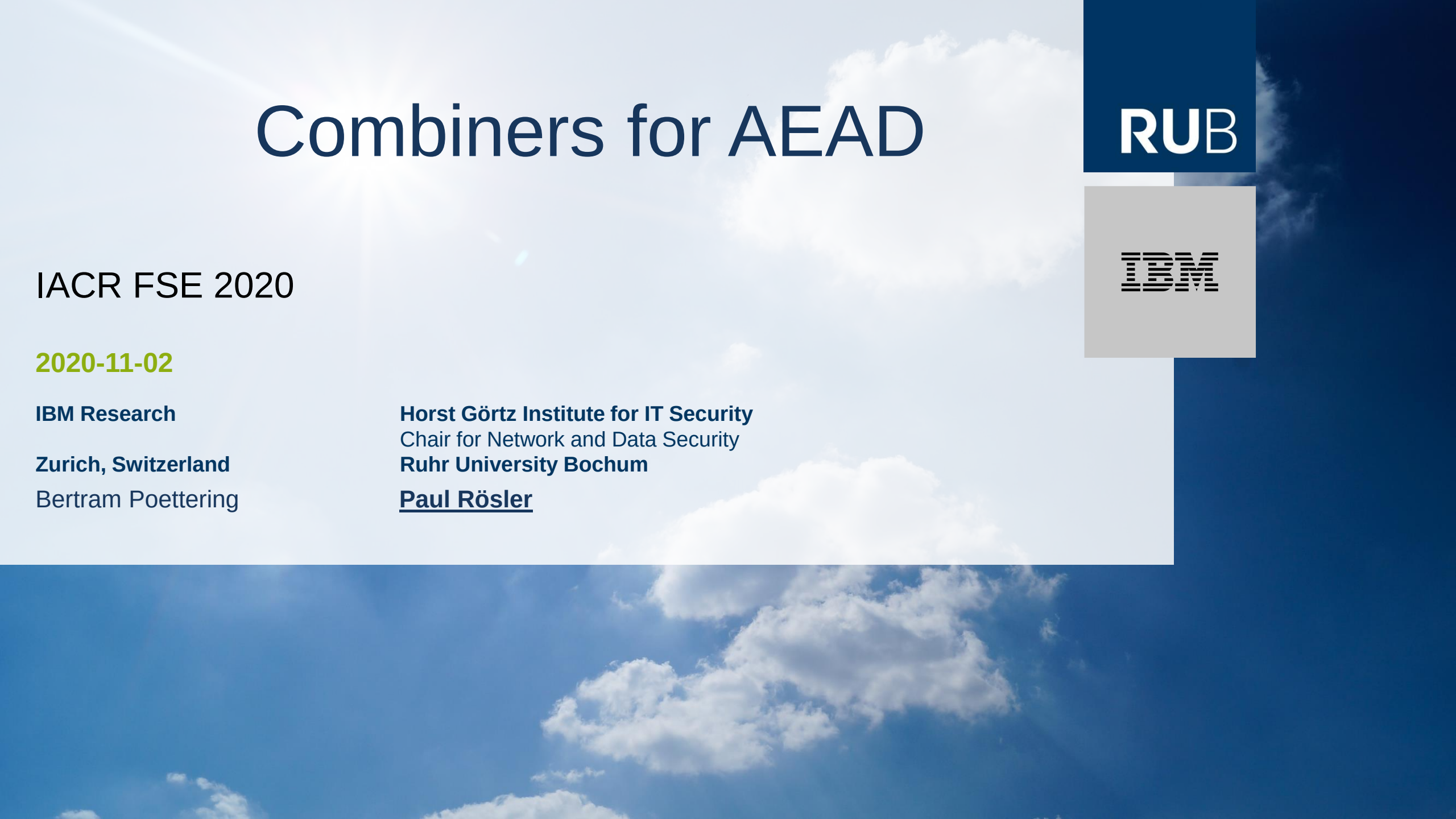
Combiners for AEAD



IACR FSE 2020

2020-11-02

IBM Research

Zurich, Switzerland

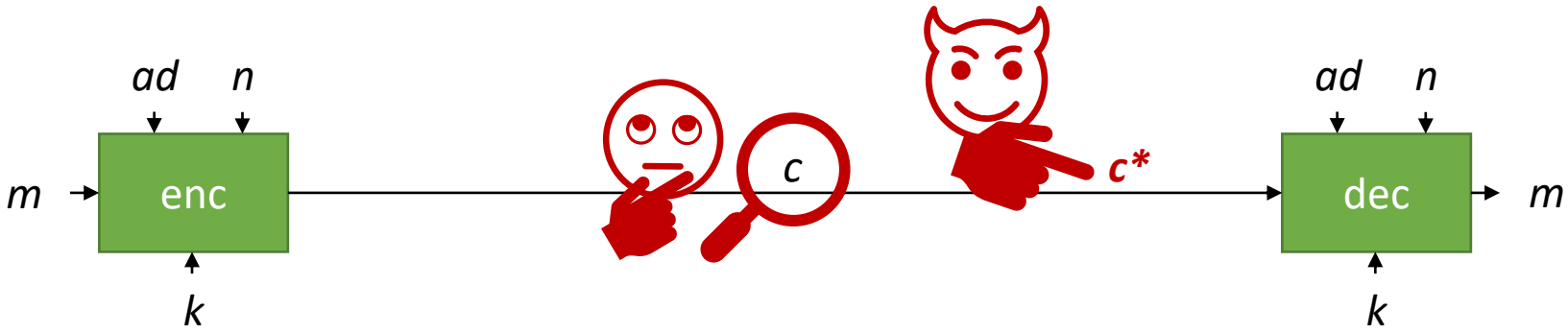
Bertram Poettering

Horst Görtz Institute for IT Security
Chair for Network and Data Security
Ruhr University Bochum

Paul Rösler

AEAD

- Symmetric encryption and authentication of data



  - Confidentiality via indistinguishability of ciphertexts

$$\mathrm{Adv} = |\Pr[\mathcal{A}(c) \to_\$ 0 : c \leftarrow \mathrm{enc}(k, m, ad, n)] - \Pr[\mathcal{A}(c) \to_\$ 0 : c \leftarrow_\$ \{0,1\}^{|c'|}, c' \leftarrow \mathrm{enc}(k, m, ad, n)]|$$

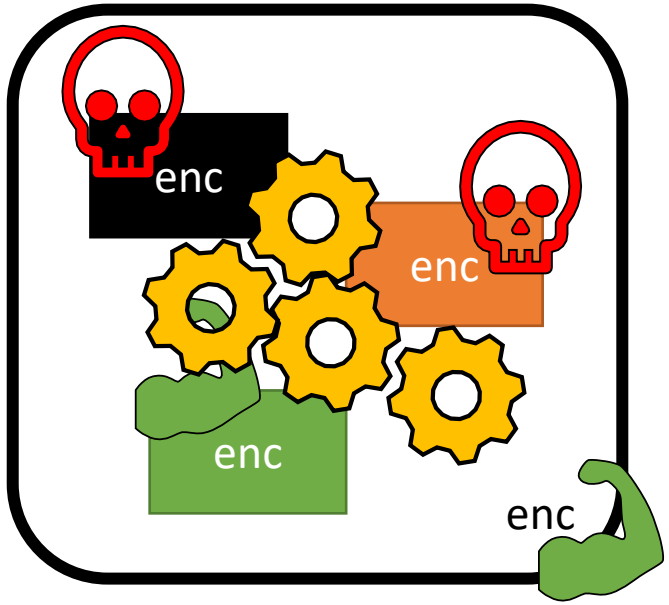
  - Authenticity via integrity of ciphertexts

