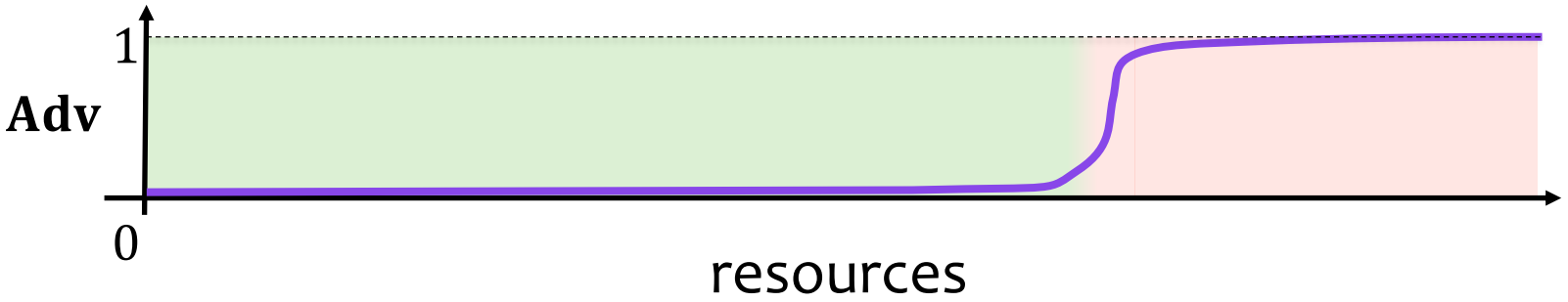
The Memory-Tightness of Authenticated Encryption

Ashrujit Ghoshal **Joseph Jaeger** **Stefano Tessaro**

University of Washington

CRYPTO 2020

Concrete security theorems: $\mathbf{Adv}(\text{resources}) \leq \epsilon$



Traditionally: time t, data complexity/queries q $\quad \mathbf{Adv}(t, q) \leq \epsilon$

This work: time t, data complexity/queries q, memory S $\quad \mathbf{Adv}(t, q, S) \leq \epsilon$

Prior work

Time-memory tradeoffs for symmetric encryption [TT18, JT19, Dinur20, SS20]

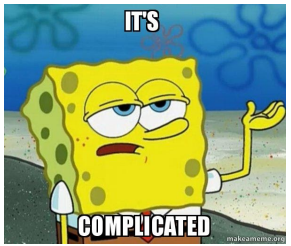
Focus: **confidentiality**

Memory-tight reductions [ACFK17, WMHT18, GT20, Bhattacharya20]

Focus: **public-key crypto**

This work: Time-memory tradeoffs for (nonce-based) **authenticated encryption (AE)**

Tl;dr:

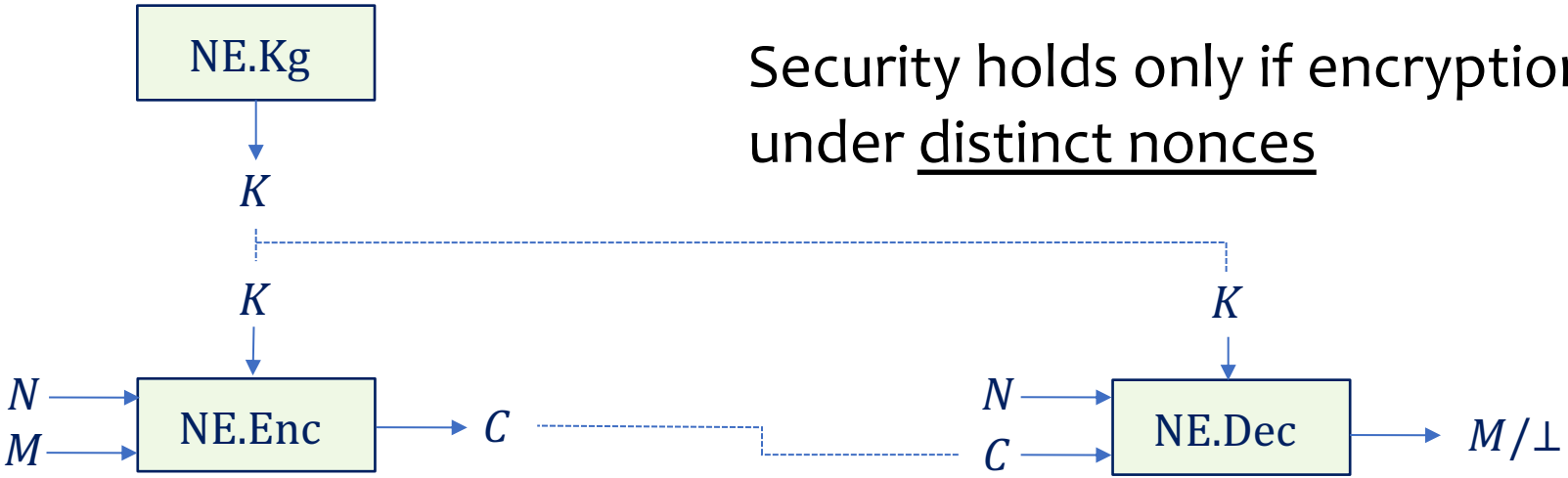


Positive results

Negative results

Nonce-based encryption

NE=(NE.Kg, NE.Enc, NE.Dec)



Security holds only if encryption under distinct nonces

Long line of work on concrete security of nonce-based AE

[BR00, RBBK01, R02, RS06 …]

Can we extend them to consider memory?

