Lecture 5 – Authenticated encryption

TEK4500

21.09.2022

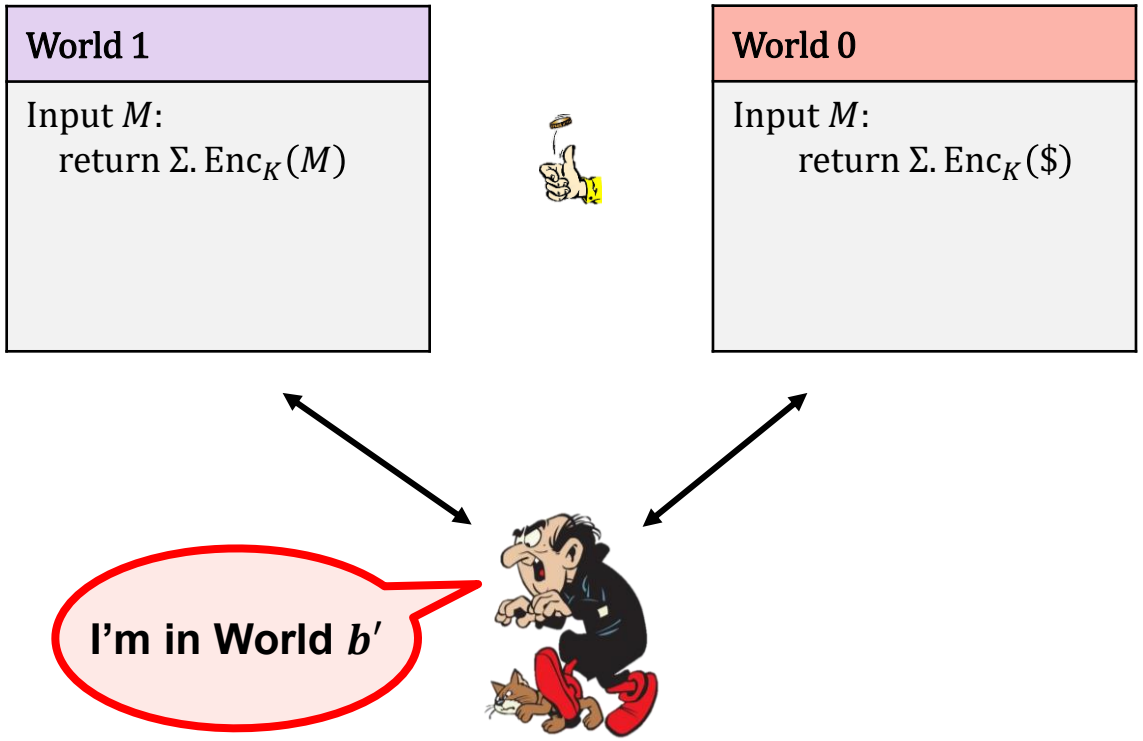
Håkon Jacobsen

hakon.jacobsen@its.uio.no

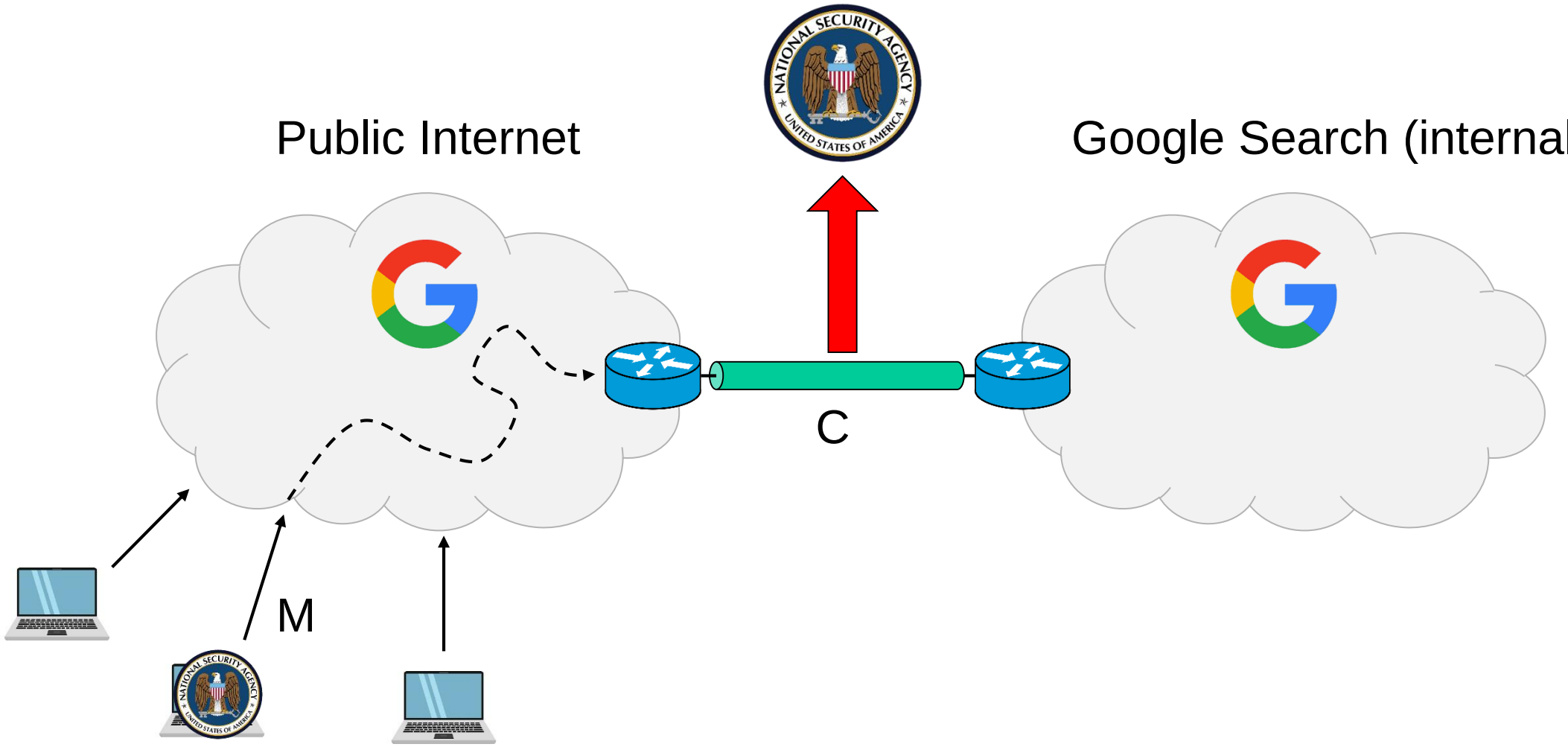
Basic goals of cryptography

	Message privacy	**Message integrity / authentication**
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures

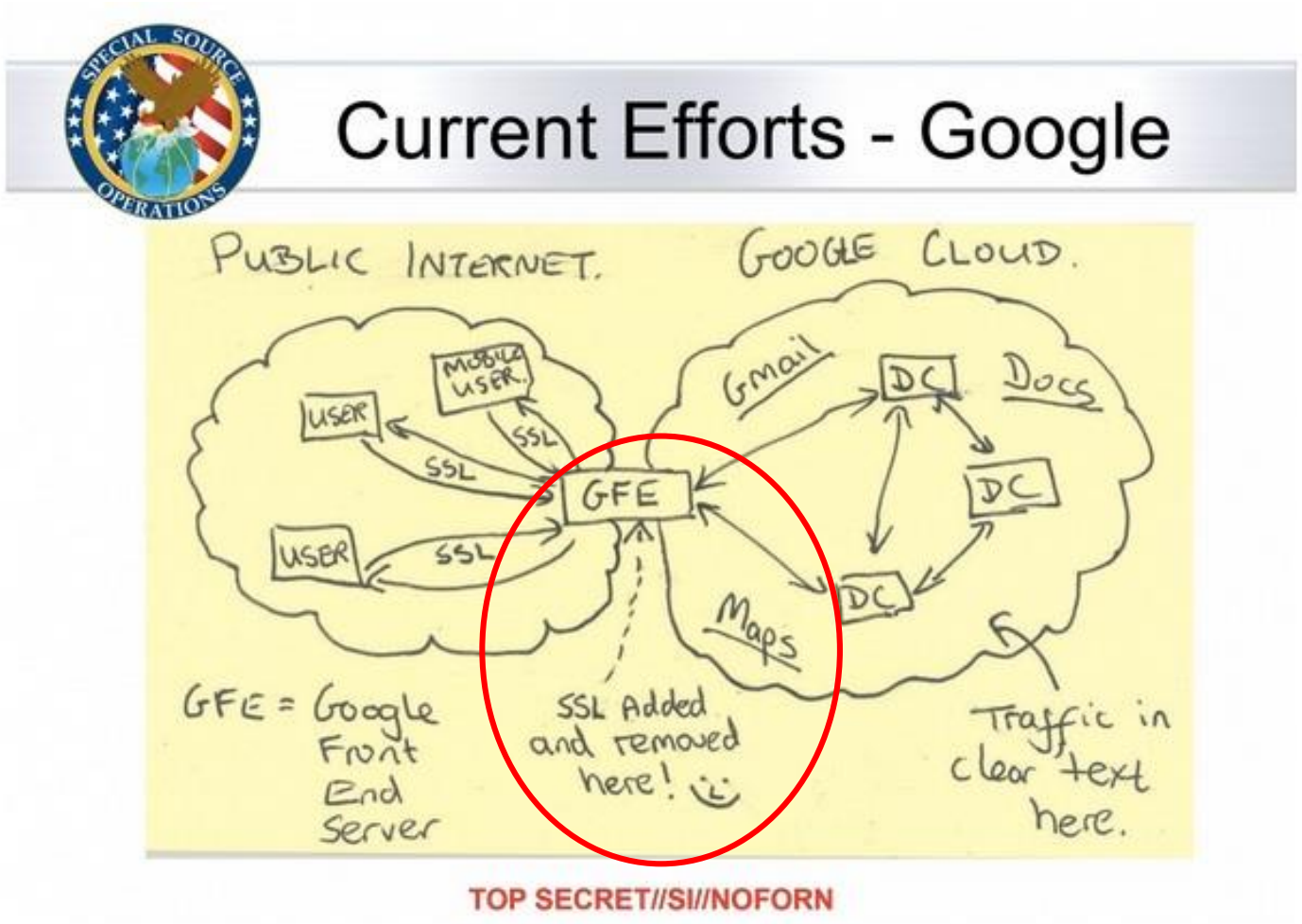
IND-CPA – Indistinguishability against chosen-plaintext attacks



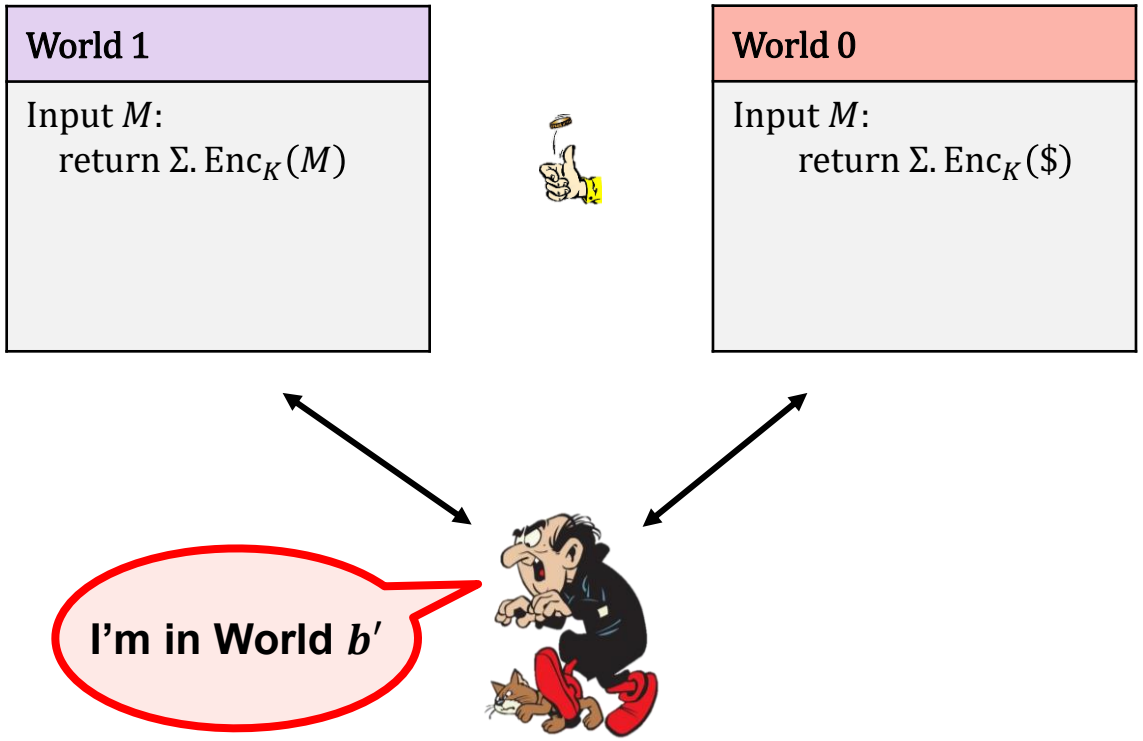
Chosen-plaintext attacks



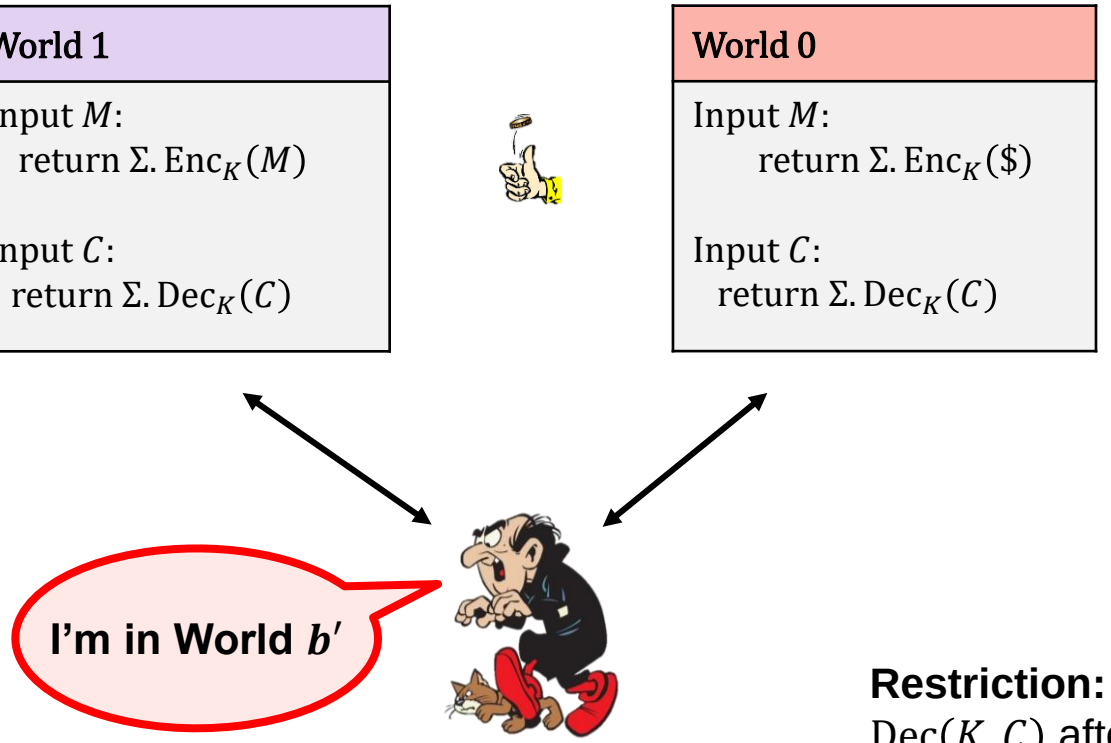
Chosen-plaintext attacks



IND-CPA – Indistinguishability against chosen-plaintext attacks



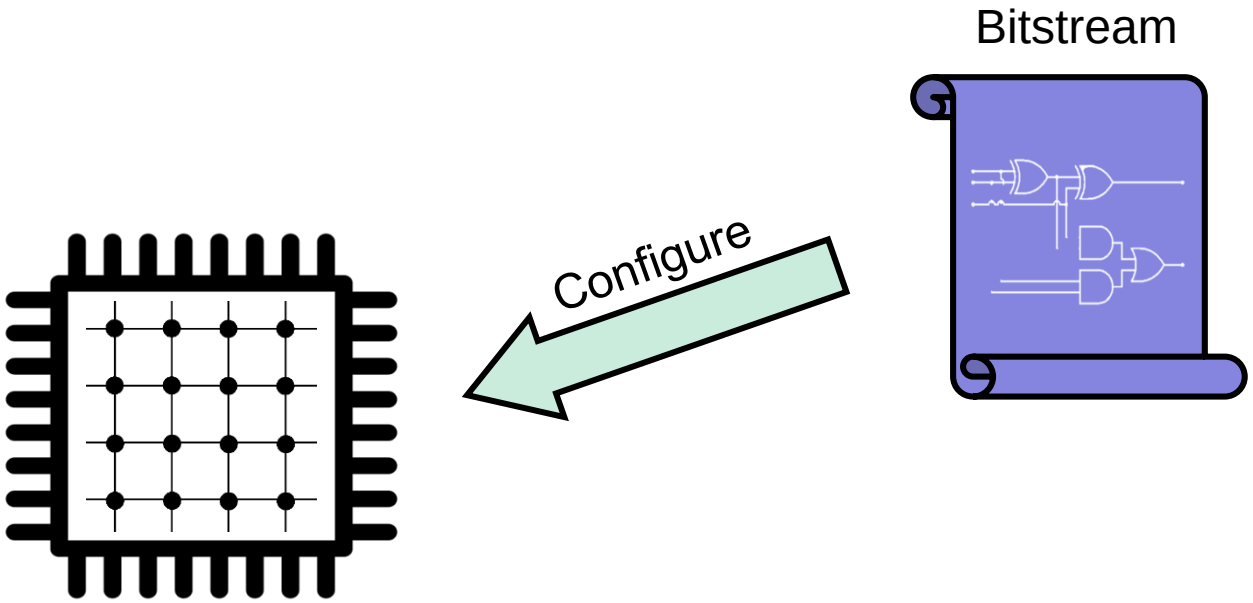
IND-CCA – Indistinguishability against chosen-ciphertext attacks