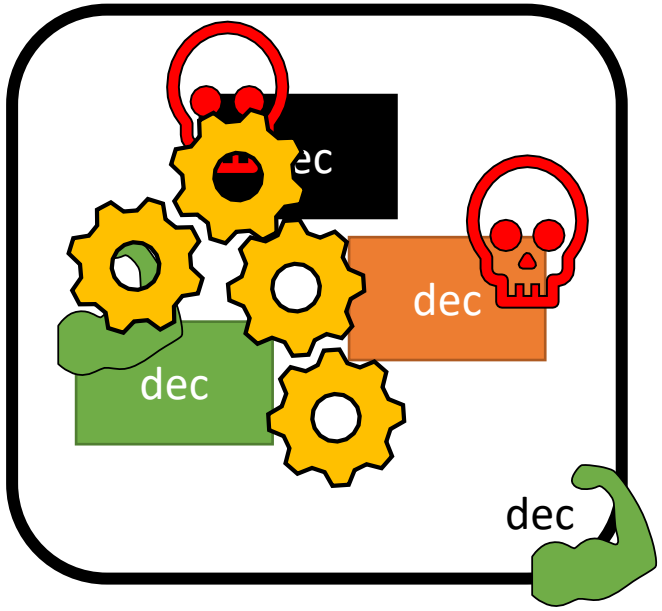
$$\mathrm{Adv} = \Pr[\mathcal{A} \to_\$ (c, ad, n) : \mathrm{dec}(k, c, ad, n) \neq \perp]$$

- GCM, OCB~~2~~,3, EAX, CWC, CAESAR Competition, …
- Important for all* modern communication protocols

Combiners

- What happens if your AEAD scheme turns out to be weak?
- Would be nice to minimize risk
- Combine multiple schemes (e.g., GCM with OCB3 and EtM)
  - Secure if one underlying scheme remains secure
- Would be nice, but how?





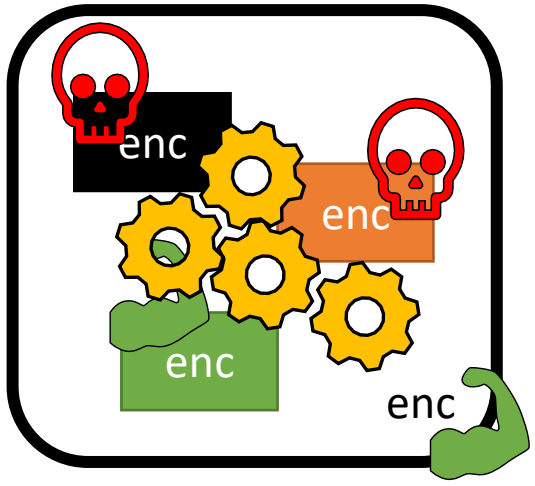
Combiners

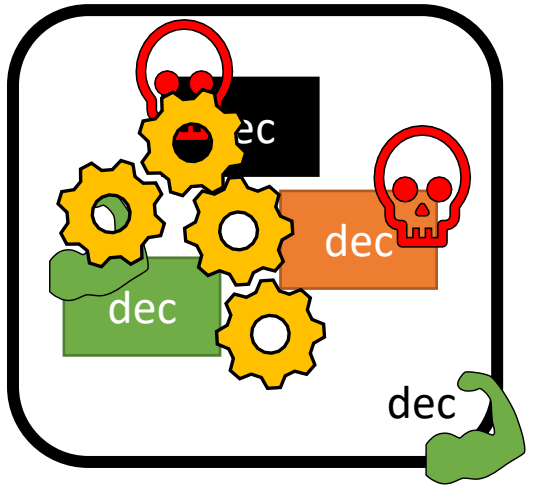
Further reasons:

- Political reasons (standards, monoculture, …)
- Untested schemes (“new” ideas in new schemes)
- Proof of GCM was wrong (still GCM was secure), EAX’ was broken
- …

Desired properties of combiners:

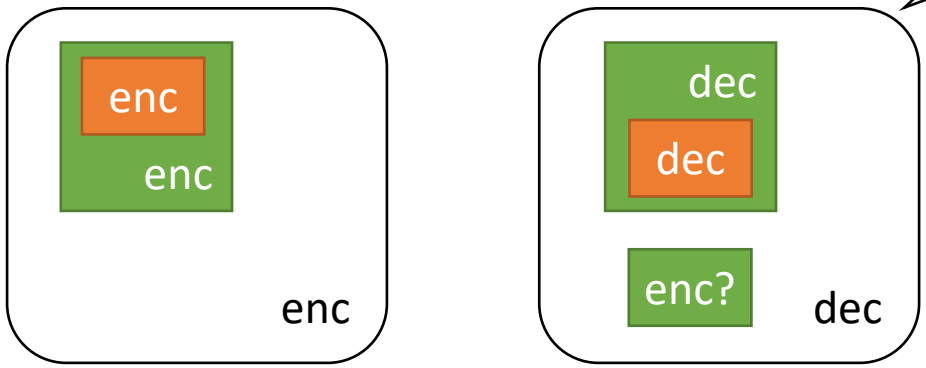
- Few assumptions on schemes
- Clear
- Easy proof
- No additional building blocks
- Performant
- …