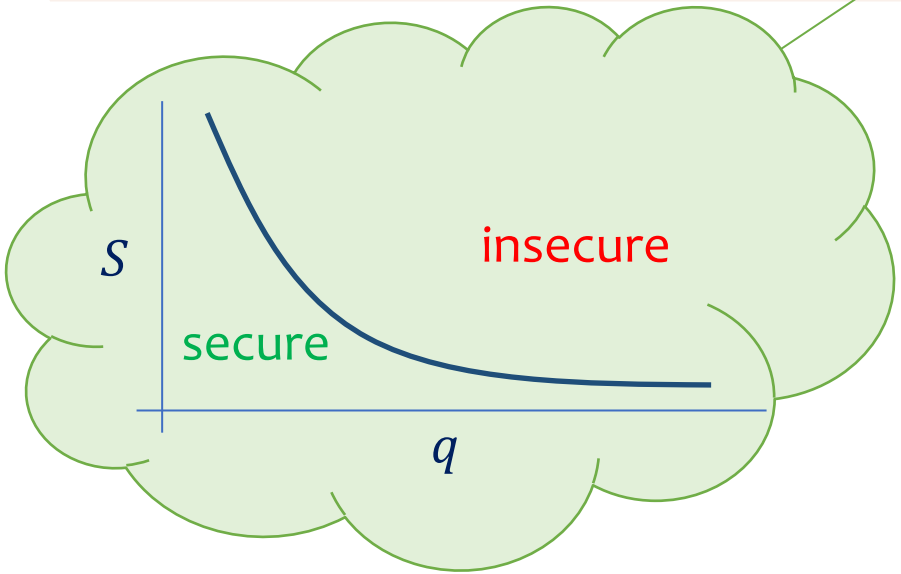
Example: $\mathrm{NE.Enc}(K, N, M) = \mathrm{E}_K(N) \oplus M$ $\quad$ $\mathrm{E} = n$-bit block cipher

indr = indistinguishability from random ciphertexts

Theorem. [JT19 + Dinur20]

$$\mathbf{Adv}^{\mathrm{indr}}_{\mathrm{NE}}(t, q, S) \leq \frac{S \cdot q \log q}{2^n} + \mathbf{Adv}^{\mathrm{prp}}_{\mathrm{E}}(t, q, S)$$

t = time, q = # encryptions, S = memory



e.g. beyond-birthday security for $S < 2^{\frac{n}{2}}$

Goal: similar results for **AE security**? (like **GCM** [MV04])

Target: combined AE security notion (confidentiality + integrity)

Usual proof approach: INDR + CTXT ⇒ AE

indistinguishability from random ciphertexts

ciphertext integrity

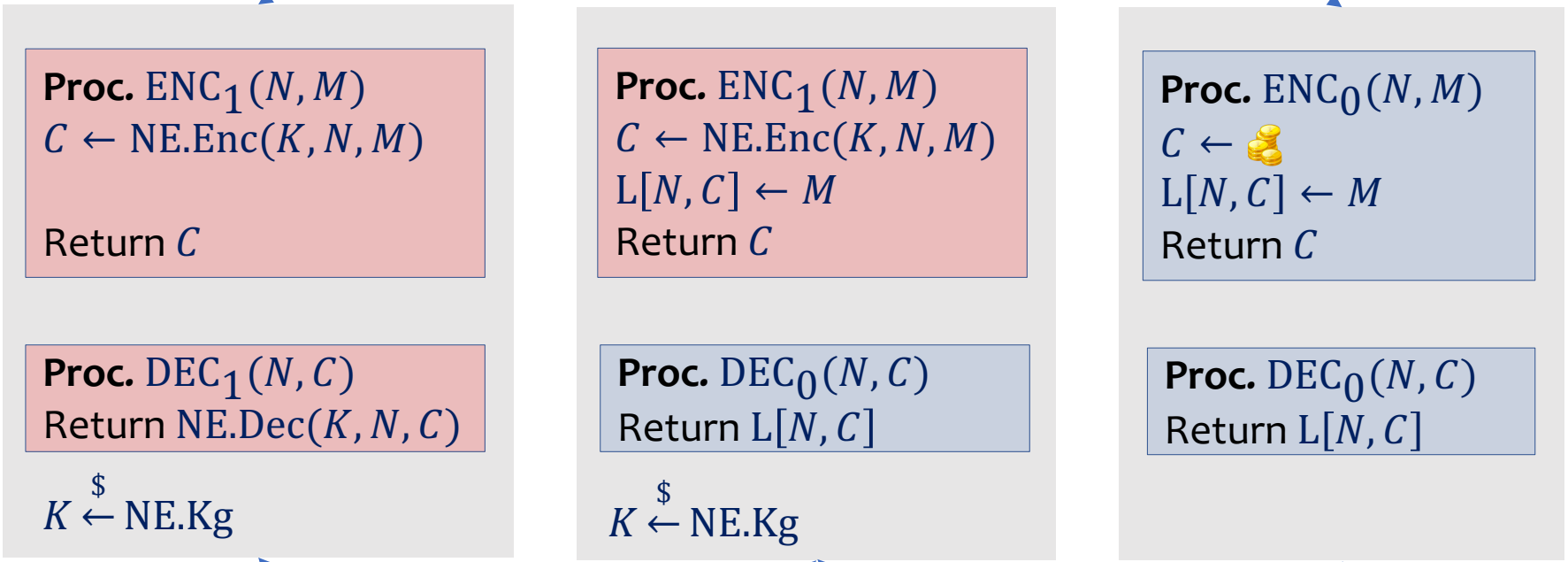
Theorem. $\mathbf{Adv}^{\mathbf{ae}}_{\mathbf{NE}}(t,q) \leq \mathbf{Adv}^{\mathbf{indr}}_{\mathbf{NE}}(t,q,S_1) + \mathbf{Adv}^{\mathbf{ctxt}}_{\mathbf{NE}}(t,q,S_2)$

