 0$ **then** $\Delta \leftarrow iI$; **return** $\mathrm{AES4}_{\boldsymbol{k}}(X \oplus \Delta)$ **fi**
if $1 \le j \le 2$ **then** $\Delta \leftarrow (2^{3+\lfloor (i-1)/8 \rfloor} + ((i-1) \bmod 8))I$; **return** $\mathrm{AES4}_{\boldsymbol{k}_j}(X \oplus \Delta)$ **fi**
if $j \ge 3$ **and** $i = 0$ **then** $\Delta \leftarrow 2^{j-3} \cdot L$; **return** $\mathrm{AES4}_{\boldsymbol{k}}(X \oplus \Delta) \oplus \Delta$ **fi**
if $j \ge 3$ **and** $i \ge 1$ **then** $\Delta \leftarrow 2^{j-3} \cdot L \oplus (2^{3+\lfloor (i-1)/8 \rfloor} + (i - 1 \bmod 8))J$; **return** $\mathrm{AES4}_{\boldsymbol{k}}(X \oplus \Delta) \oplus \Delta$ **fi**

algorithm Extract(K) // Map key to subkeys
if $|K| = 384$ **then return** K
else return BLAKE2b(K)



M1
M1′
Mm
Mm′
1
m
X1
Xm
S
Y1
Ym
C1
C1′
Cm
Cm′
···
Mu
Mv
Xu
Xv
S
Yu
Yv
Cu
Cv
Mx
My
X
Δ
S
Δ
Y
Cx
Cy
Z1
Zℓ−1
Zℓ
i, 1
i, ℓ−1
i, ℓ
···
Δi
10*
L
R
Δ ⊕ 0
Δ ⊕ 1
Δ ⊕ 2
Δ ⊕ 3
Δ ⊕ 4
Δ ⊕ 5
Δ ⊕ 6
Δ ⊕ 7
L*
R*