Our AEAD Combiners

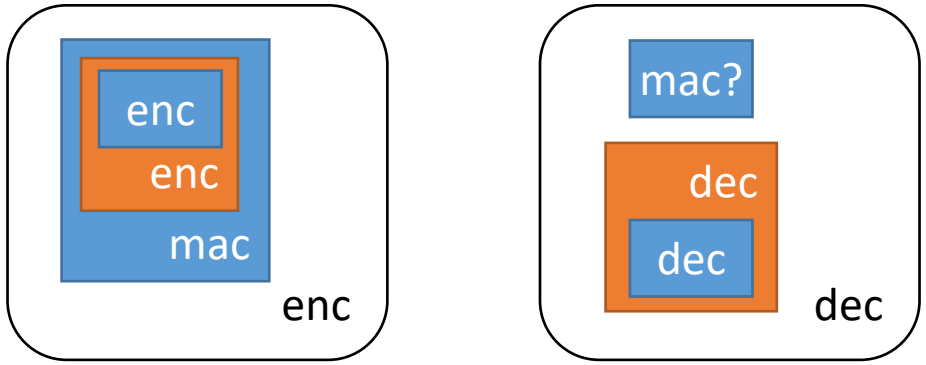
Blackbox



Super fast for blackbox combiner

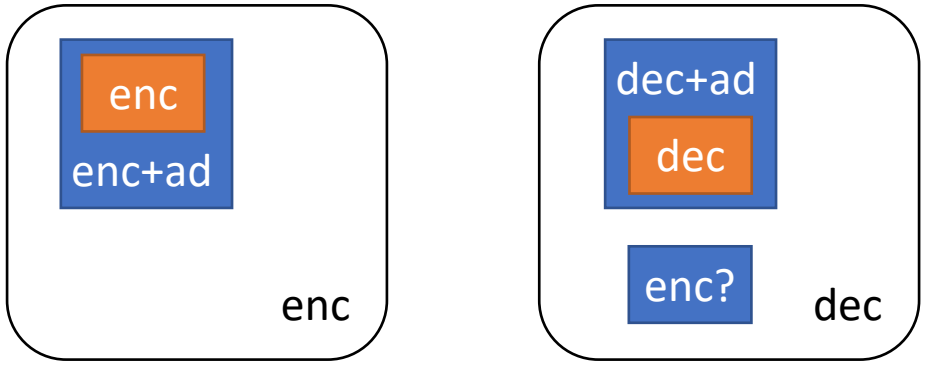
Even optimal*

Enc-then-MAC I

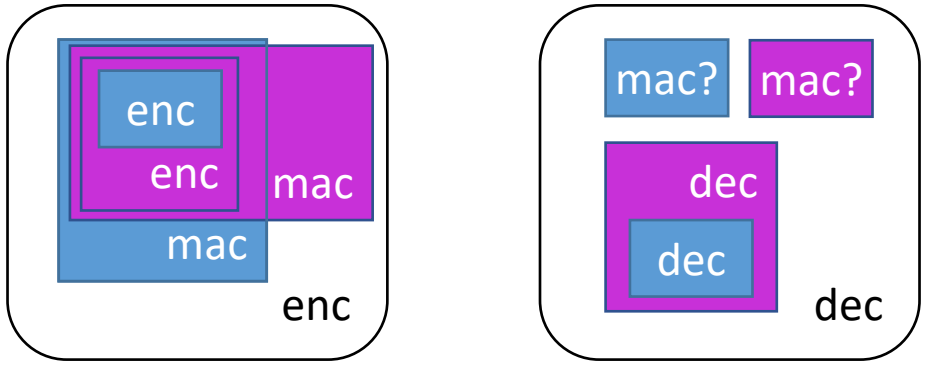


Special AEADs → Better performance

Ciphertext Translation

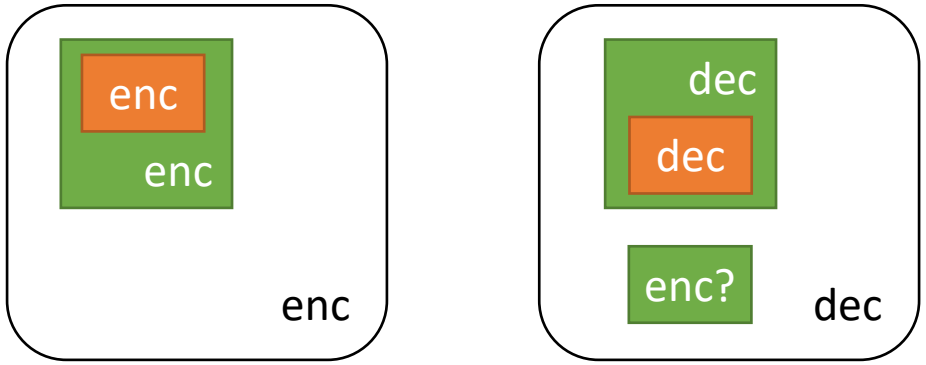


Enc-then-MAC II



Our AEAD Combiners: Design

Blackbox



Proc $\mathrm{enc}(k, n, ad, m)$

00 $(k_0, k_1) \leftarrow k$

01 $c_0 \leftarrow \mathrm{enc}_0(k_0, n, ad, m)$

02 $c_1 \leftarrow \mathrm{enc}_1(k_1, n, ad, c_0)$

03 Return c_1

Proc $\mathrm{dec}(k, n, ad, c)$

04 $(k_0, k_1) \leftarrow k$

05 $c_0 \leftarrow \mathrm{dec}_1(k_1, n, ad, c)$

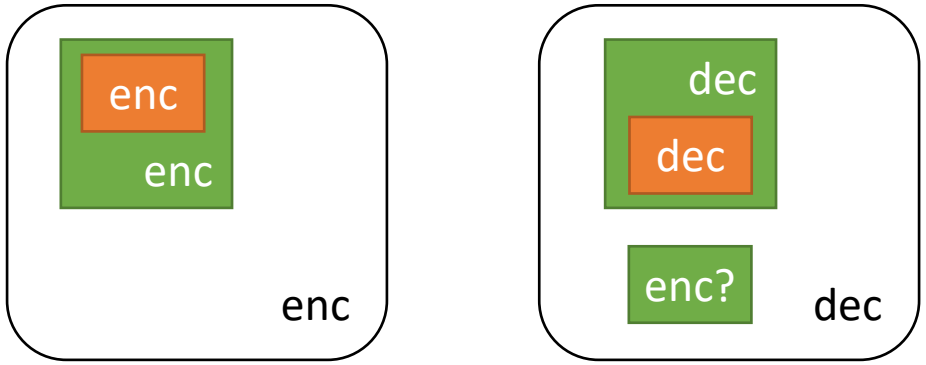
07 $m \leftarrow \mathrm{dec}_0(k_0, n, ad, c_0)$

11 Return m

- Blackbox:  generic syntax & security
- Idea:
  - Nest encryptions
  - Reverse at decryption
- Confidentiality: Nested encryption

Our AEAD Combiners: Design

Blackbox



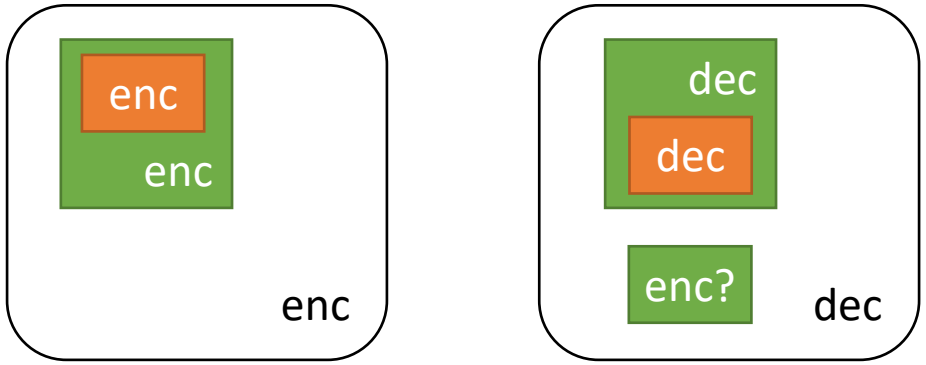
Proc $\mathrm{enc}(k, n, ad, m)$
00 $(k_0, k_1) \leftarrow k$
01 $c_0 \leftarrow \mathrm{enc}_0(k_0, n, ad, m)$
02 $c_1 \leftarrow \mathrm{enc}_1(k_1, n, ad, c_0)$
03 Return c_1