Wanted: memory-tight reduction [ACFK17] $S_1 = S_2 = S$

Unclear! **Known reduction is not memory-tight!**

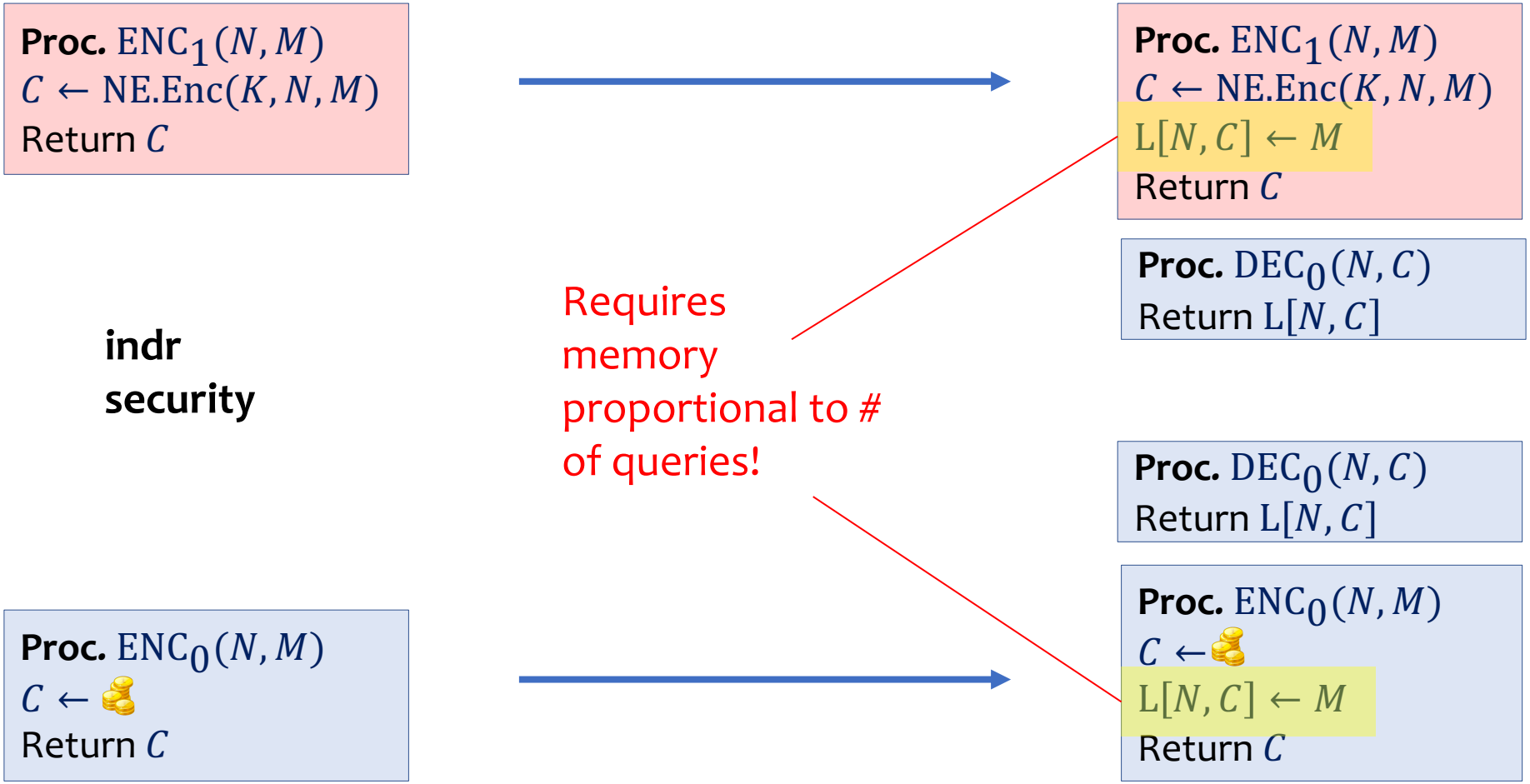
NE=(NE.Kg, NE.Enc, NE.Dec)

$\mathbf{Adv}^{\mathrm{ae}}_{\mathrm{NE}}(t, q, S)$?