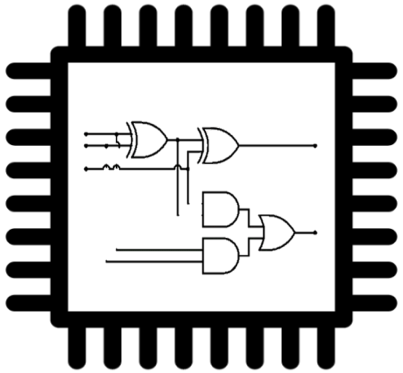
Restriction: not allowed to query $\mathrm{Dec}(K, C)$ after $\mathrm{Enc}(K, M)$ call returned C

Xilinx FPGA – Starbleed attack

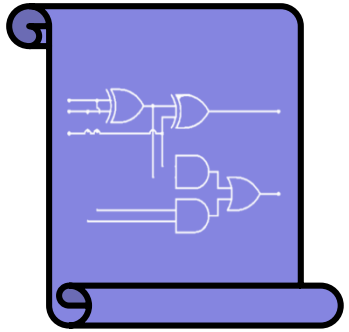
FPGA – Field Programmable Gate Array



FPGA – Field Programmable Gate Array

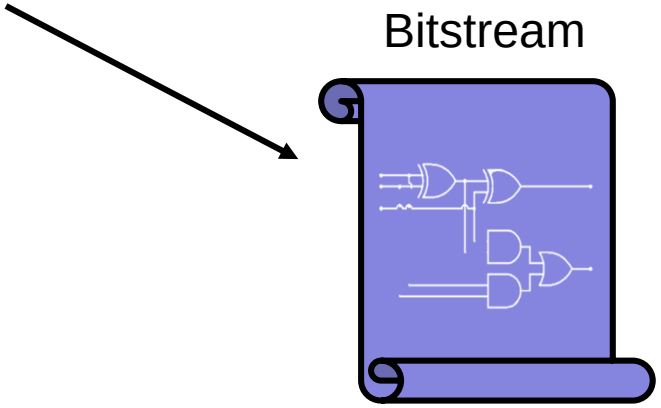


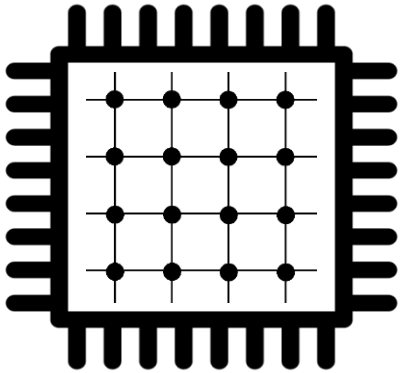
Bitstream



FPGA – Field Programmable Gate Array

Bitstream design typically a business secret
(or even a national/military secret)

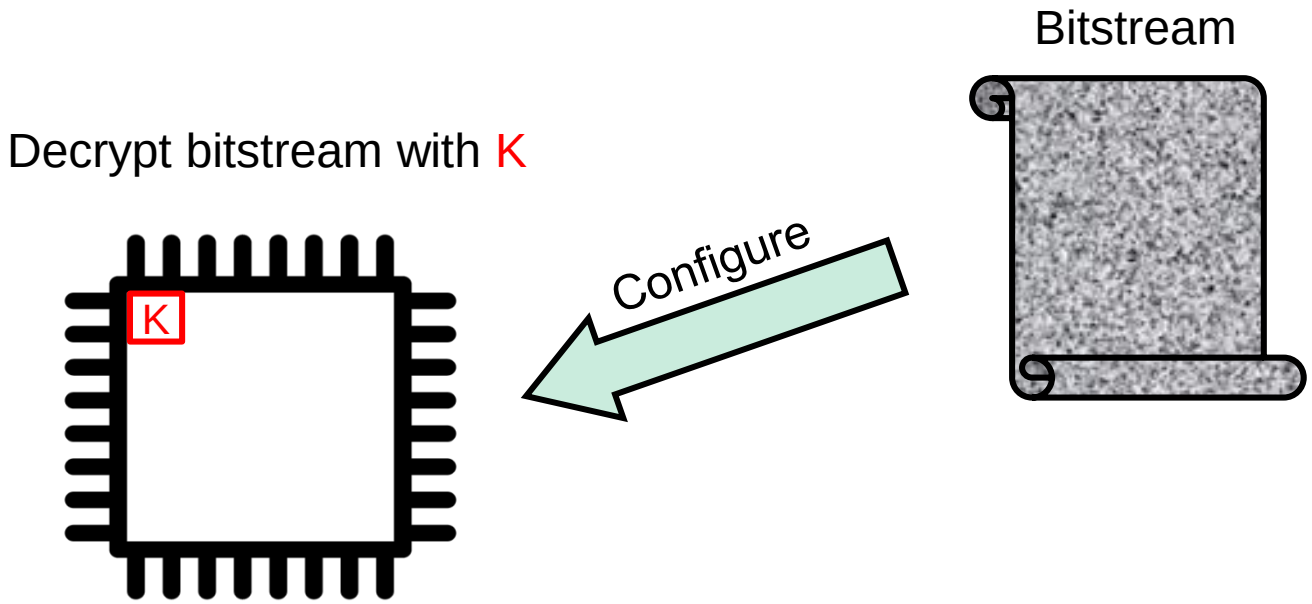




FPGA applications:

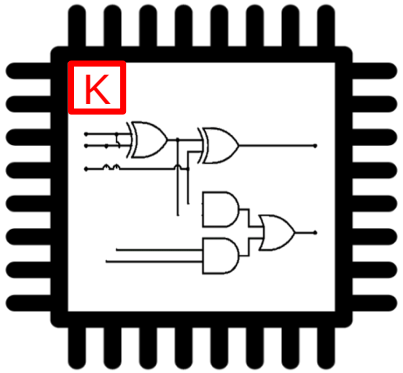
- Aerospace and avionics
- Digital signal processors
- Defense and military
- Medical devices
- General hardware accelerators (e.g. cryptography)

FPGA – Field Programmable Gate Array

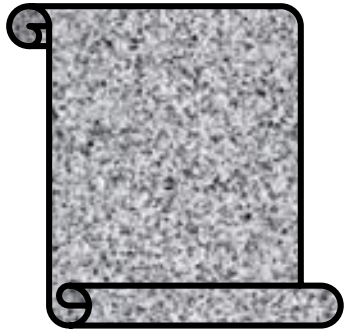


FPGA – Field Programmable Gate Array

Decrypt bitstream with K

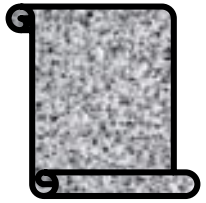


Bitstream

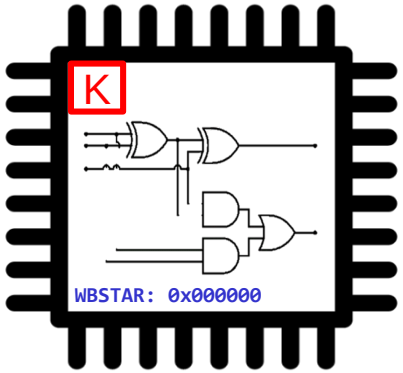


Xilinx Starbleed attack

Bitstream



Decrypt bitstream with K



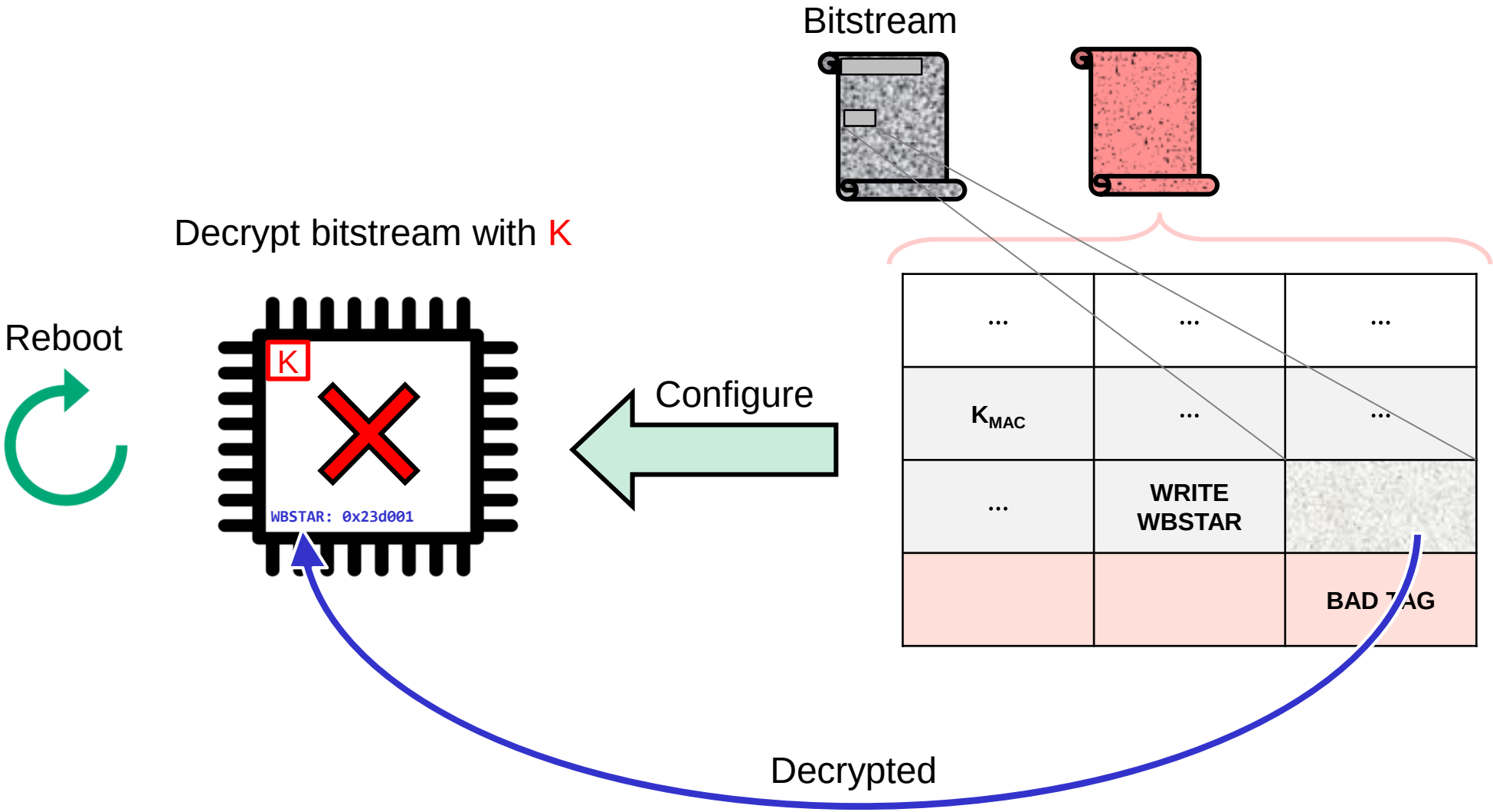
Header

AES-CBC encrypted

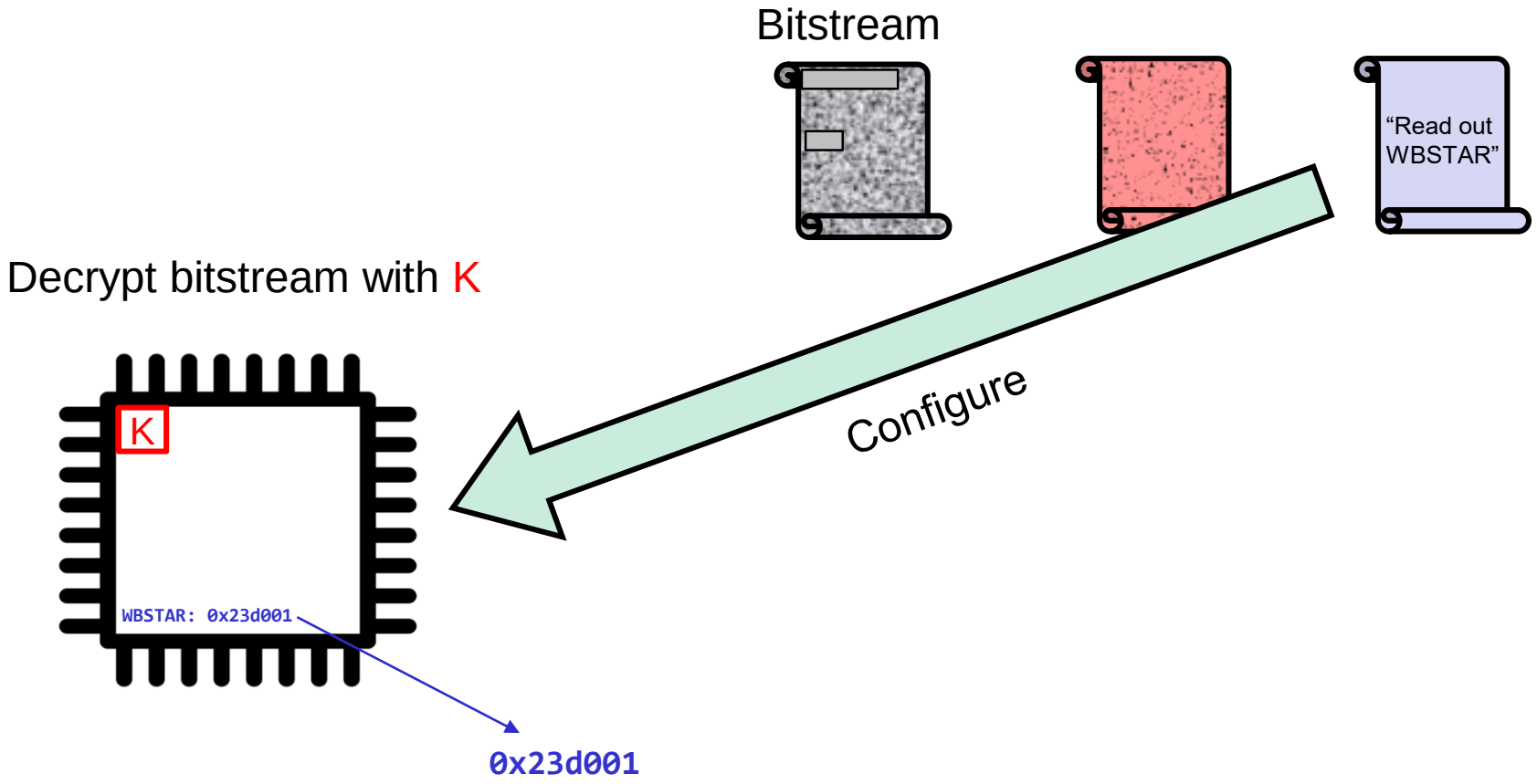
...	...	...
K_{MAC}	...	...
...	**WRITE WBSTAR**	**0x00000000**
		MAC TAG