Proc $\mathrm{dec}(k, n, ad, c)$
04 $(k_0, k_1) \leftarrow k$
05 $c_0 \leftarrow \mathrm{dec}_1(k_1, n, ad, c)$
06 If $c_0 = \perp$: Return $\perp$
07 $m \leftarrow \mathrm{dec}_0(k_0, n, ad, c_0)$
11 Return m

- Blackbox:  generic syntax & security
- Idea:
  - Nest encryptions
  - Reverse at decryption
- Confidentiality: Nested encryption
- Integrity:
  - If outer integrity protection secure: check

Our AEAD Combiners: Design

Blackbox



Proc $\mathrm{enc}(k, n, ad, m)$
00 $(k_0, k_1) \leftarrow k$
01 $c_0 \leftarrow \mathrm{enc}_0(k_0, n, ad, m)$
02 $c_1 \leftarrow \mathrm{enc}_1(k_1, n, ad, c_0)$
03 Return c_1

Proc $\mathrm{dec}(k, n, ad, c)$
04 $(k_0, k_1) \leftarrow k$
05 $c_0 \leftarrow \mathrm{dec}_1(k_1, n, ad, c)$
06 If $c_0 = \perp$: Return $\perp$
07 $m \leftarrow \mathrm{dec}_0(k_0, n, ad, c_0)$
08 If $m = \perp$: Return $\perp$
09 $c_1' \leftarrow \mathrm{enc}_1(k_1, n, ad, c_0)$
10 If $c_1' \neq c$: Return $\perp$
11 Return m

- Blackbox:  generic syntax & security
- Idea:
  - Nest encryptions
  - Reverse at decryption
  - Recompute & verify outer ciphertext
- Confidentiality: Nested encryption
- Integrity:
  - If outer integrity protection secure: check
  - Else:
    - Inner ciphertext is secure
    - Outer ciphertext deterministic from it
      If same result as input: check

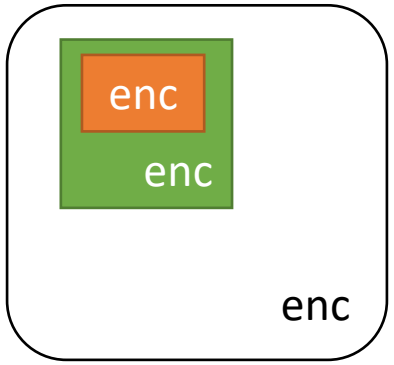
Generalizes to INT-PTXT → INT-CTXT

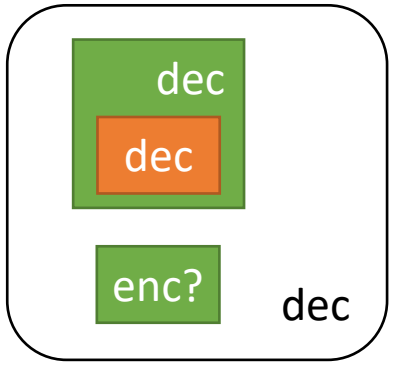
Our AEAD Combiners: Design

EtM AEADs

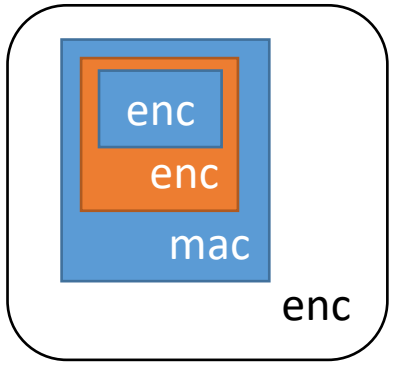
CT AEAD:
ad is processed separately and XORed