$\mathbf{Adv}^{\mathrm{ctxt}}_{\mathrm{NE}}(t, q, S)$ 👍

$\mathbf{Adv}^{\mathrm{indr}}_{\mathrm{NE}}(t, q, S + O(q))$ 👎


Proc. $\mathrm{ENC}_1(N, M)$
$C \leftarrow \mathrm{NE.Enc}(K, N, M)$
Return C
Proc. $\mathrm{ENC}_1(N, M)$
$C \leftarrow \mathrm{NE.Enc}(K, N, M)$
$\mathrm{L}[N, C] \leftarrow M$
Return C
Proc. $\mathrm{DEC}_0(N, C)$
Return $\mathrm{L}[N, C]$
indr
security
Requires
memory
proportional to #
of queries!
Proc. $\mathrm{DEC}_0(N, C)$
Return $\mathrm{L}[N, C]$
Proc. $\mathrm{ENC}_0(N, M)$
$C \leftarrow$
Return C
Proc. $\mathrm{ENC}_0(N, M)$
$C \leftarrow$
$\mathrm{L}[N, C] \leftarrow M$
Return C

Our results, in a nutshell

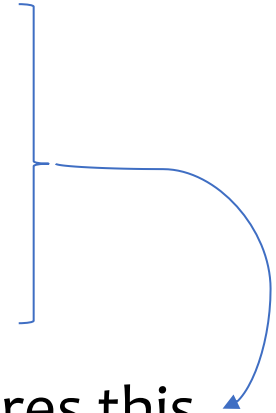
1. **Memory-tight reduction** and **time-memory trade-offs** in the channel setting
   - Typical usage within protocols like **TLS**
   - New technique: **memory-adaptive** reduction

2. **Impossibility result** for general memory-tight reduction INDR + CTXT ⇒ AE!

Channel setting: motivation

AE often used to establish a secure communication channel, as in **TLS**

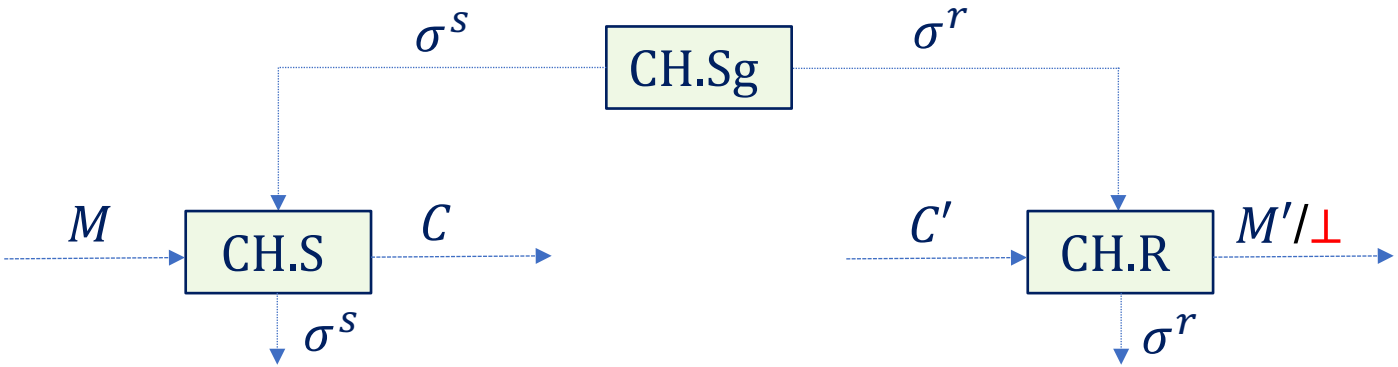
- **implicit** nonces = counter
  $\mathrm{ENC}(K, 0, M_0), \mathrm{ENC}(K, 1, M_1), \cdots$
- receiver **aborts** upon the first decryption failure
- **in-order** delivery