WBSTAR = **W**arm-**B**oot **Star**t-address

Xilinx Starbleed attack

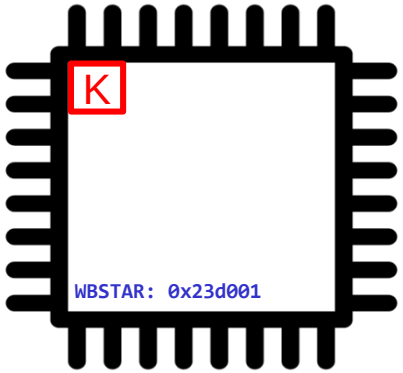


Xilinx Starbleed attack

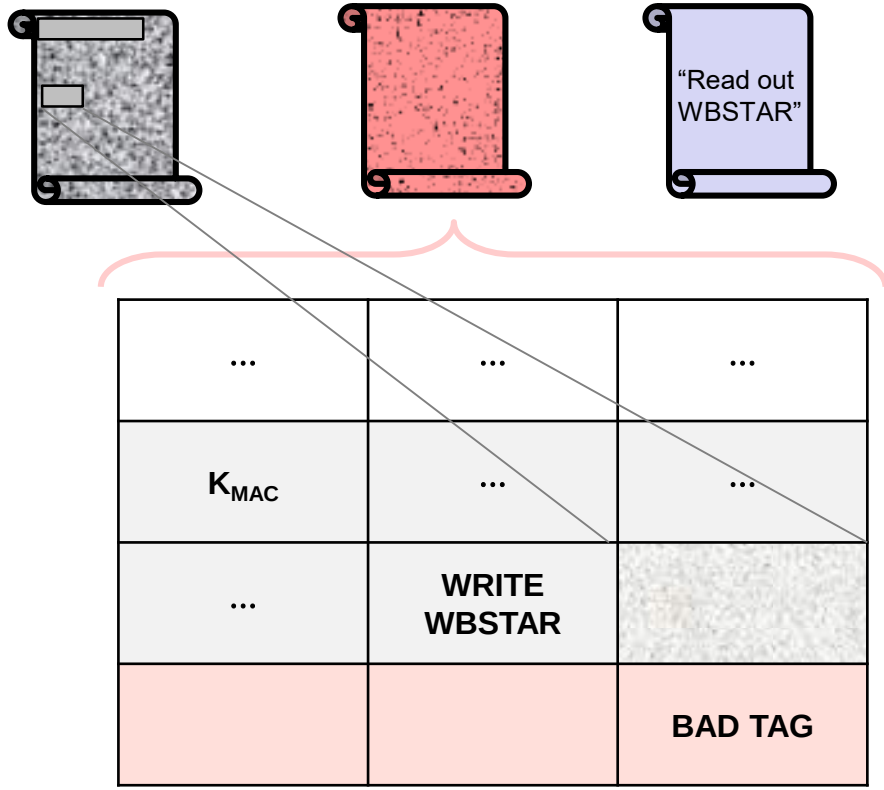


Xilinx Starbleed attack

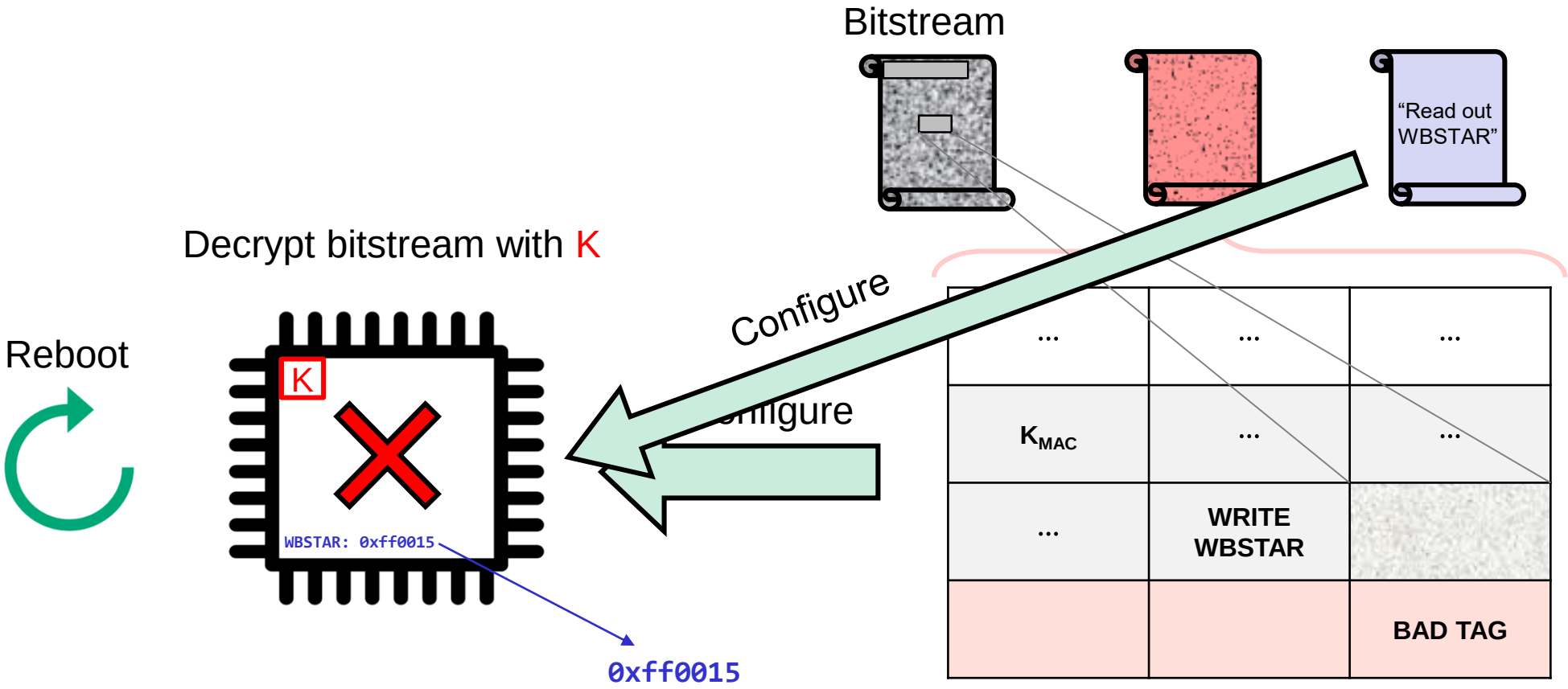
Decrypt bitstream with K



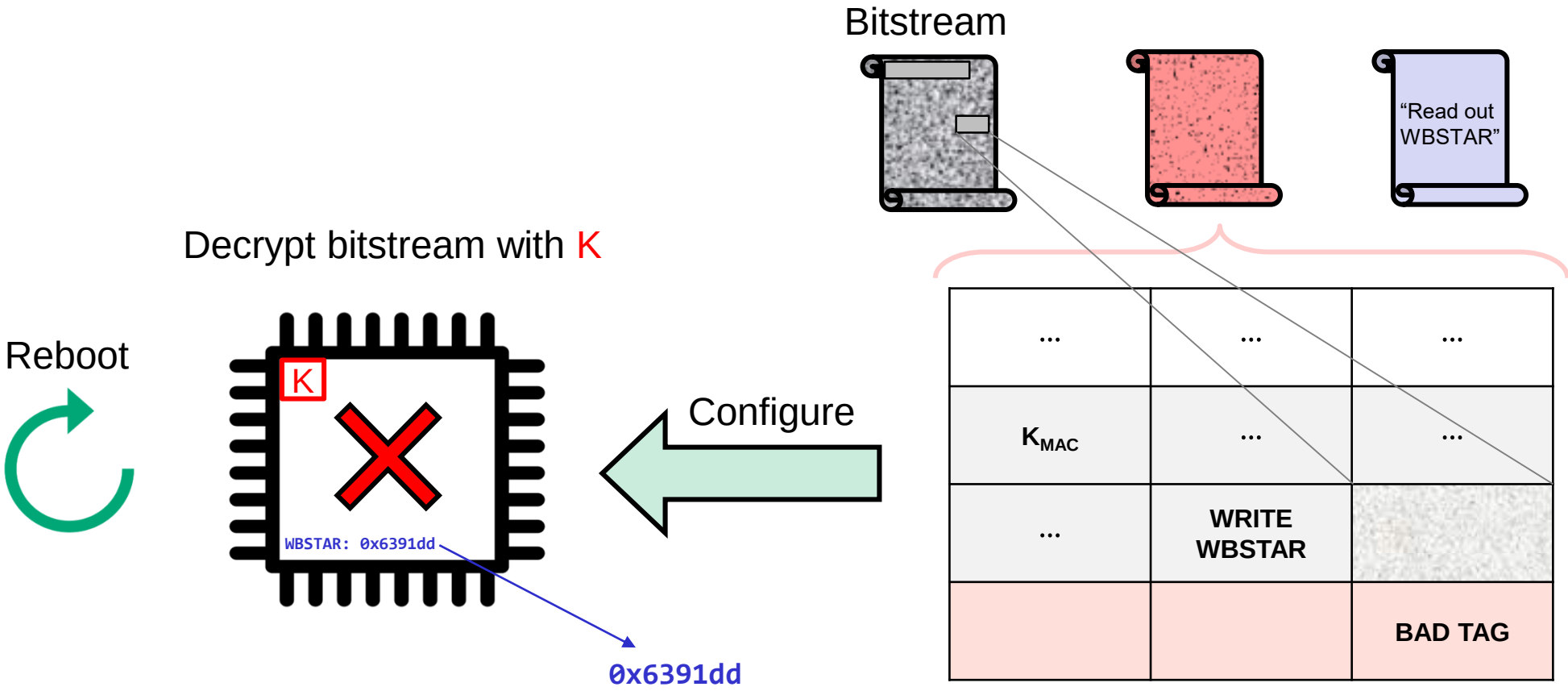
Bitstream



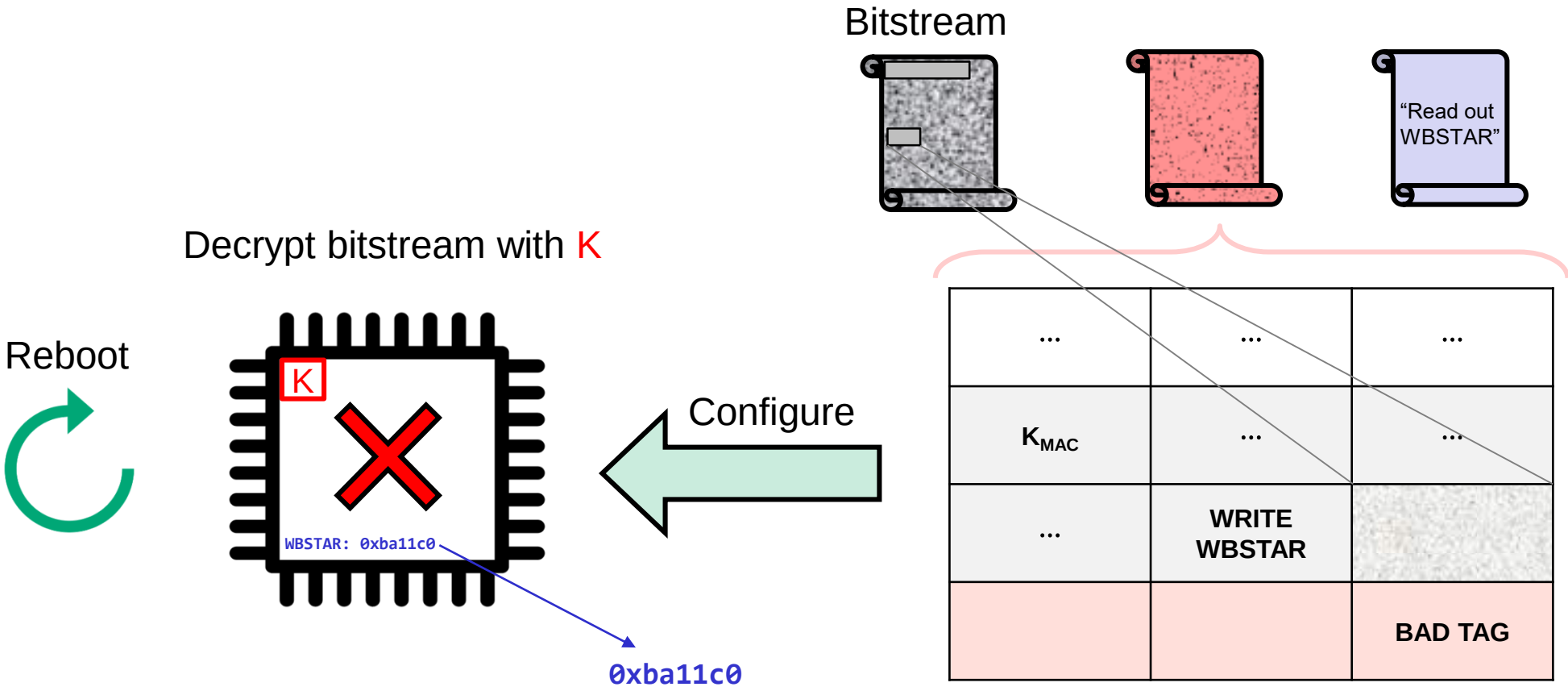
Xilinx Starbleed attack



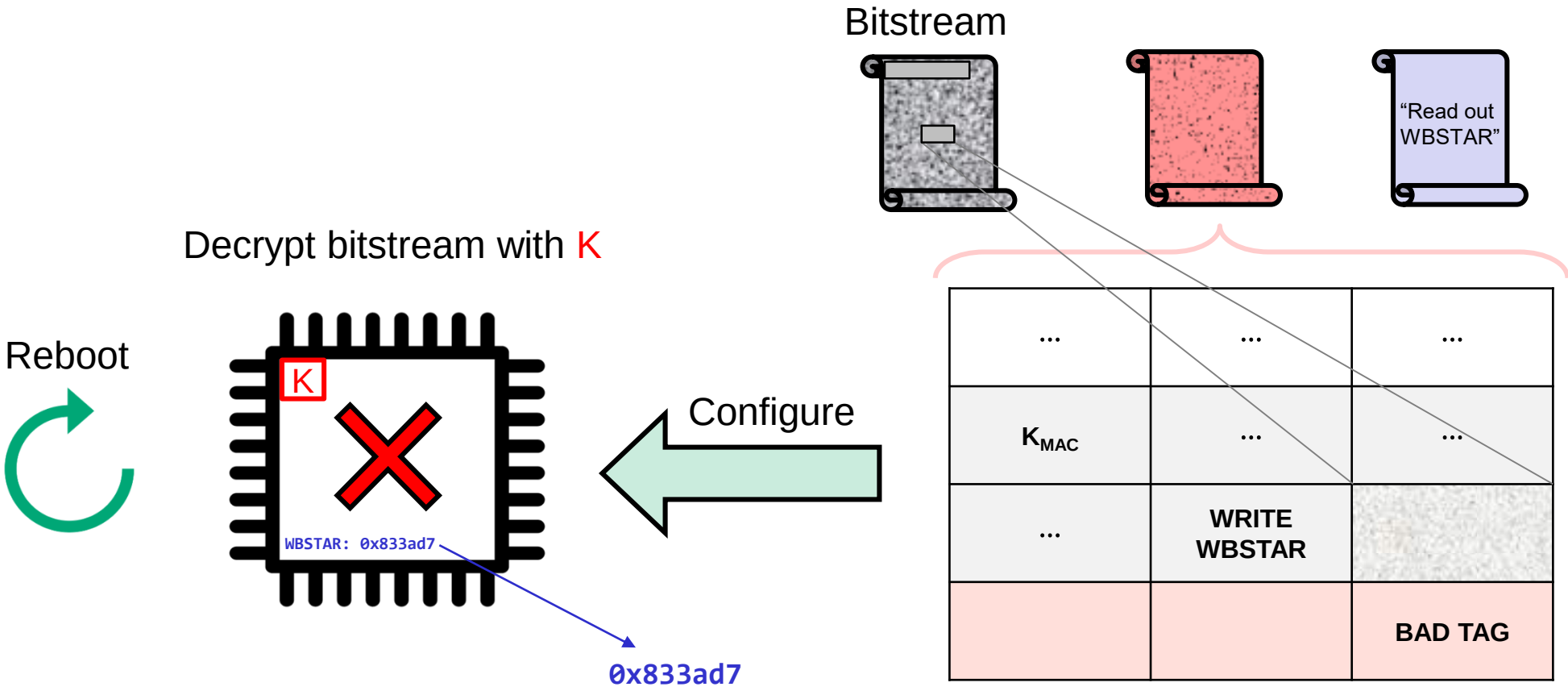
Xilinx Starbleed attack



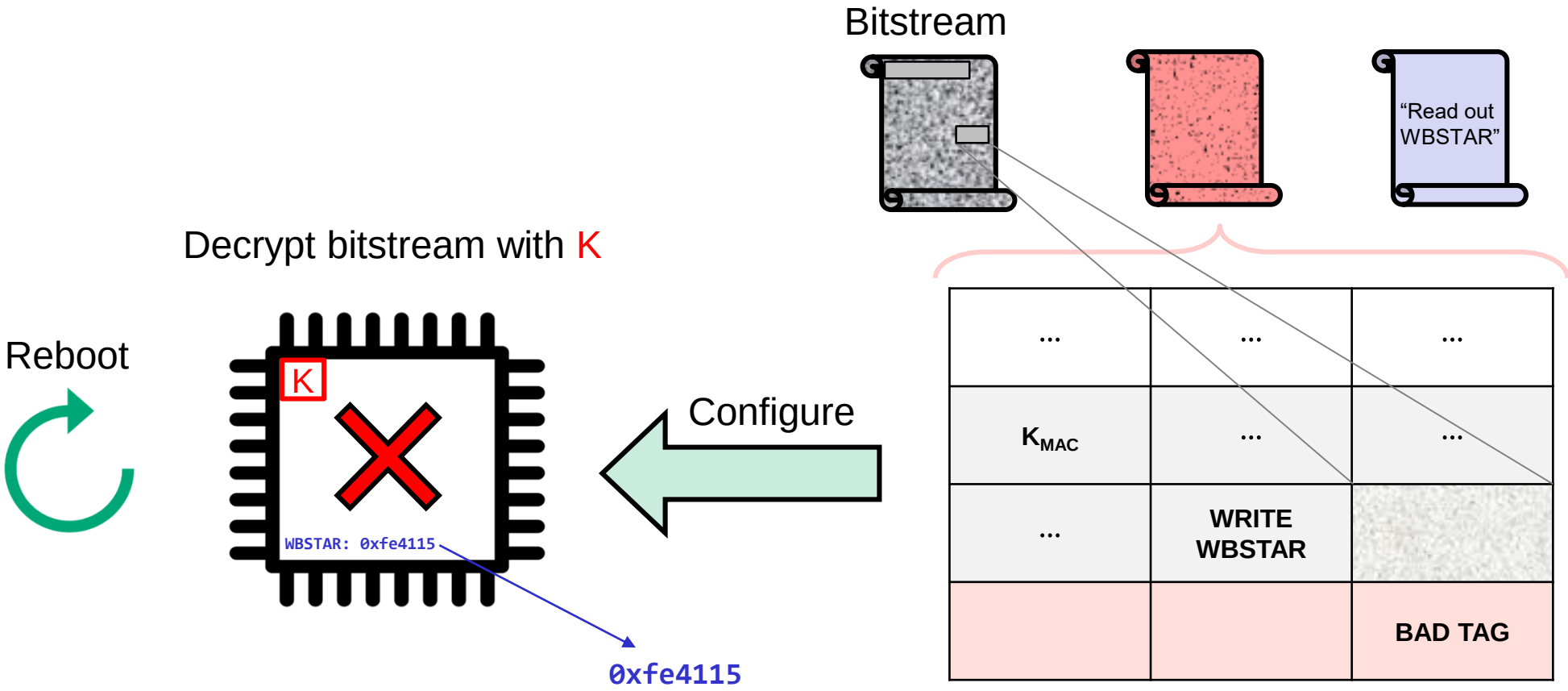
Xilinx Starbleed attack



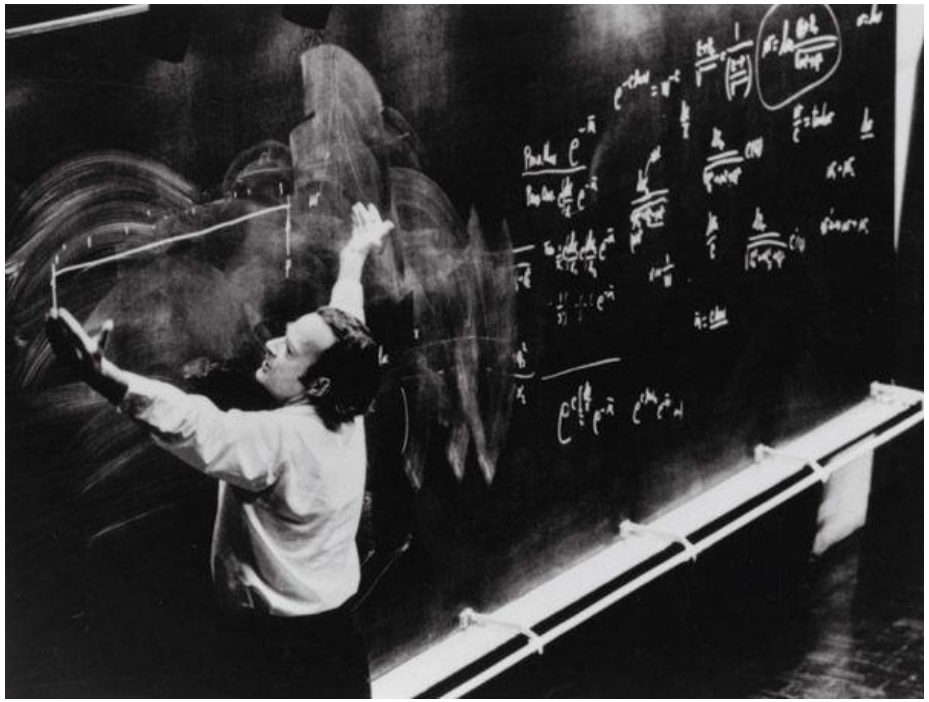
Xilinx Starbleed attack



Xilinx Starbleed attack



Time to fully decrypt bitstream: 26 hours



Authenticated encryption

Authenticated encryption

- **Authenticated encryption:** privacy *and* integrity from a single primitive:

- Syntactically (almost) the same as a normal encryption scheme

Authenticated encryption – syntax

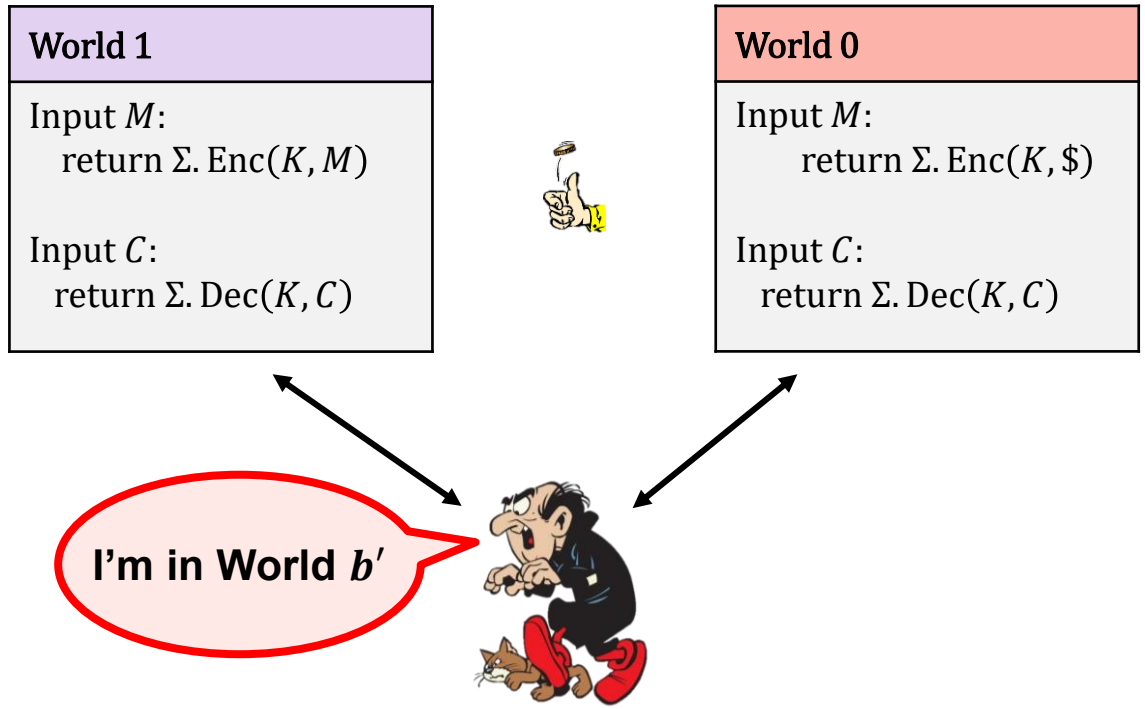
$\mathrm{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

$\mathrm{Enc}(K, M) = \mathrm{Enc}_K(M) = C$

$\mathrm{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$\mathrm{Dec}(K, C) = \mathrm{Dec}_K(C) = M \,/\, \perp$

Authenticated encryption – security



Restriction: not allowed to query $\mathrm{Dec}(K, C)$ after $\mathrm{Enc}(K, M)$ call returned C

Authenticated encryption – security

$\mathbf{Exp}^{\mathrm{ae}}_{\Sigma}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. Ciphertexts ← [ ]
3. $K \xleftarrow{\$} \Sigma.\mathrm{KeyGen}$