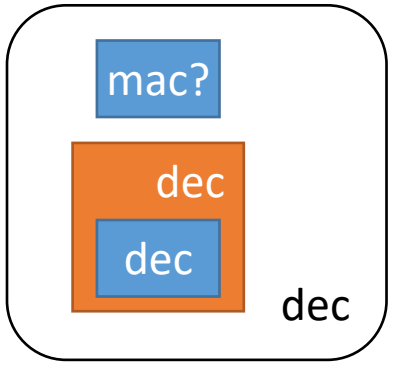
Blackbox





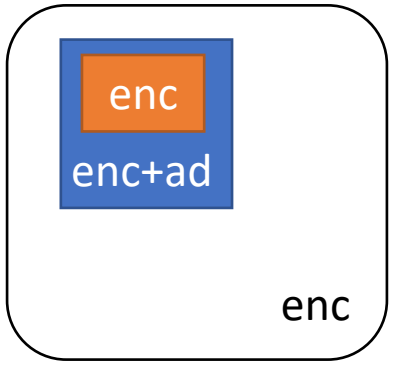
Enc-then-MAC I

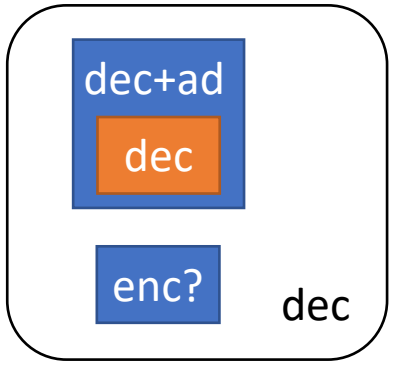




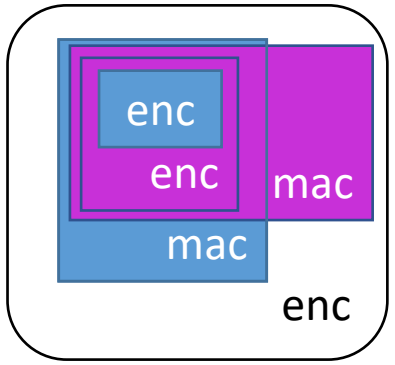
Nested encryption for confidentiality

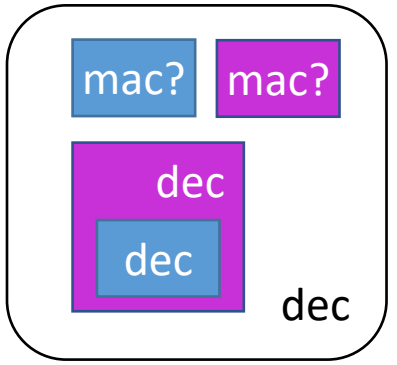
Ciphertext Translation





Enc-then-MAC II



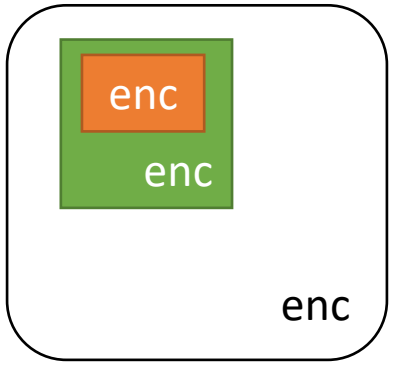


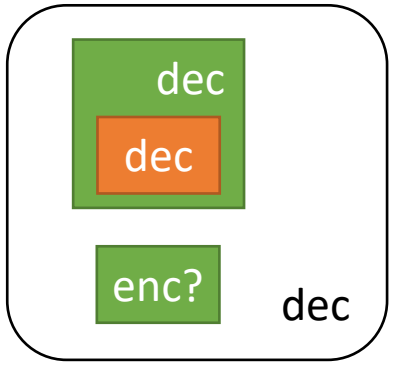
Our AEAD Combiners: Design

EtM AEADs

CT AEAD:
ad is processed separately and XORed

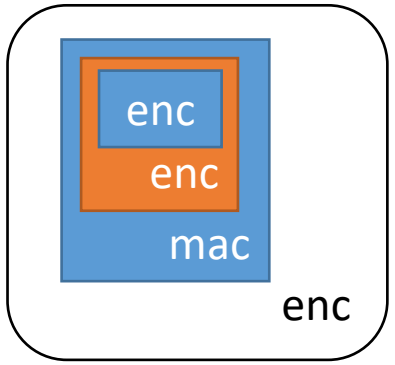
Blackbox

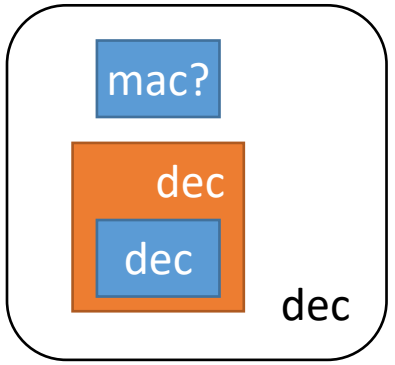




Processing:
enc: **2**·|m|+**2**·|ad|
dec: 3·|m|+3·|ad|

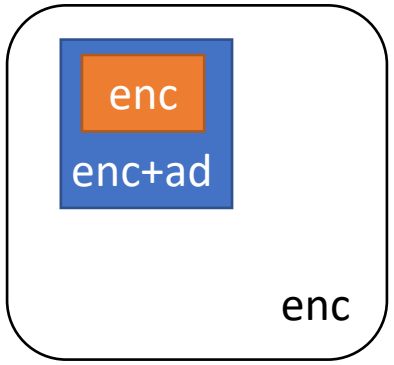
Enc-then-MAC I

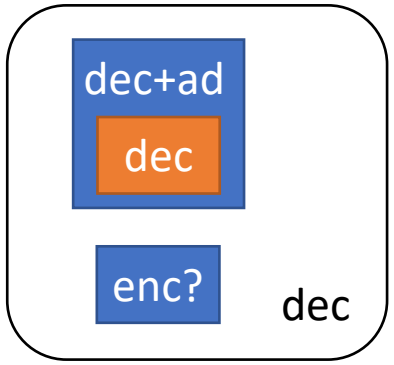




Re-computation for integrity

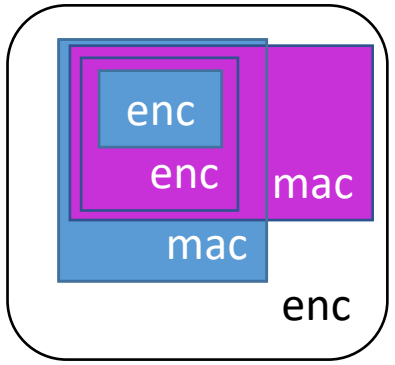
Ciphertext Translation

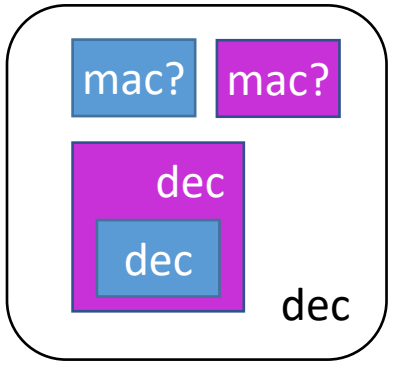




Processing:
enc: **2**·|m|+**2**·|ad|
dec: 3·|m|+**2**·|ad|

Enc-then-MAC II



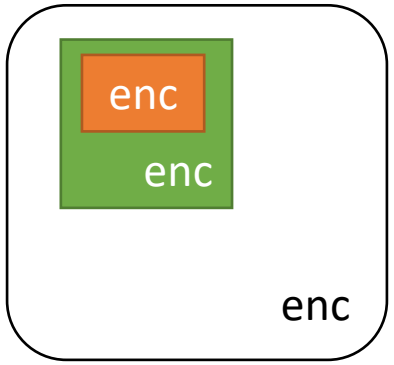


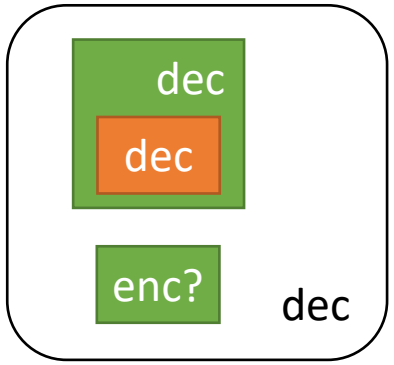
Our AEAD Combiners: Design

EtM AEADs

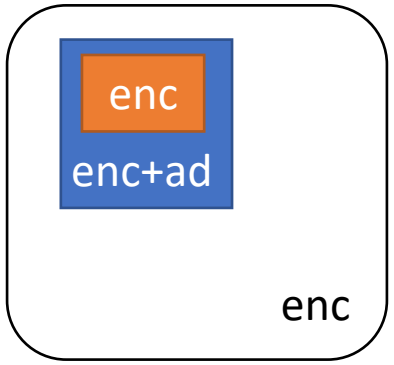
CT AEAD:
ad is processed separately and XORed

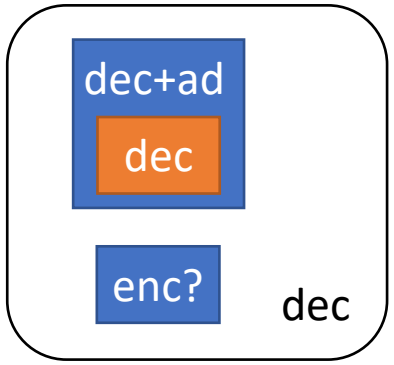
Blackbox





Ciphertext Translation



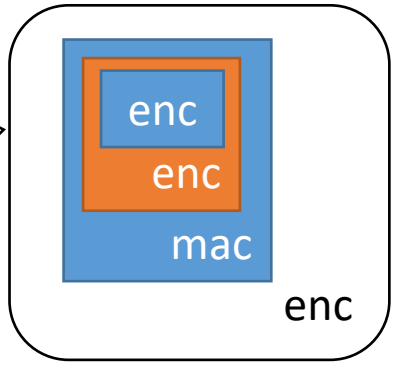


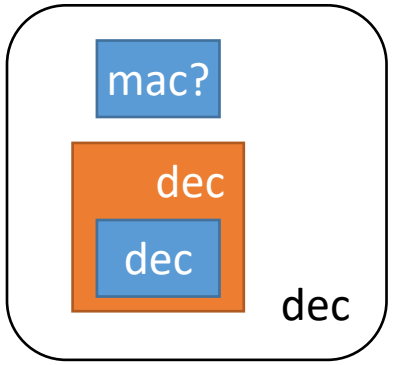
Processing:
enc: 3·|m|+**2**·|ad|
dec: 3·|m|+**2**·|ad|