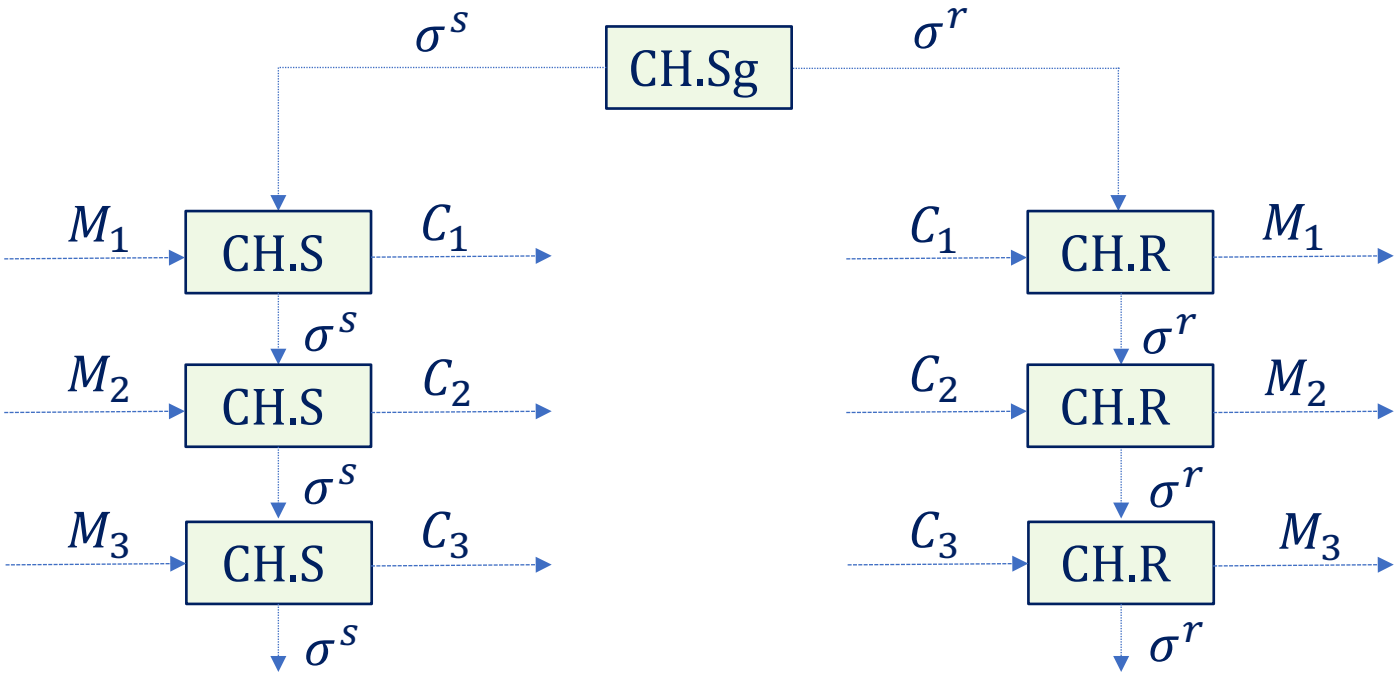
Channel setting captures this

The channel setting

CH=(CH.Sg, CH.S, CH.R)