4. $b' \leftarrow A^{\mathcal{E}(\cdot),\ \mathcal{D}(\cdot)}$
5. **return** $b' \stackrel{?}{=} b$

$\mathcal{E}(M)$

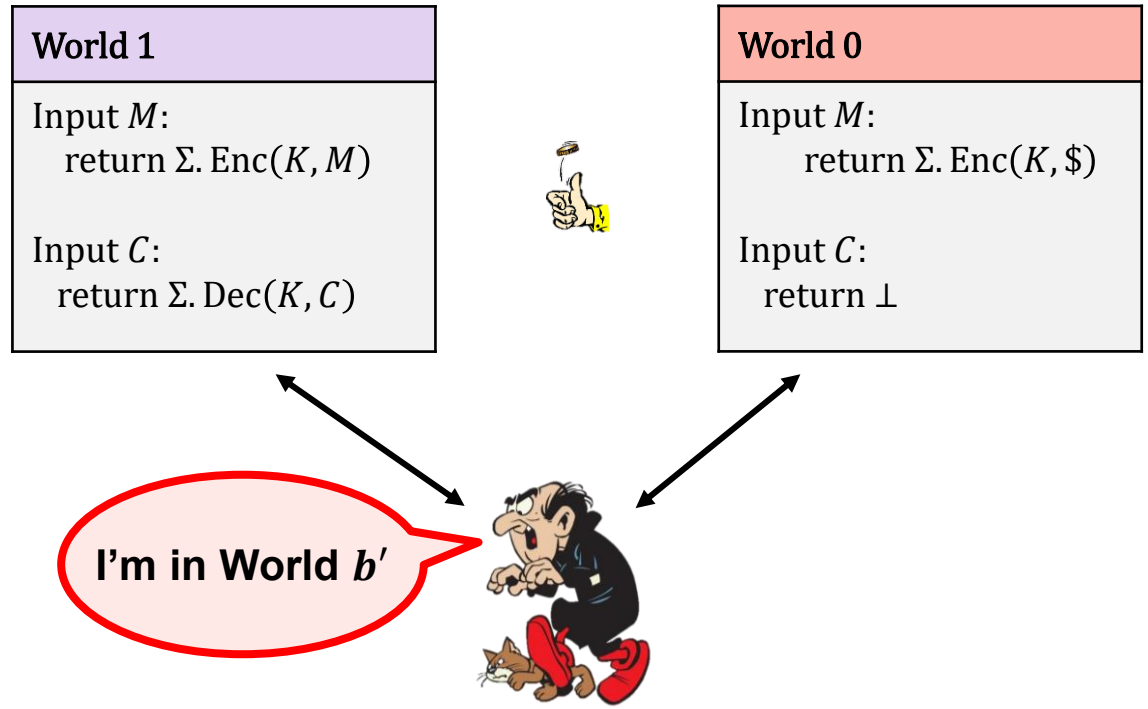
---

1. $R \xleftarrow{\$} \{0,1\}^{|M|}$
2. $C_0 \leftarrow \Sigma.\mathrm{Enc}(K, R)$
3. $C_1 \leftarrow \Sigma.\mathrm{Enc}(K, M)$
4. Ciphertexts.add(C_b)
5. **return** C_b

$\mathcal{D}(C)$

---

1. **if** $C \in$ Ciphertexts **then** // cheating!
2. **return** $\perp$
3. $M_0 \leftarrow \perp$
4. $M_1 \leftarrow \Sigma.\mathrm{Dec}(K, C)$
5. **return** M_b



Restriction: not allowed to query $\mathrm{Dec}(K, C)$ after $\mathrm{Enc}(K, M)$ call returned C

Definition: The **AE-advantage** of an adversary A is

$$\mathbf{Adv}^{\mathrm{ae}}_{\Sigma}(A) = |2 \cdot \Pr[\mathbf{Exp}^{\mathrm{ae}}_{\Sigma}(A) \Rightarrow \mathrm{true}] - 1|$$

AE security definition – implications

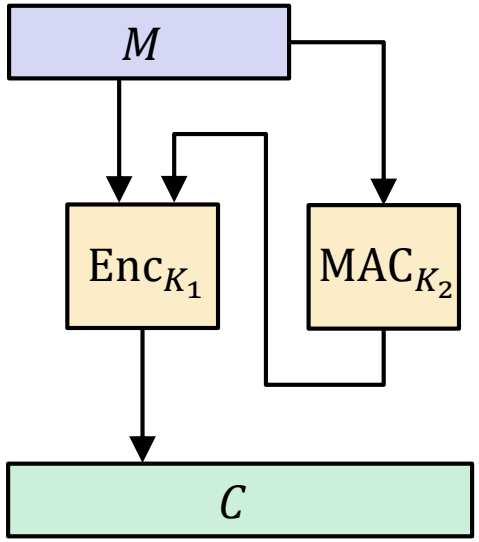
- **Privacy:** adversary cannot distinguish encryption of real messages from encryption random messages

- **Integrity:** adversary is not able to **forge** ciphertexts: any ciphertext *not* produced by the legitimate sender will decrypt to $\perp$

- Does *not* protect against **replay attacks**

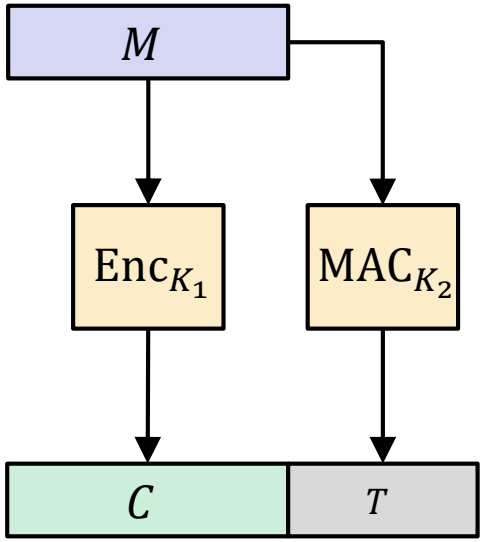
Generic composition: AE from Encryption + MAC

MAC-then-Encrypt (MtE)



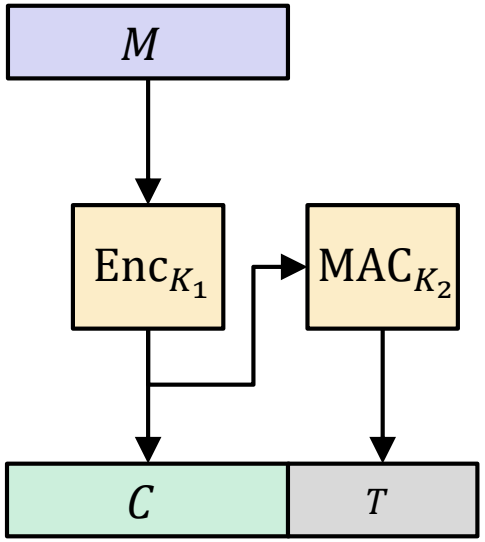
Used in TLS 1.2

Encrypt-and-MAC (E&M)



Used in SSH

Encrypt-then-MAC (EtM)



Used in IPsec

Generic composition: AE secure?