MAC and outer enc or 2·MAC for integrity

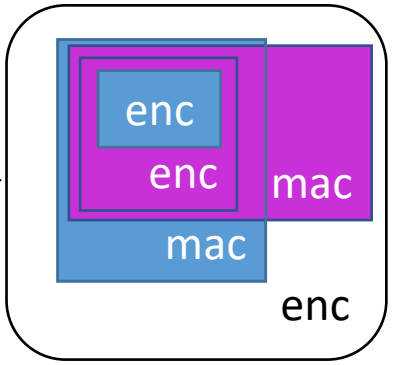
Transmission:
|c|=|m|+2·ε
+max(|tag|)

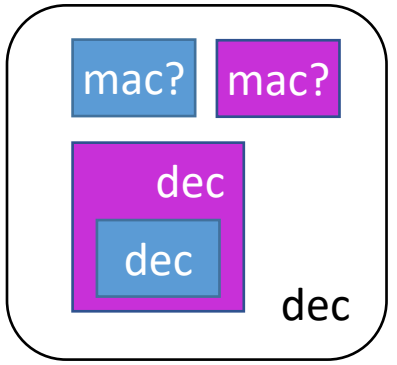
Enc-then-MAC I





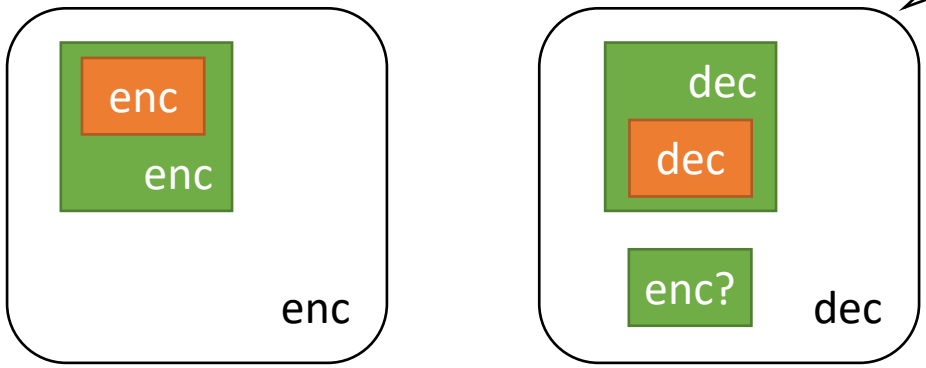
Enc-then-MAC II





Our AEAD Combiners: Performance

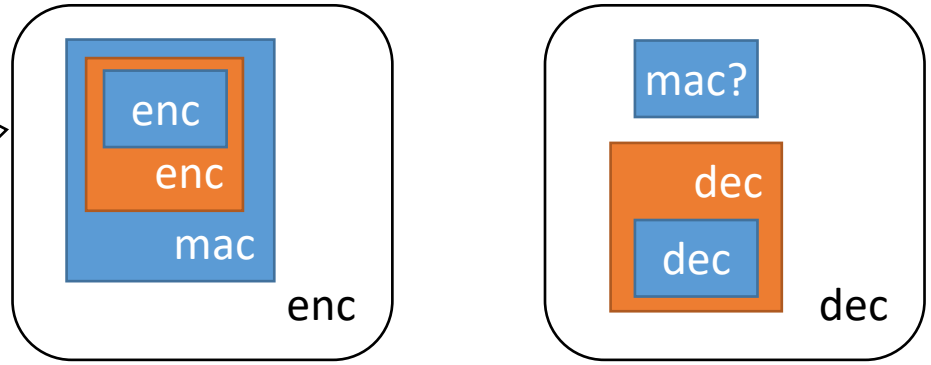
Blackbox



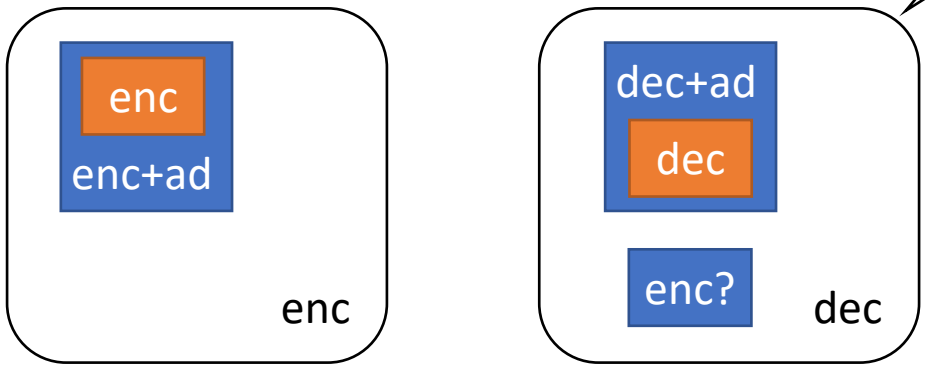
Processing:
enc: **2**·|m|+**2**·|ad|
dec: 3·|m|+3·|ad|