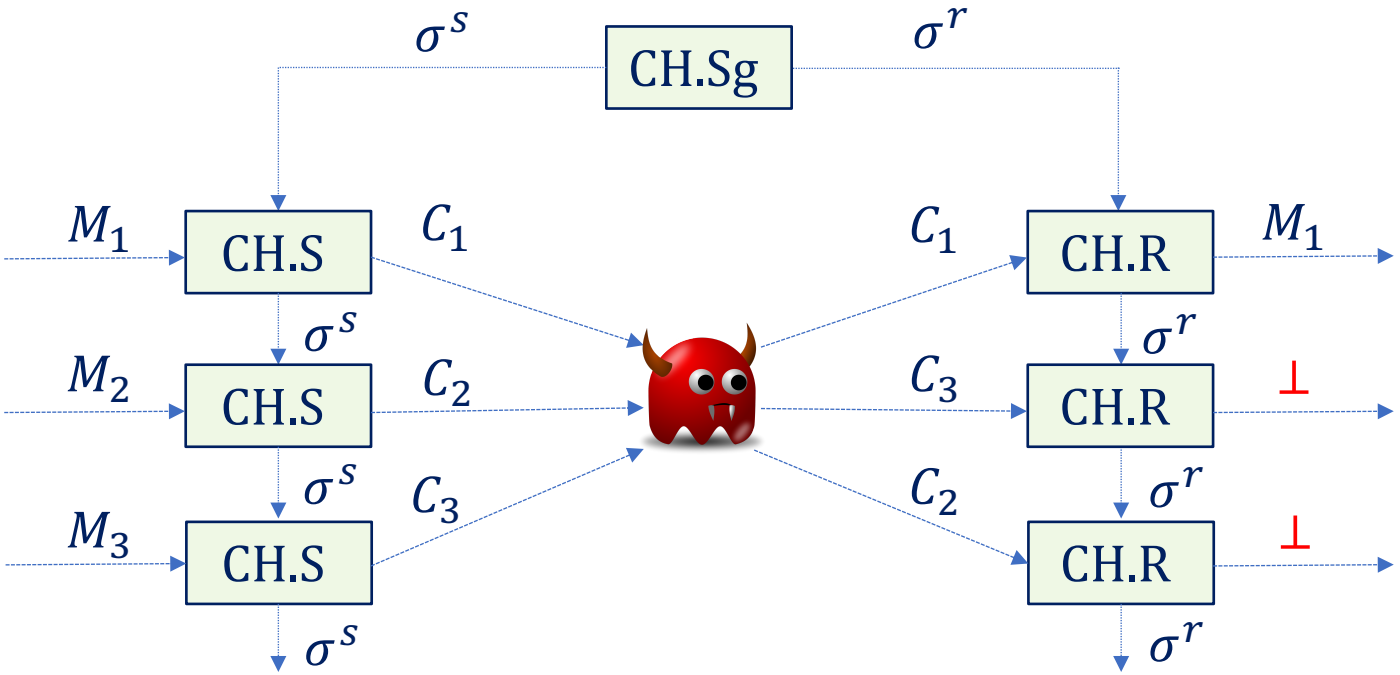


The channel setting: correctness CH=(CH.Sg, CH.S, CH.R)



The channel setting: security

CH=(CH.Sg, CH.S, CH.R)



AE security for channels

CH=(CH.Sg, CH.S, CH.R)

Proc. $\mathrm{ENC}_1(M)$
$(\sigma^s, C) \leftarrow \mathrm{CH.S}(\sigma^s, M)$
Return C

Proc. $\mathrm{DEC}_1(C)$
$(\sigma^r, M) \leftarrow \mathrm{CH.R}(\sigma^r, C)$
Return M

$(\sigma^s, \sigma^r) \xleftarrow{\$} \mathrm{CH.Sg}$

$\mathbf{Adv}^{\mathrm{ch-ae}}_{\mathrm{NE}}(t, q, S)$

Proc. $\mathrm{ENC}_0(M)$
$C \leftarrow$ 
$\mathrm{Enqueue}(M, C)$
Return C

Proc. $\mathrm{DEC}_0(C)$
$(M', C') \leftarrow \mathrm{Dequeue}()$
If sync then
  If $C = C'$ then return M'
  sync $\leftarrow$ false
Return $\perp$

sync $\leftarrow$ true

Main theorem

ae security for channels

ciphertext integrity for channels

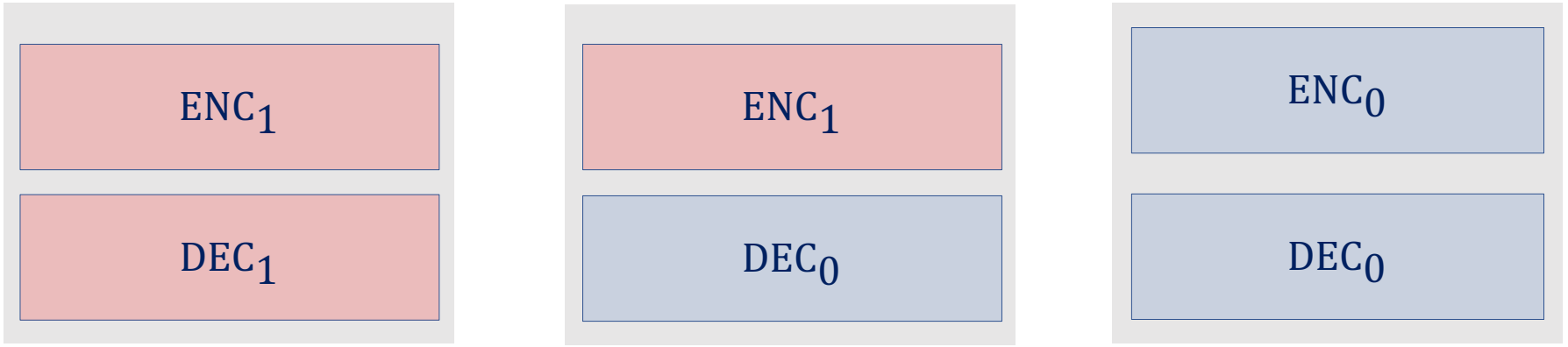
indistinguishability from random ciphertexts for channels

Theorem. [this work] $\forall\, \lambda \in \mathbb{N}$

$$\mathbf{Adv}^{\text{ch–ae}}_{\text{CH}}(t, q, S) \leq \mathbf{Adv}^{\text{ch–ctxt}}_{\text{CH}}(t, q, S) + 2 \cdot \mathbf{Adv}^{\text{ch–indr}}_{\text{CH}}(t, q, 3S + O(\log q) + \lambda) + \frac{1}{2^{\lambda}}$$

Memory-tight!