IND-CPA secure

UF-CMA secure

- E&M: $C \,||\, T \leftarrow \mathrm{Enc}(K_1, M) \,||\, \mathrm{MAC}(K_2, M)$ ✗

$$\mathrm{Enc}(K_1, M) \,||\, M \,||\, \overbrace{\mathrm{MAC}(K_2, M)}^{\mathrm{MAC}'(K_2, M)}$$

- MtE: $C \leftarrow \mathrm{Enc}\big(K_1, M \,||\, \mathrm{MAC}(K_2, M)\big)$ ✗

$$\underbrace{0000 \,||\, \mathrm{Enc}\big(K_1, M, \mathrm{MAC}(K_2, M)\big)}_{C}$$

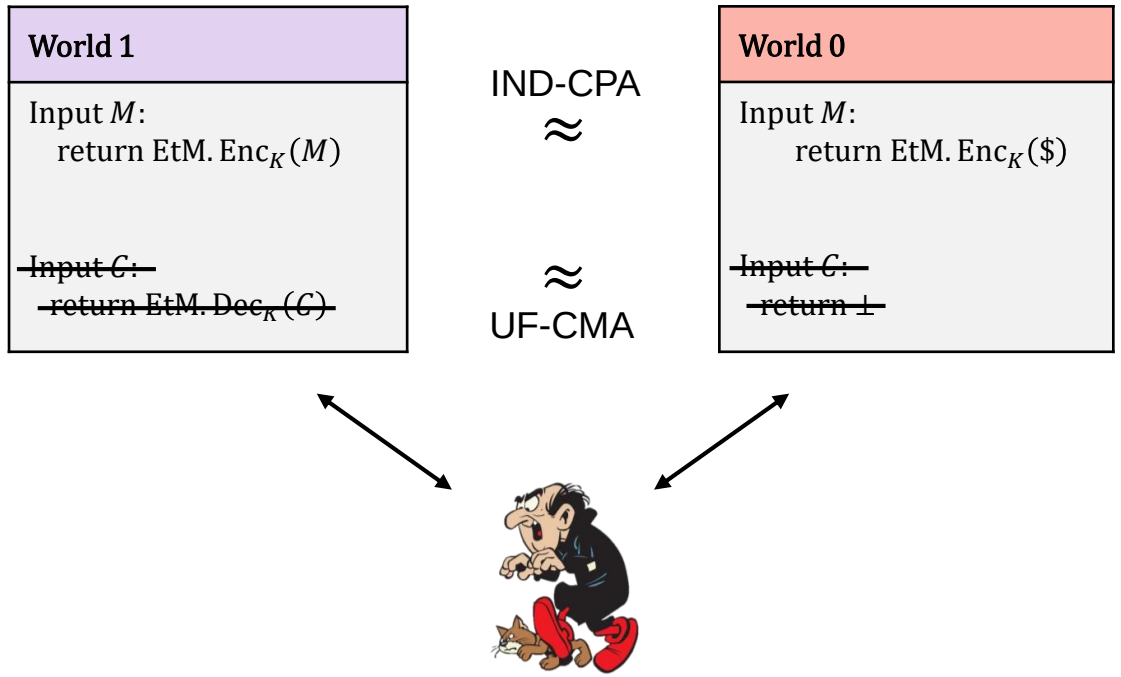
- EtM: $C \,||\, T \leftarrow \mathrm{Enc}(K_1, M) \,||\, \mathrm{MAC}(K_2, C)$ ✓

EtM – security

Theorem: for *any* AE adversary A against EtM there are adversaries B and C against Enc and MAC such that

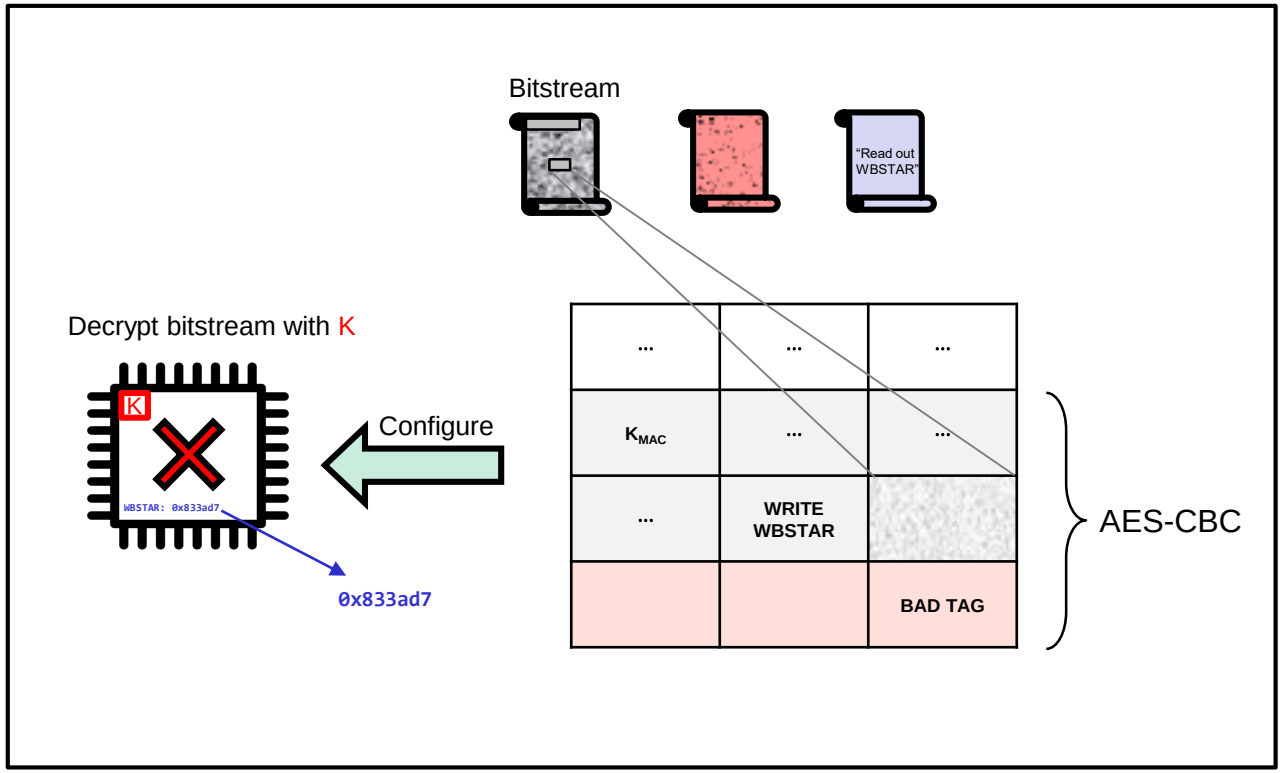
$$\mathbf{Adv}^{\mathrm{ae}}_{\mathrm{EtM}}(A) \leq \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathrm{Enc}}(B) + 2 \cdot \mathbf{Adv}^{\mathrm{uf\text{-}cma}}_{\mathrm{MAC}}(C)$$

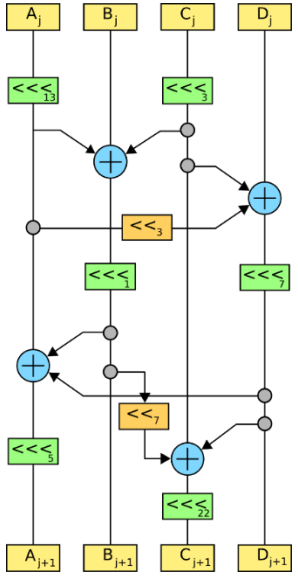
Proof sketch:

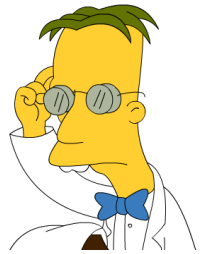


CCA-attacks – revisited

- MtE used in Starbleed attack
- Gave rise to (partial) decryption oracle
- Attack would not have been possible with an AE secure scheme

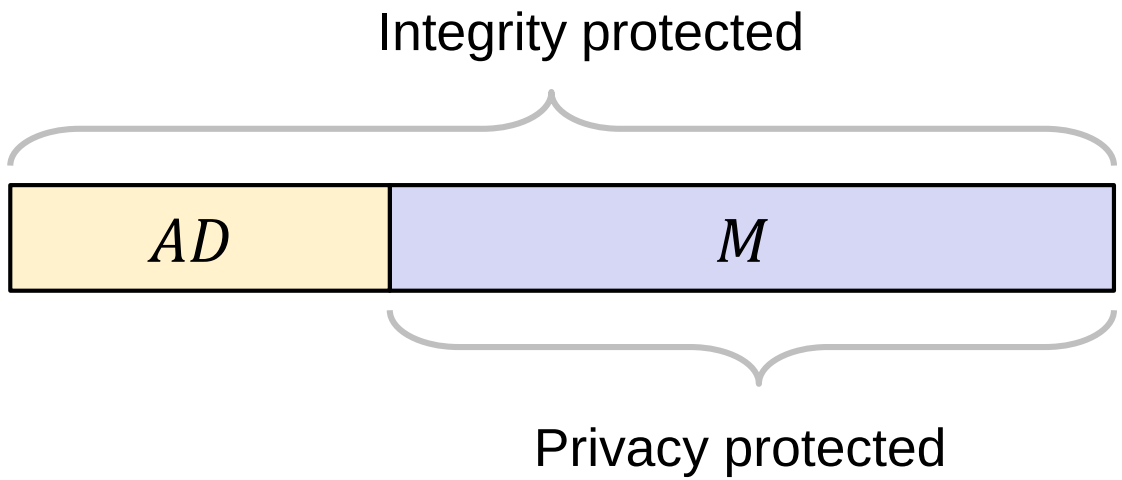


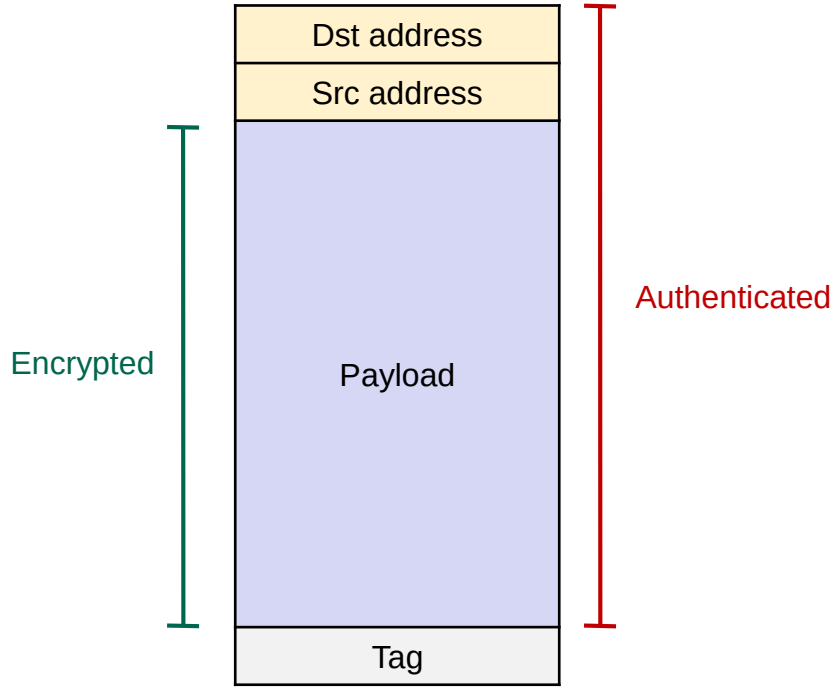




AEAD – Authenticated Encryption with Associated Data

AEAD – AE with associated data (AD)





- AD – data that can't be encrypted but still need integrity protection
  - Headers in protocols
  - Configuration data
  - Metadata

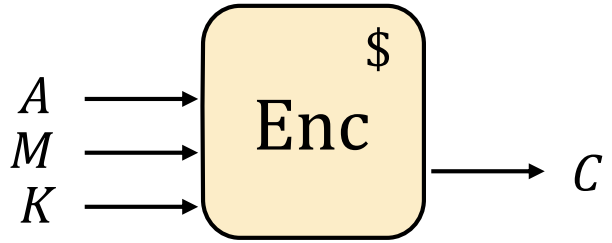
Authenticated encryption w/associated data (AEAD) – syntax

$\text{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

$\text{Enc}(K, M) = \text{Enc}_K(M) = C$

$\text{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$\text{Dec}(K, C) = \text{Dec}_K(C) = M/\perp$



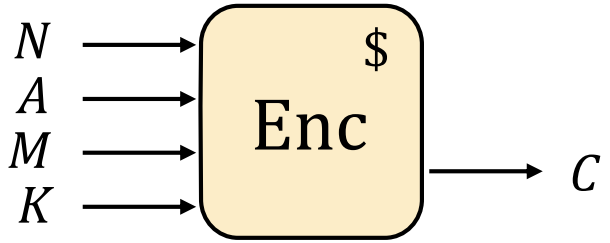
Authenticated encryption w/associated data (AEAD) – syntax

$\text{Enc} : \mathcal{K} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C}$

$\text{Enc}(K, A, M) = \text{Enc}_K(A, M) = \text{Enc}_K^A(M) = C$

$\text{Dec} : \mathcal{K} \times \mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$\text{Dec}(K, A, C) = \text{Dec}_K(A, C) = \text{Dec}_K^A(C) = M/\perp$



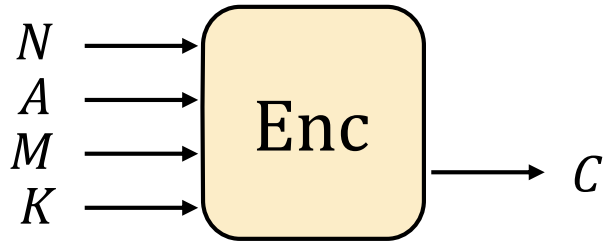
Authenticated encryption w/associated data (AEAD) – syntax

$\mathrm{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C}$

$\mathrm{Enc}(K, N, A, M) = \mathrm{Enc}_K^N(A, M) = \mathrm{Enc}_K^{N,A}(M) = C$

$\mathrm{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$\mathrm{Dec}(K, N, A, C) = \mathrm{Dec}_K^N(A, C) = \mathrm{Dec}_K^{N,A}(C) = M/\perp$



$\mathcal{K}$ – key space
$\mathcal{N}$ – nonce space
$\mathcal{A}$ – associated data space
$\mathcal{M}$ – message space
$\mathcal{C}$ – ciphertext space

Correctness requirement: $\forall K \in \mathcal{K}, \forall N \in \mathcal{N}, \forall A \in \mathcal{A}, \forall M \in \mathcal{M}$:

$$\mathrm{Dec}_K^{N,A}\left(\mathrm{Enc}_K^{N,A}(M)\right) = M$$

AEAD security (nonce-based)

$\mathbf{Exp}^{\mathbf{aead}}_{\Sigma}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. Nonces ← [ ]
3. Ciphertexts ← [ ]
4. $K \xleftarrow{\$} \Sigma.\mathrm{KeyGen}$
5. $b' \leftarrow A^{\mathcal{E}(\cdot,\cdot,\cdot),\ \mathcal{D}(\cdot,\cdot,\cdot)}$
6. **return** $b' \stackrel{?}{=} b$

$\mathcal{E}(N, A, M)$

---

1. **if** $N \in$ Nonces **then** // cheating!
2. **return** $\perp$
3. $R \xleftarrow{\$} \{0,1\}^{|M|}$
4. $C_0 \leftarrow \Sigma.\mathrm{Enc}(K, N, A, R)$
5. $C_1 \leftarrow \Sigma.\mathrm{Enc}(K, N, A, M)$
6. Nonces.add(N)
7. Ciphertexts.add(N, A, C_b)
8. **return** C_b

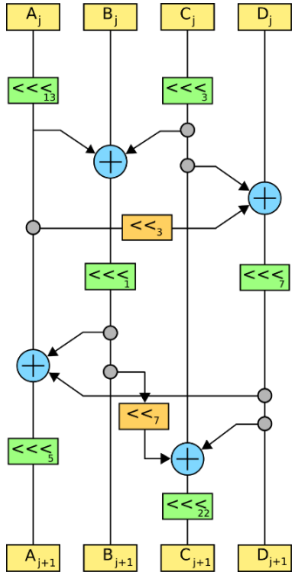
$\mathcal{D}(N, A, C)$

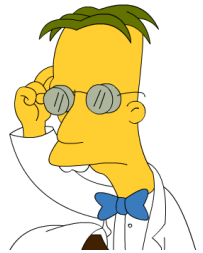
---

1. **if** $(N, A, C) \in$ Ciphertexts **then** // cheating!
2. **return** $\perp$
3. $C_0 \leftarrow \perp$
4. $C_1 \leftarrow \Sigma.\mathrm{Dec}(K, N, A, C)$
5. **return** C_b