Enc-then-MAC I

Processing:
enc: 3·|m|+**2**·|ad|
dec: 3·|m|+**2**·|ad|



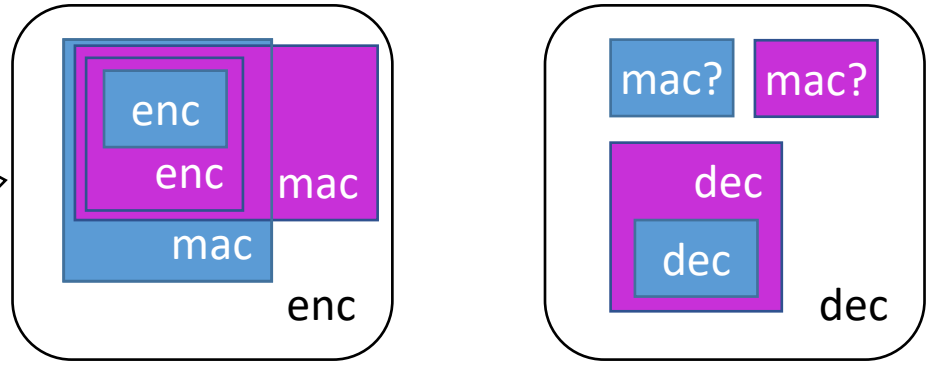
Ciphertext Translation



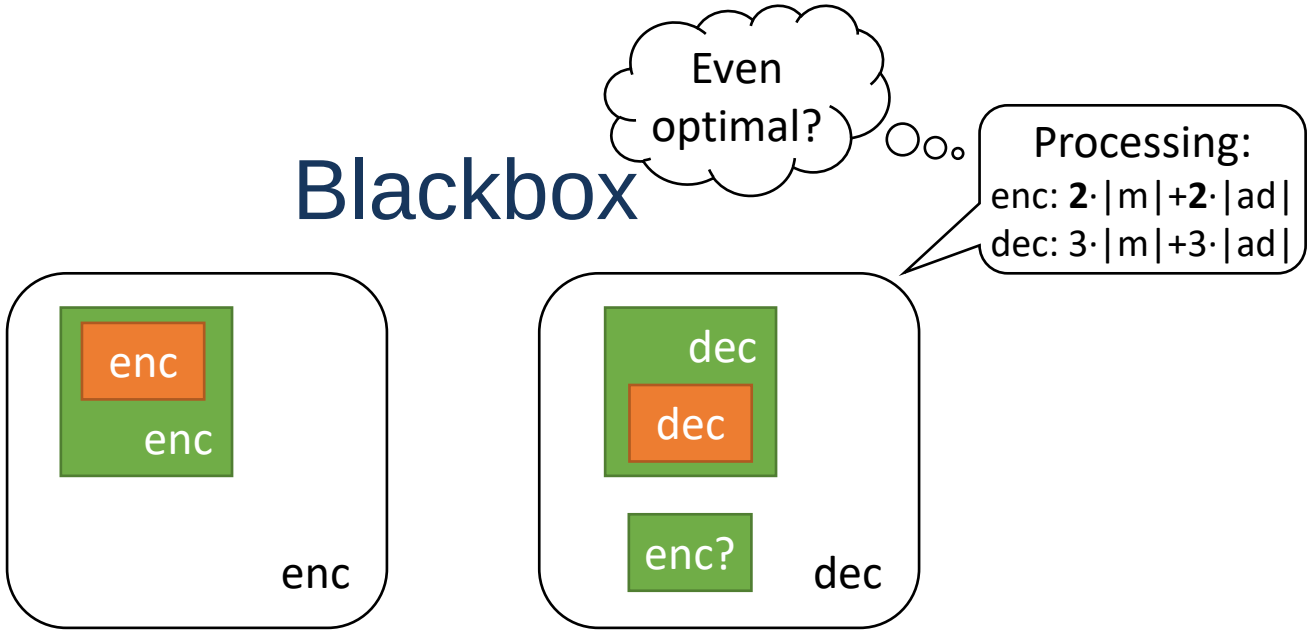
Processing:
enc: **2**·|m|+**2**·|ad|
dec: 3·|m|+**2**·|ad|

Enc-then-MAC II

Transmission:
|c|=|m|+2·ε
+max(|tag|)

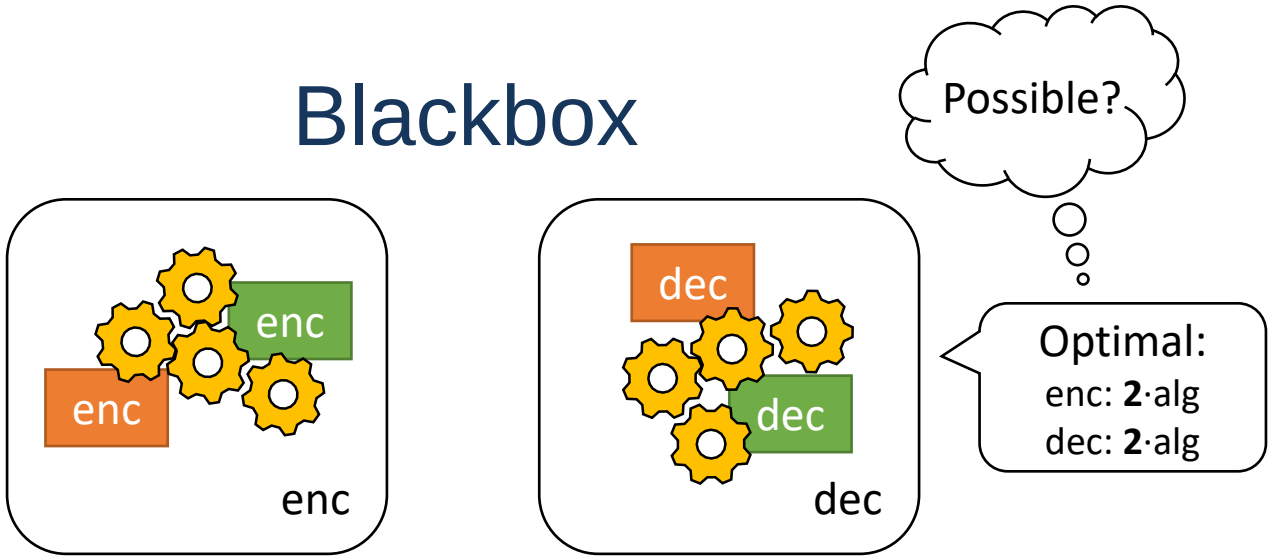


Our AEAD Combiners: Performance



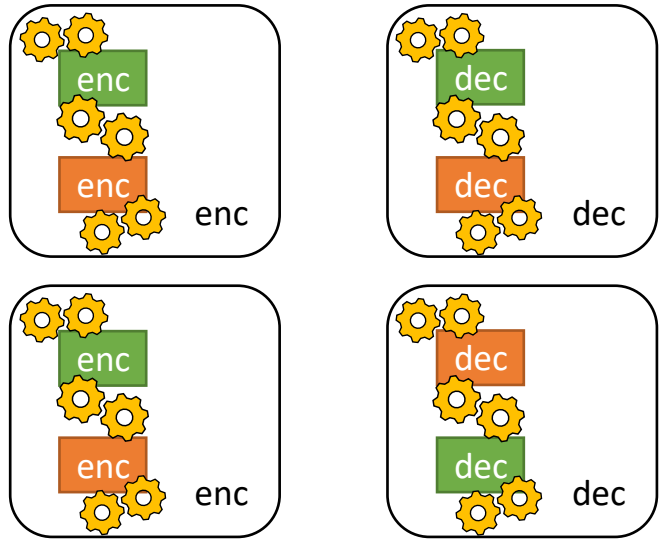
Our AEAD Combiners: Optimality

Blackbox



a) Synchronized

b) Reversed



→ Capture all (remaining) cases

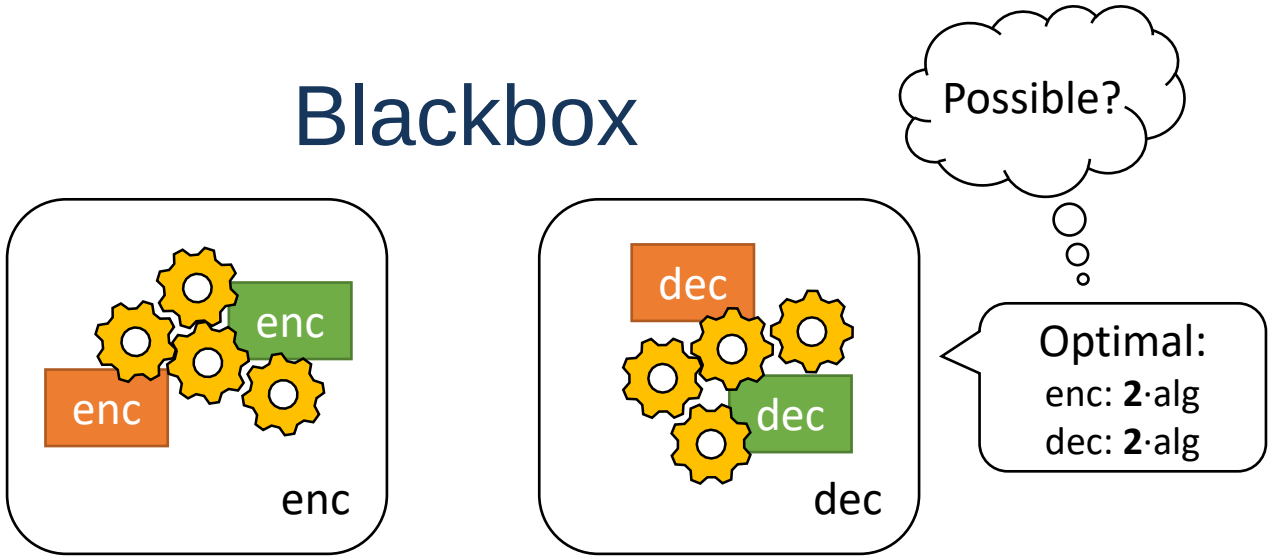
- We use four weak AEAD schemes enc dec :

a) Last ciphertext bit = 0
b) Last ciphertext bit = 1
c) enc = a), dec: tolerant
d) enc = b), dec: tolerant