New technique: **Memory-adaptive reduction**



$$\mathbf{Adv}_{\mathrm{CH}}^{\mathrm{ch-ctxt}}(t, q, S)$$

$$2 \cdot \mathbf{Adv}_{\mathrm{CH}}^{\mathrm{ch-indr}}(t, q, 3S + O(\log q) + \lambda) + \frac{1}{2^{\lambda}}$$

Issue: size of queue grows with the number of queries

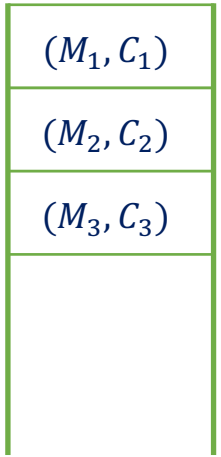
$\mathrm{ENC}_b(M_1), \mathrm{ENC}_b(M_2), \mathrm{ENC}_b(M_3)$

C_1, C_2, C_3

$\mathrm{DEC}_0(C_1)$

M_1

$b \xleftarrow{\$} \{0,1\}$



Queue

Key idea: bounding queue size does not change behavior

Example: only store ≤ 2 pairs

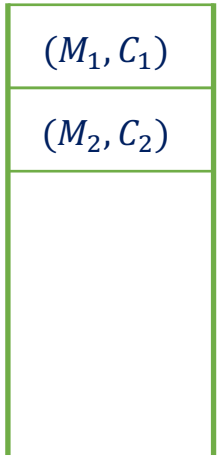
$$b \xleftarrow{\$} \{0,1\}$$

$$\mathrm{ENC}_b(M_1), \mathrm{ENC}_b(M_2), \mathrm{ENC}_b(M_3) \longrightarrow$$

$$\longleftarrow C_1, C_2, C_3$$

$$\mathrm{DEC}_0(C_1), \mathrm{DEC}_0(C_2), \mathrm{DEC}_0(C_3) \longrightarrow$$

$$\longleftarrow M_1, M_2, \perp$$



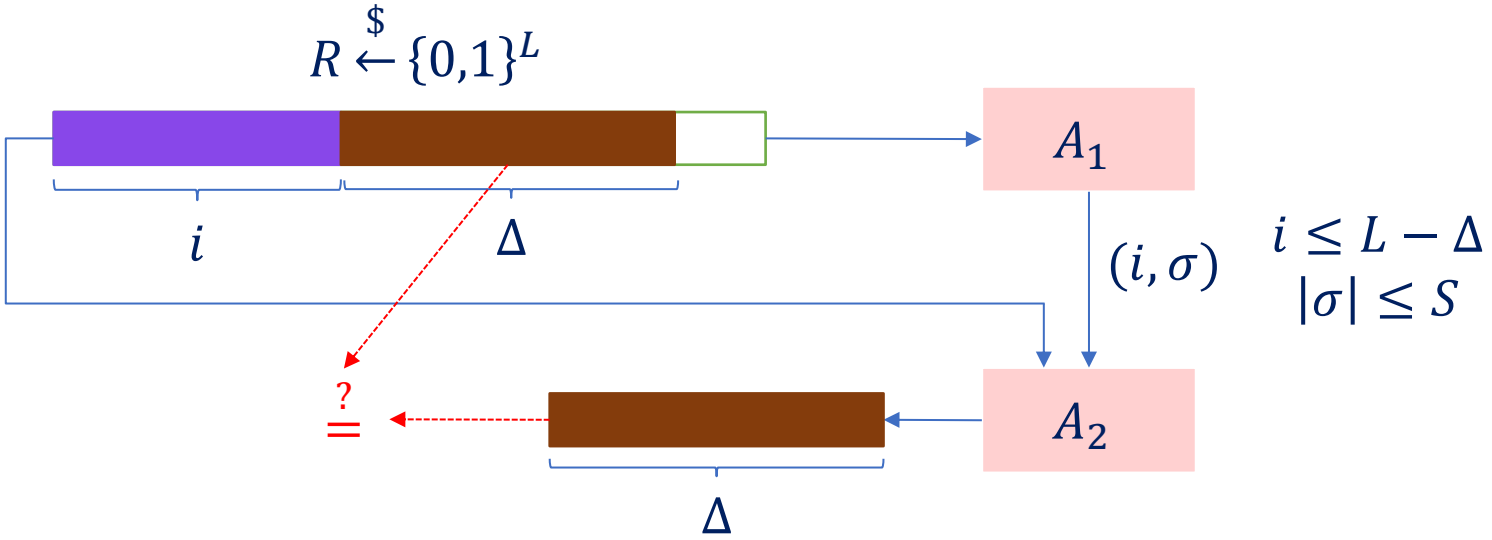
Queue

Adversary had to remember C_1, C_2, C_3 to cause this!

Bound queue size to $\Delta = 2S + \log q + \lambda$ bits

Information-theoretic game

$L, \Delta \in \mathbb{N}, \Delta \leq L$



Lemma. If $\Delta = 2S + O(\log L) + \lambda$ then

$$\Pr[(A_1, A_2) \text{ wins}] \leq \frac{1}{2^\lambda}$$

Application to GCM

one of the most widely deployed encryption schemes

CAU [BT16] : an abstraction of GCM

encryption scheme from block cipher E and hash function H

n-bit block cipher

AXU

Theorem. [this work]

$$\mathbf{Adv}^{\mathrm{ch-ae}}_{\mathrm{NCH}}(t,q,S) \le 4 \cdot \mathbf{Adv}^{\mathrm{prp}}_{\mathrm{E}}(t, O(q), O(S)) + O\left(\frac{Sq \log q}{2^n}\right)$$

channel induced by CAU

Our results, in a nutshell

1. **Memory-tight reduction** and **time-memory trade-offs** in the channel setting
   - Typical usage within protocols like **TLS**
   - New technique: **memory-adaptive** reduction

2. **Impossibility result** for general memory-tight reduction INDR + CTXT ⇒ AE!