Definition: The (nonc-based) **AEAD-advantage** of an adversary A is

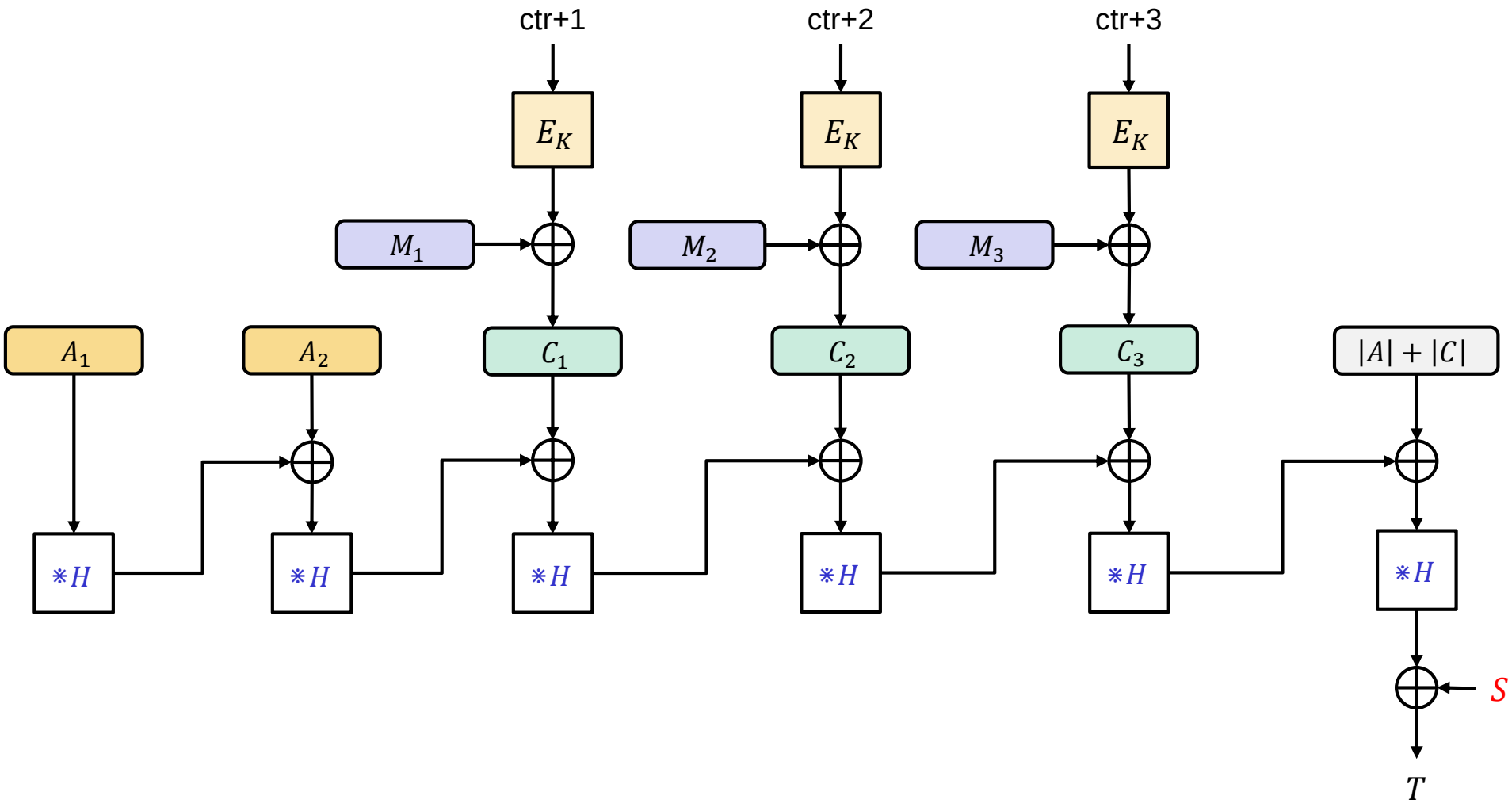
$$\mathbf{Adv}^{\mathrm{aead}}_{\Sigma}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}^{\mathrm{aead}}_{\Sigma}(A) \Rightarrow \mathrm{true}\right] - 1\right|$$





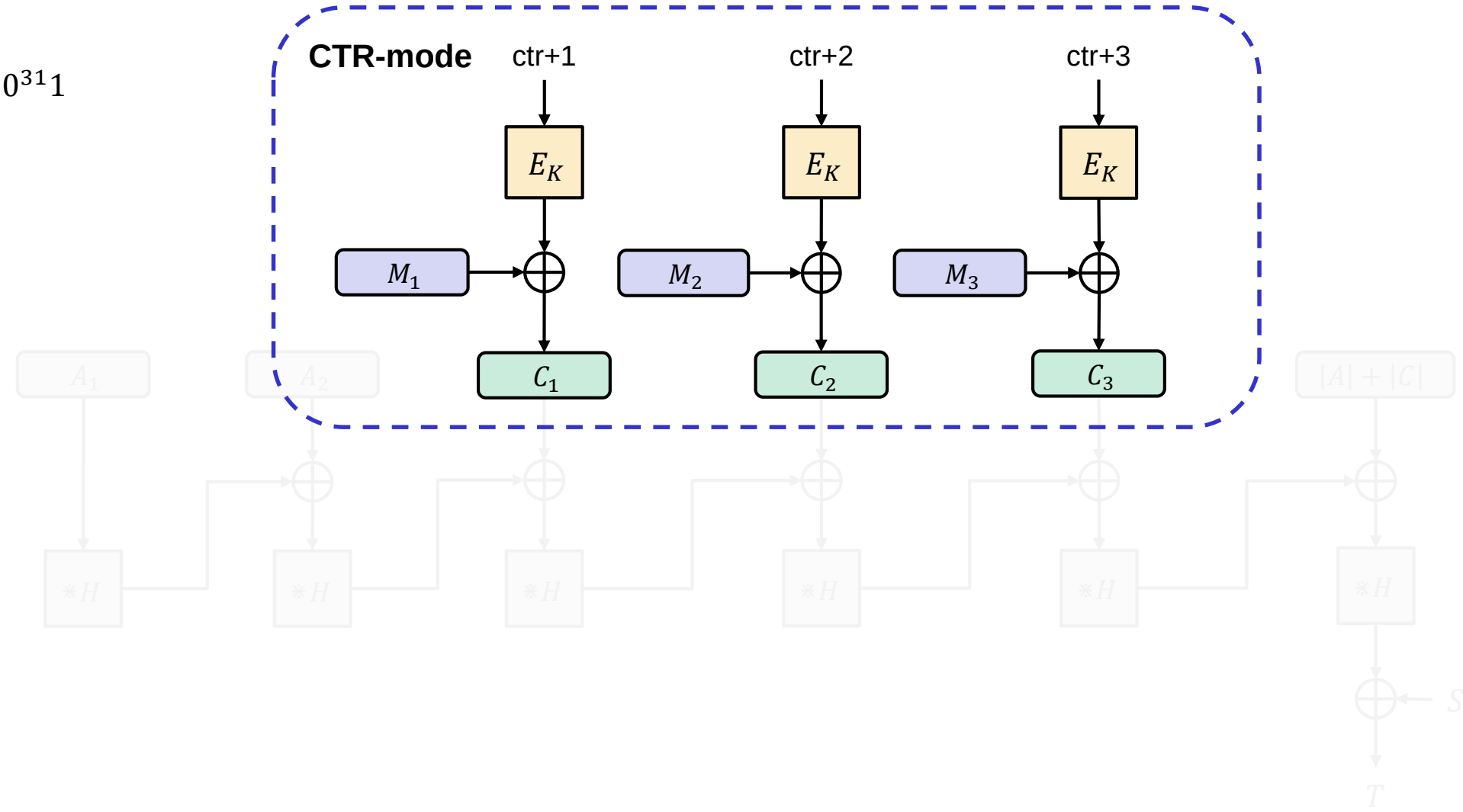
Constructing AEAD schemes

GCM – Galois/Counter Mode



GCM – Galois/Counter Mode

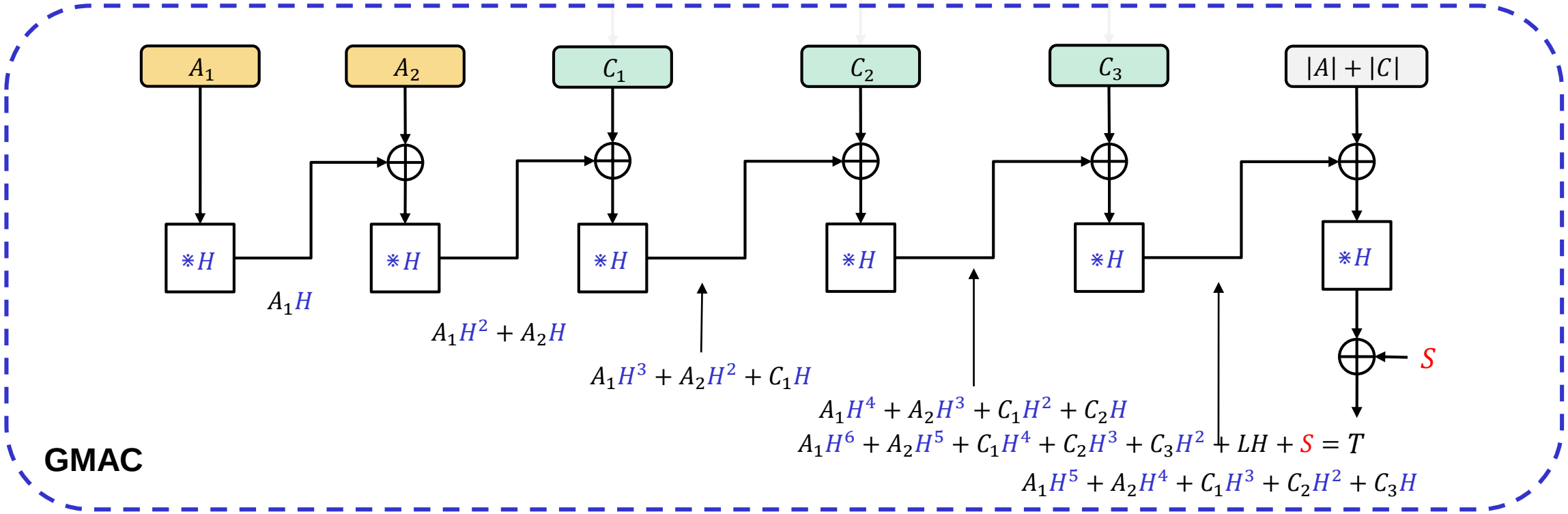
$\text{ctr} = N||0^{31}1$



GCM – Galois/Counter Mode

$H = E_K(0^{128})$

$S = E_K(N||0^{31}1)$

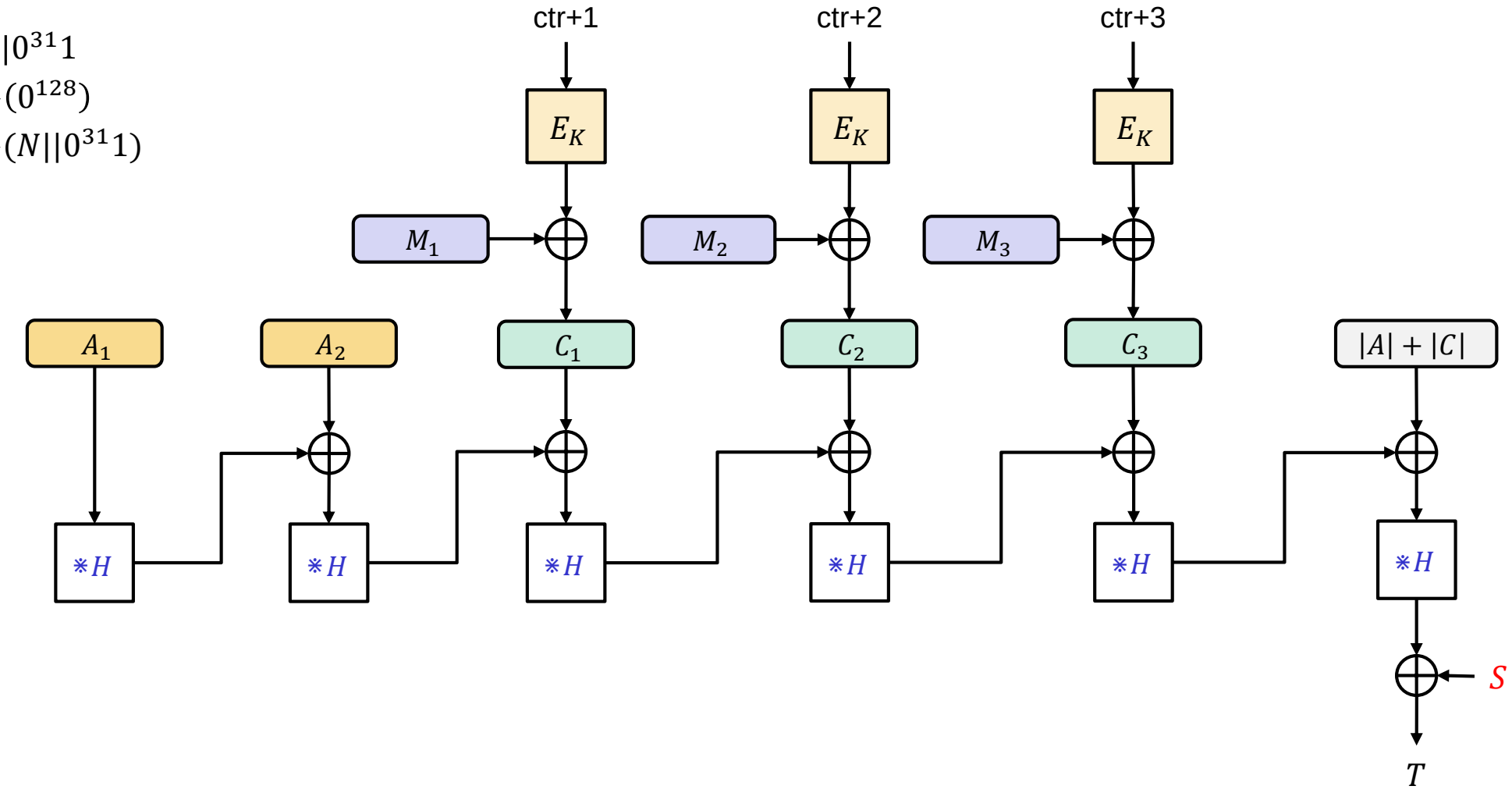


GCM – Galois/Counter Mode

$\text{ctr} = N||0^{31}1$

$H = E_K(0^{128})$

$S = E_K(N||0^{31}1)$



GCM – properties

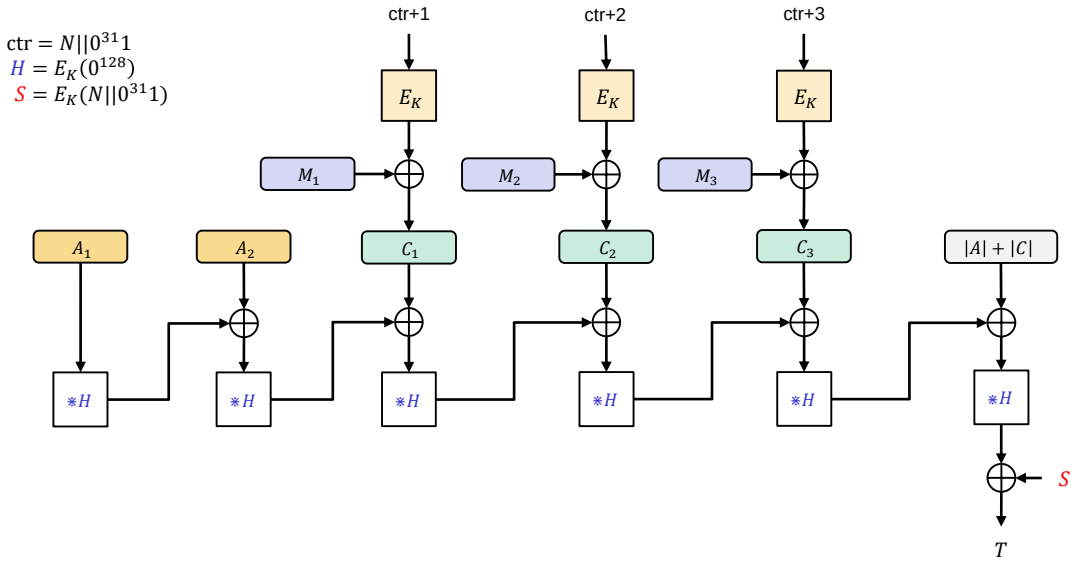
- **Theorem:** AE-secure if E is a secure PRF/PRP