- Combiner cannot distinguish (a,b) from (c,d)
- If combiner protects against c) or d), correctness for a) or b) is not reached

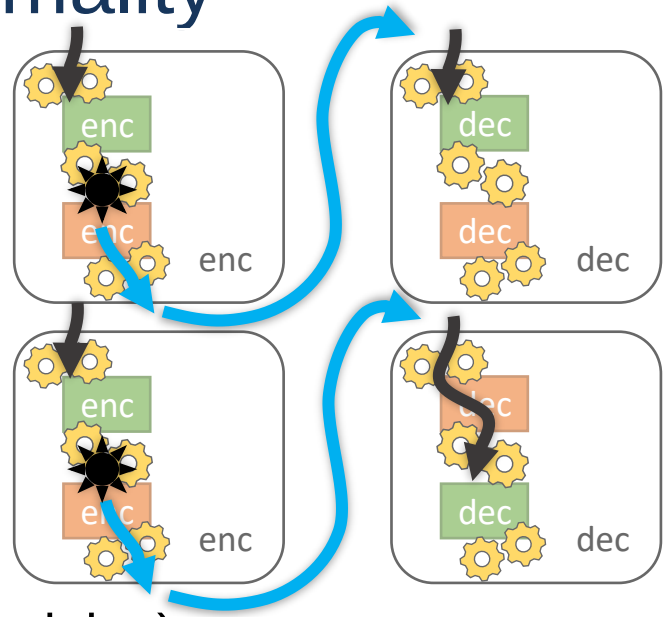
Our AEAD Combiners: Optimality

Blackbox



a) Synchronized

b) Reversed



→ Capture all (remaining) cases

- We use four weak AEAD schemes enc dec :

a) Last ciphertext bit = 0
b) Last ciphertext bit = 1
c) enc = a), dec: tolerant
d) enc = b), dec: tolerant

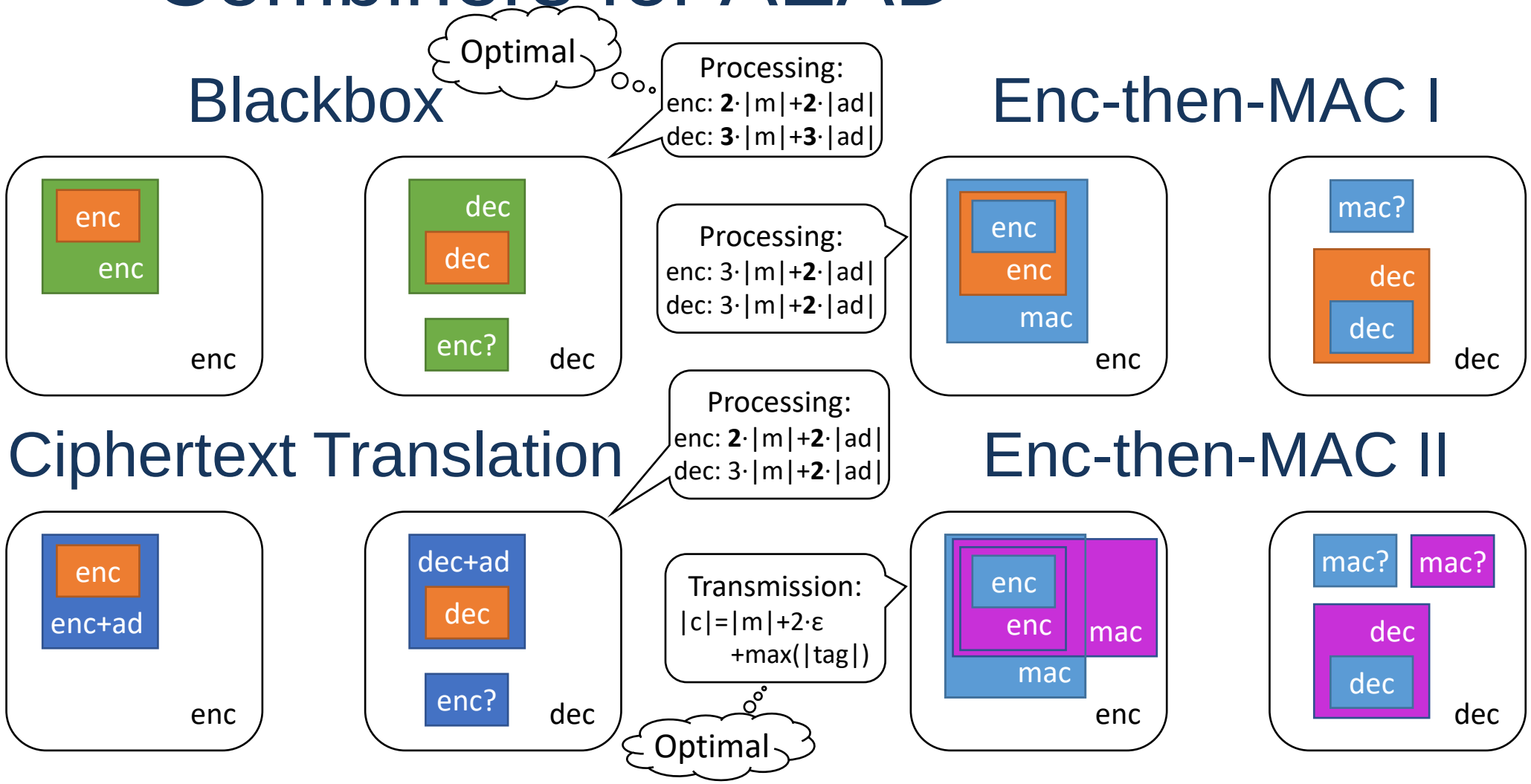
- Combiner cannot distinguish (a,b) from (c,d)
- If combiner protects against c) or d), correctness for a) or b) is not reached

Adversary generically breaks authenticity:

1. Trace combined encryption and decryption
2. Replace enc of case c) with case d) (or d) with c))
   → This is a ciphertext forgery
3. Looks like valid case b) (or a)) ciphertext to combiner
   → Combiner cannot prevent forgery (otherwise no correctness)

→ More than 2·enc and 2·dec necessary

Combiners for AEAD



ia.cr/2020/232 @roeslpa