Negative result for the general setting

- **Impossibility result** for proving INDR+CTXT⇒AE in a memory-tight way for nonce-based encryption schemes
  - Similar spirit as prior work [ACFK17,WMHT18,GT20]
- Also rules out **memory-adaptive** reductions (like the one for channels)
- Evidence that some restriction **necessary** for memory-tight reduction

Our result

inefficient

Theorem. [this work] ∀ INDR+CTXT-secure NE ∃ AE adversary A^* making q queries, using memory $O(\log q)$ s.t.

1) $\mathbf{Adv}^{\mathrm{ae}}_{\mathrm{NE}}(A^*) \approx 1$

2) ∀ "efficient" black-box reductions R using additional memory $S = o(q)$ then

$$\mathbf{Adv}^{\mathrm{indr}}_{\mathrm{NE}}(R[A^*]) = \mathrm{negl}$$

3) ∀ "efficient" black-box reductions R'

$$\mathbf{Adv}^{\mathrm{ctxt}}_{\mathrm{NE}}(R'[A^*]) = \mathrm{negl}$$

Our result

Theorem. [this work] ∀ INDR+CTXT-secure NE ∃ AE adversary A^* making q queries, using memory $O(\log q)$ s.t.

1) $\mathbf{Adv}^{\mathrm{ae}}_{\mathrm{NE}}(A^*) \approx 1$

2) ∀ "efficient" **restricted** black-box reductions R using additional memory $S = o(q)$ then

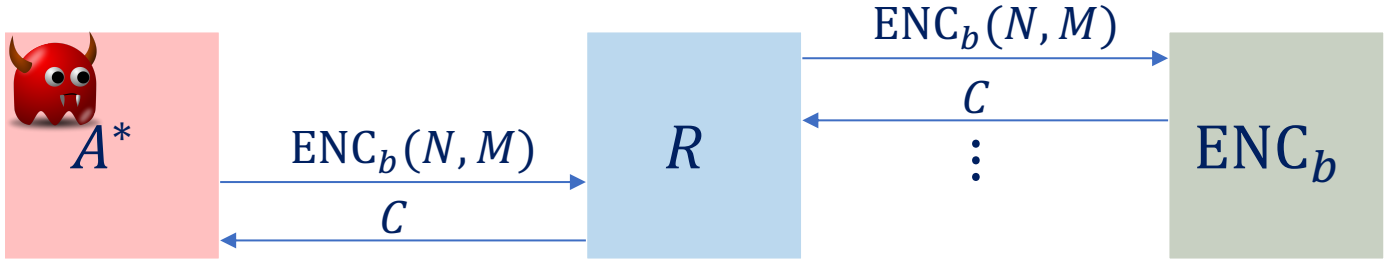
$$\mathbf{Adv}^{\mathrm{indr}}_{\mathrm{NE}}(R[A^*]) = \mathrm{negl}$$

3) ∀ "efficient" **restricted** black-box reductions R'

$$\mathbf{Adv}^{\mathrm{ctxt}}_{\mathrm{NE}}(R'[A^*]) = \mathrm{negl}$$

Restricted black-box reduction

1. faithful



2. nonce-respecting A^* ⇒ nonce-respecting R

3. straightline or fully-rewinding

The adversary A^*: basic idea

- In round $i = 1, \cdots, r$
  - Encrypt random $M_1, M_2, \cdots, M_u \xleftarrow{\$} \{0,1\}^\ell$

    $$C_j \leftarrow \mathrm{ENC}_b((i,j), M_j)$$
  - Sample $j^* \xleftarrow{\$} [u]$

    $$M \leftarrow \mathrm{DEC}_b((i,j^*), C_{j^*})$$
  - If $M_{j^*} \neq M$ then ABORT
- All rounds succeed $\Longrightarrow$ Inefficiently break the scheme

Intuition: reduction w/ memory $k \cdot \ell$ bits succeeds in each round w/ probability $\leq \frac{k}{u}$

Conclusions

- **Memory-sensitive bounds** for the AE security of channels

  Time-memory tradeoffs for the AE security of a TLS like channel instantiated with GCM

- New technique: **Memory-adaptive** reductions
- **Impossibility** for full AE security

  Evidence that restricting AE security to specific settings is inherent for memory-tight reductions

Open problems

- Memory-sensitive bounds for other practical examples of channels?
- More applications of memory-adaptive reductions?

Paper: https://eprint.iacr.org/2020/785