- Very fast
  - Especially with AES-NI and Intel PCLMULQDQ instructions

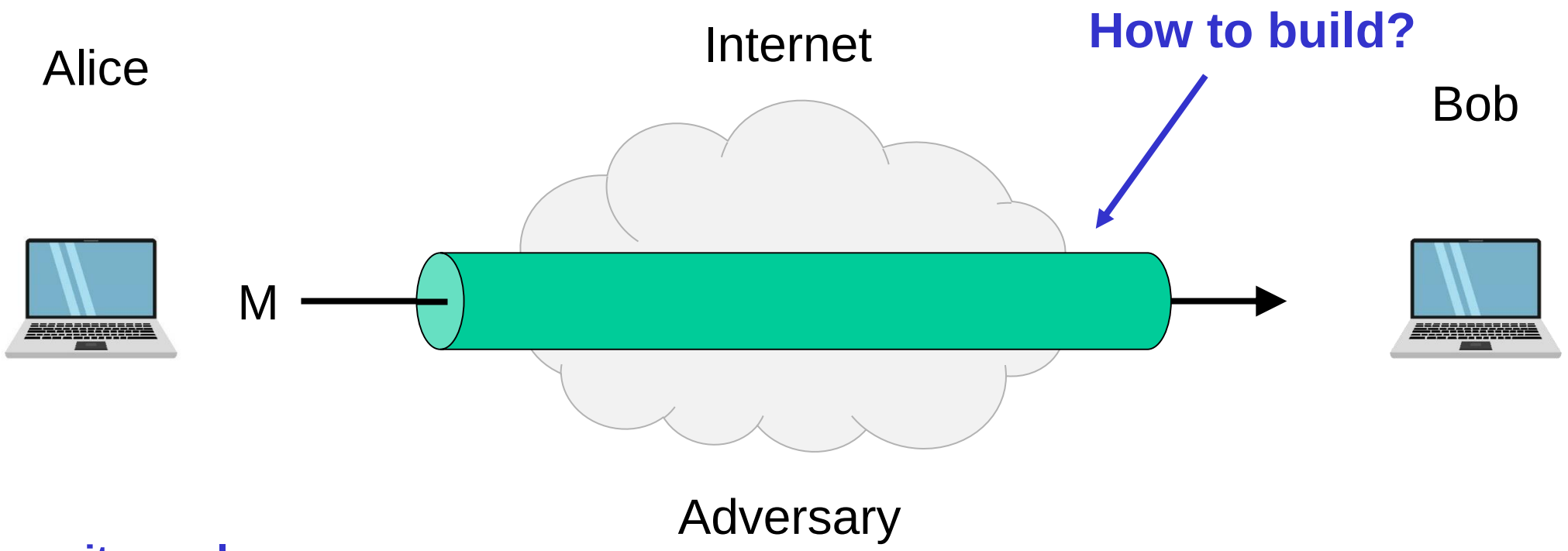
- Online
  - Doesn't need to know the length of the message before starting encryption

- Brittle
  - Nonce-reuse is *very* bad (see Problem set 5)
  - Tricky to implement correctly

- Used everywhere
  - Probably the most used mode on the Internet



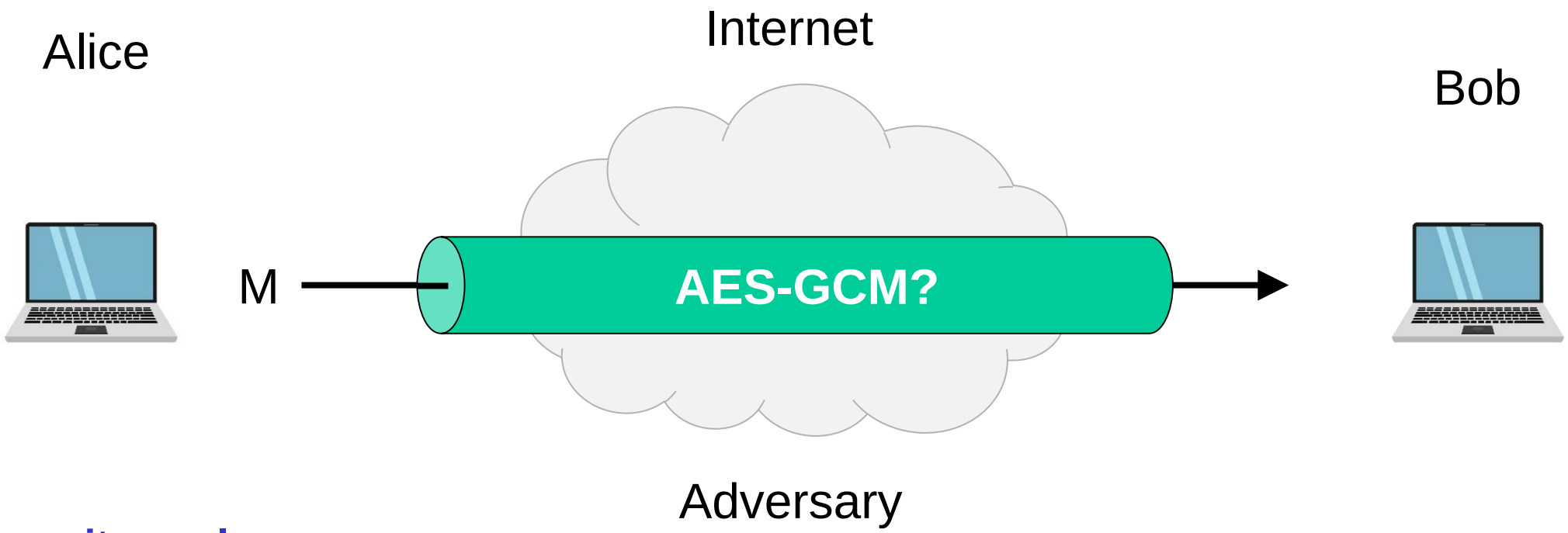
Ideal solution: secure channels



Security goals:

- **Data privacy:** adversary should not be able to read message M ✓
- **Data integrity:** adversary should not be able to modify message M ✓
- **Data authenticity:** message M really originated from Alice ✓

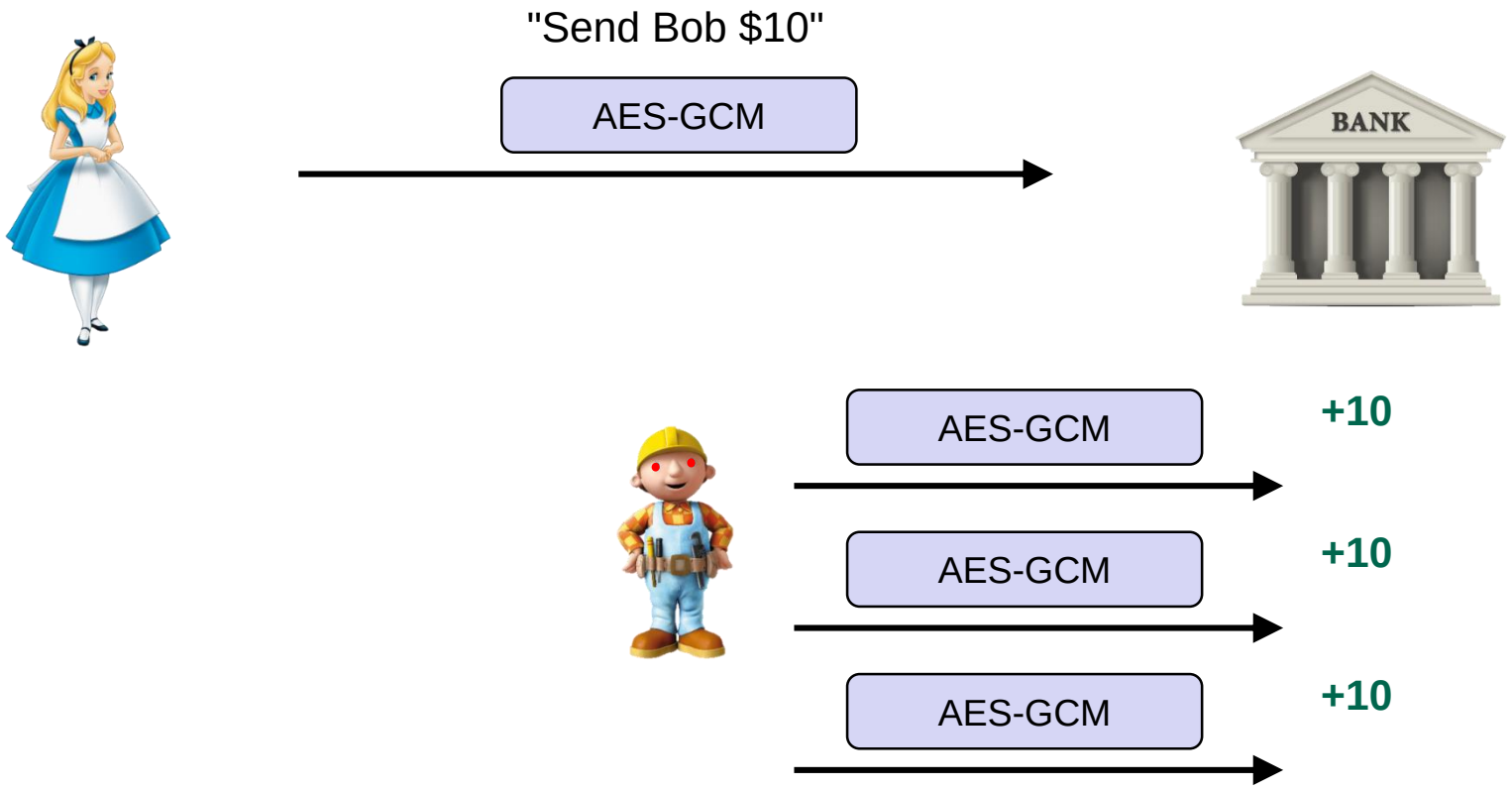
Ideal solution: secure channels



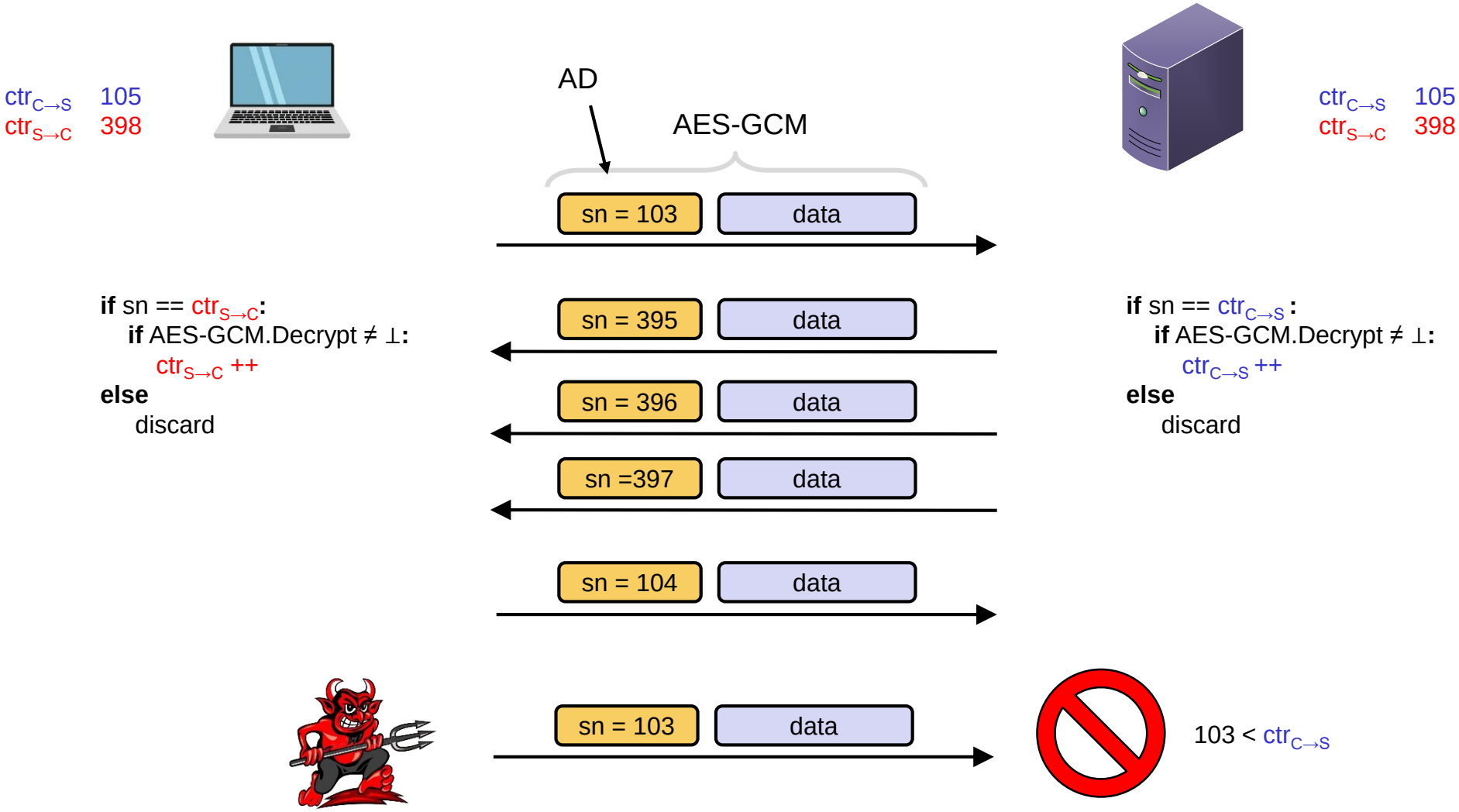
Security goals:

- **Data privacy:** adversary should not be able to read message M ✓
- **Data integrity:** adversary should not be able to modify message M ✓
- **Data authenticity:** message M really originated from Alice ✓

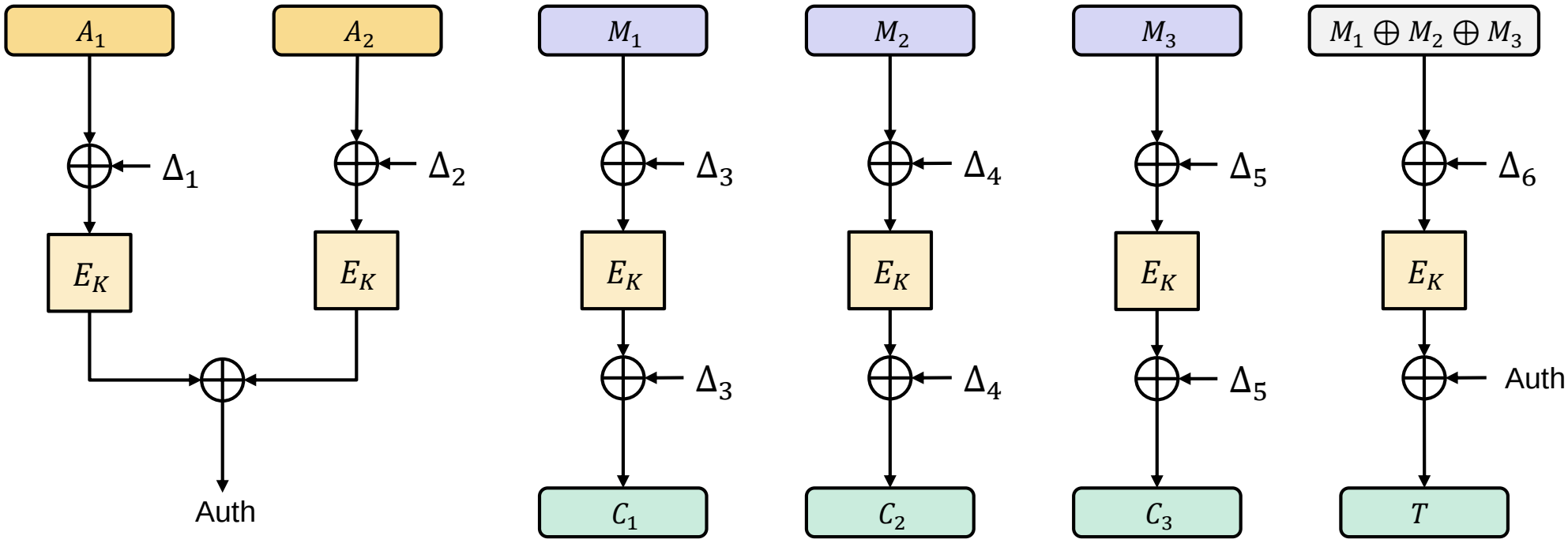
AES-GCM = secure channel?



Secure channels – TLS / IPsec



OCBv3 – Offset Codebook Mode



Δ_i - derived from 96-bit nonce N

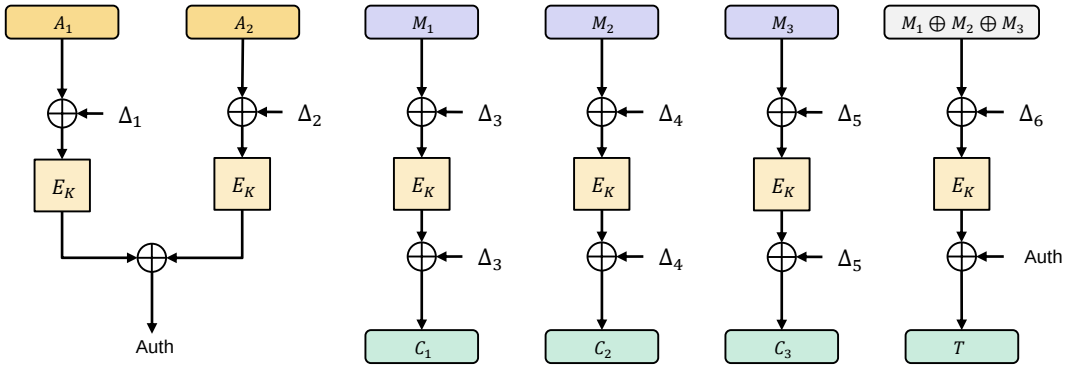
OCBv3 – properties

- **Theorem:** AE secure if E is a secure PRF
- The fastest AEAD algorithm in the west
- Fully parallelizable and online
- Incremental
- Hardly used anywhere due to patents

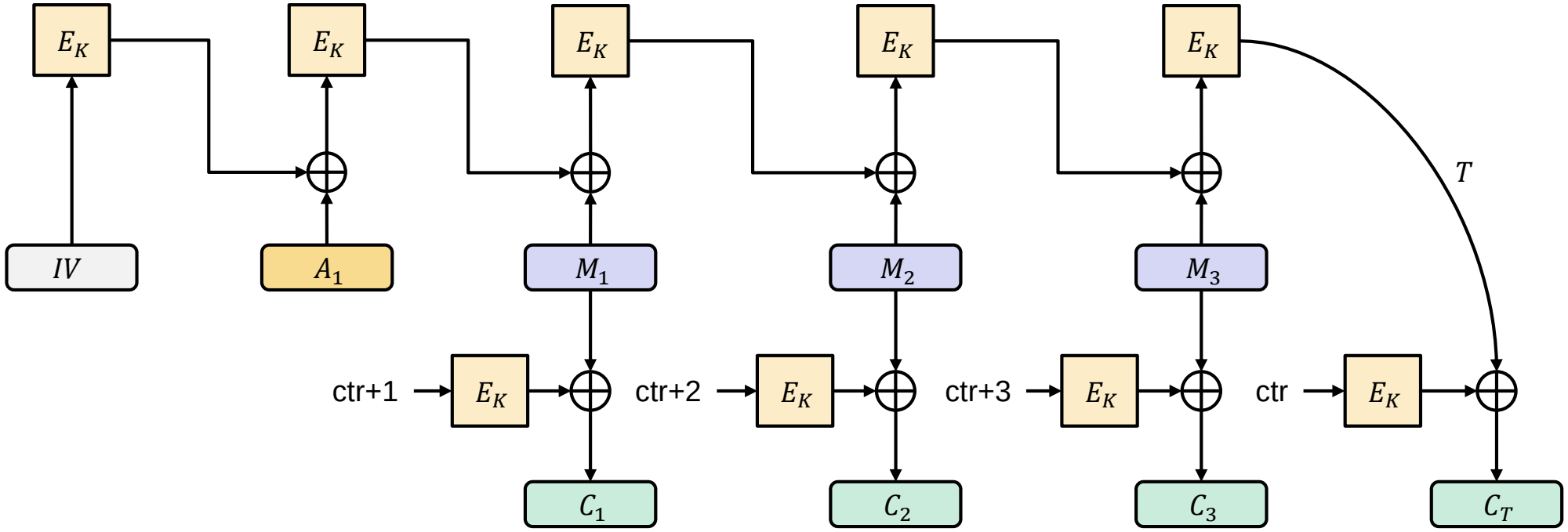
" If OCB was your kid, he' d play three sports and be on his way to Harvard. You' d brag about him to all your friends. "

" Unfortunately OCB is *not* your kid. It belongs to Philip Rogaway, who also happens to hold a patent on it."

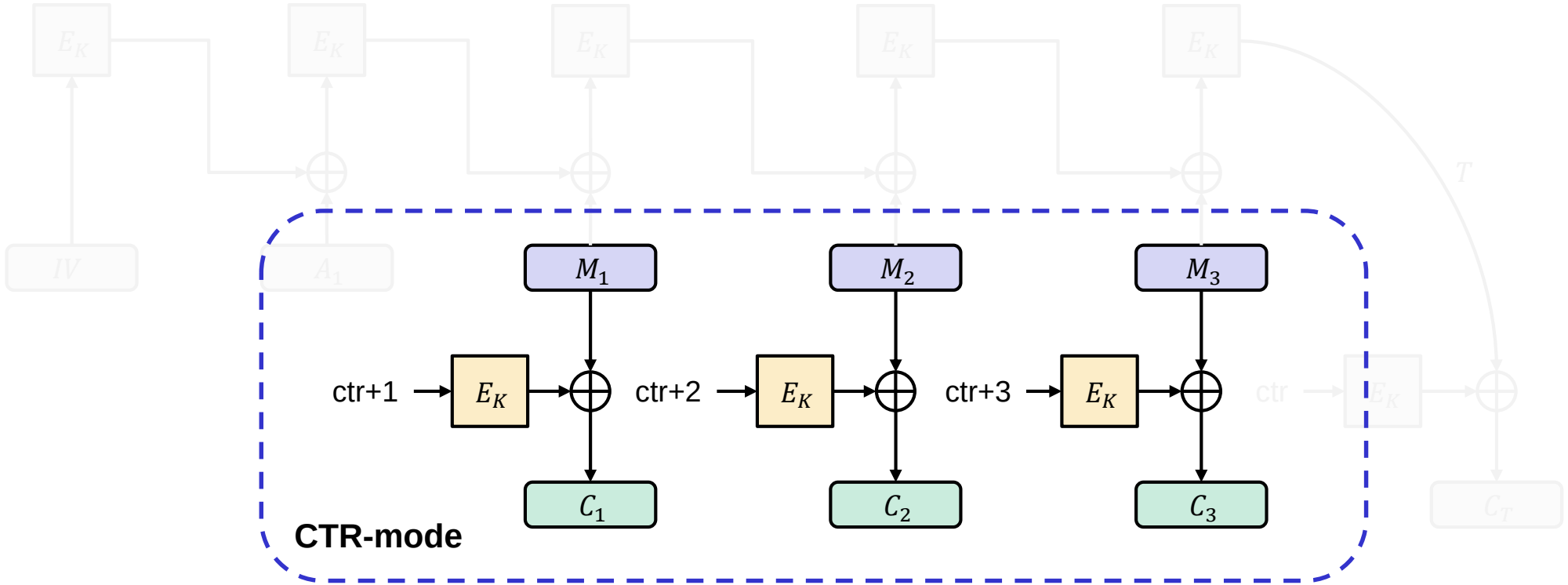
Matthew Green



CCM – Counter Mode with CBC-MAC

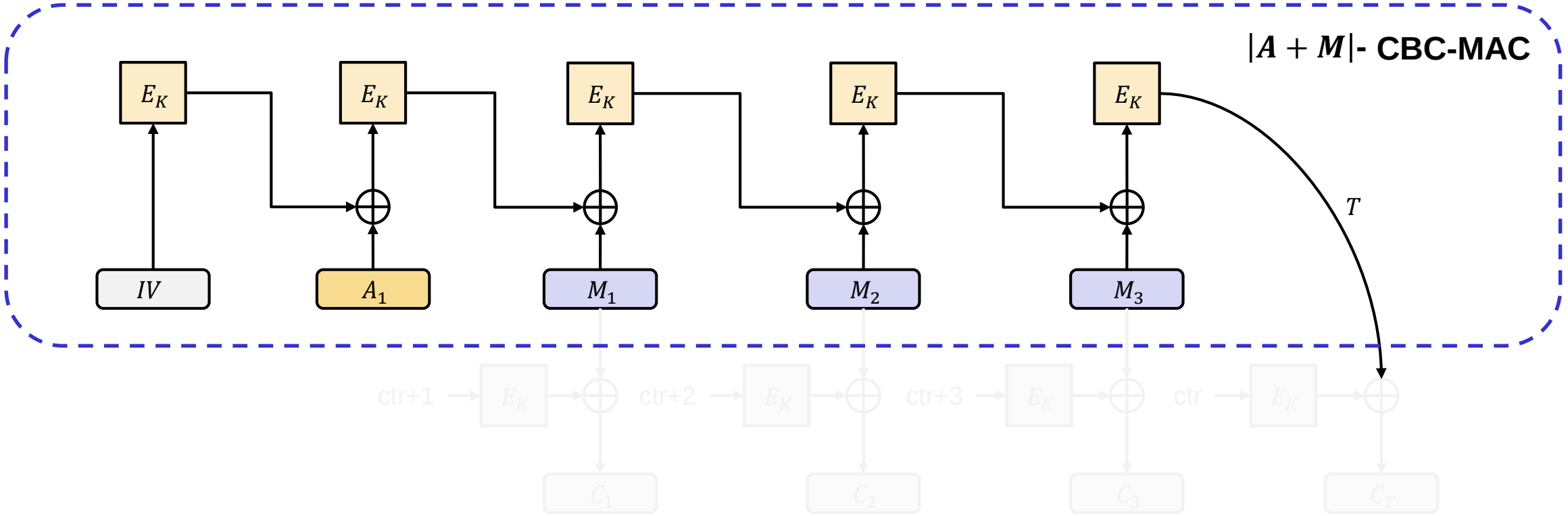


CCM – Counter Mode with CBC-MAC



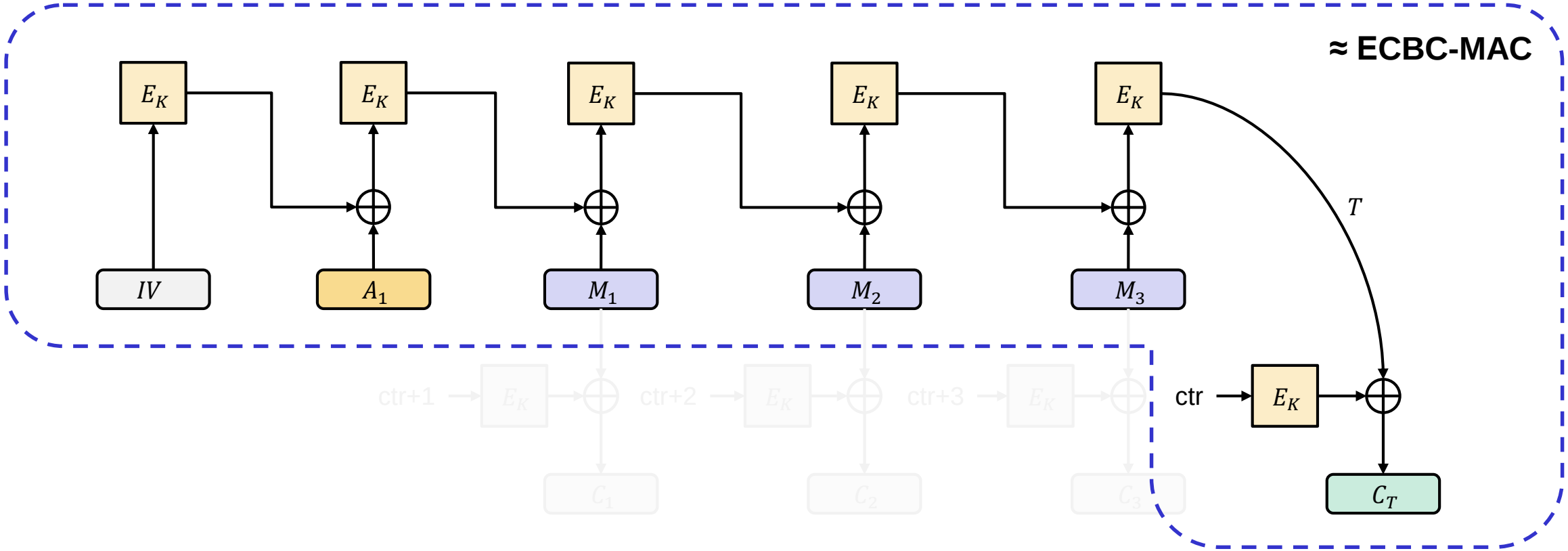
$\text{ctr} = \text{F}_1 || N || \, 0^{16}$

CCM – Counter Mode with CBC-MAC



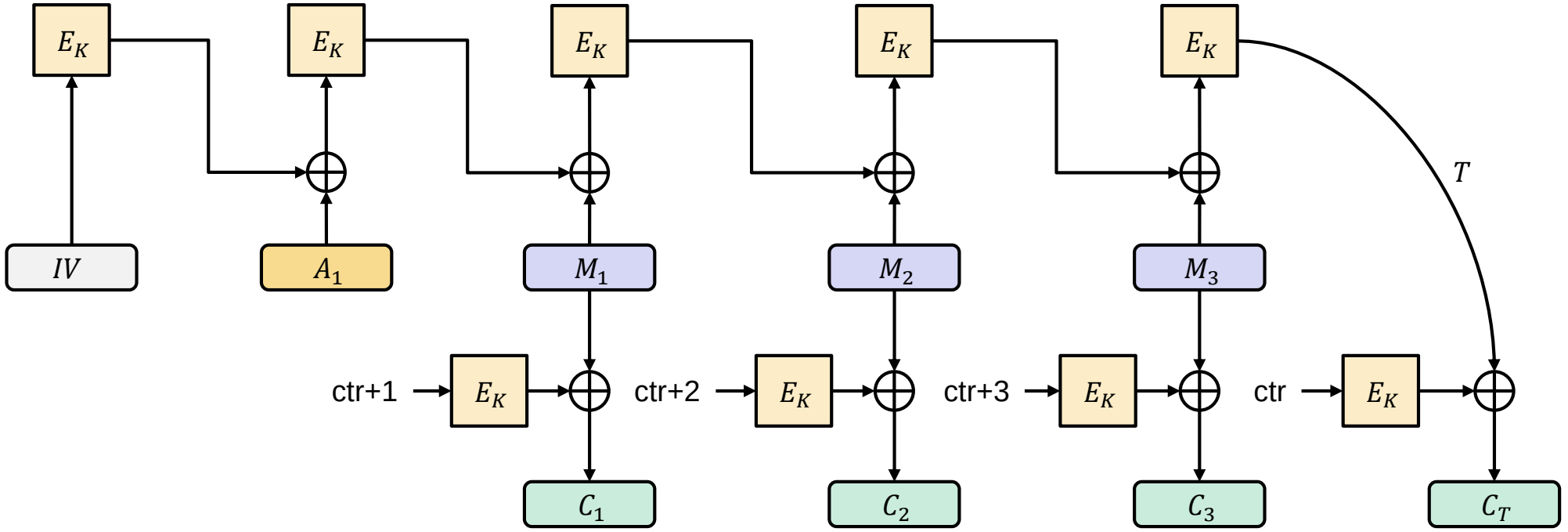
$$IV = \mathrm{F}_2 || N || \operatorname{len}_{16}(AD + M)$$

CCM – Counter Mode with CBC-MAC



$$IV = \mathrm{F}_2 || N || \operatorname{len}_{16}(AD + M)$$

CCM – Counter Mode with CBC-MAC



$\text{ctr} = \text{F}_1||N||\ 0^{16}$

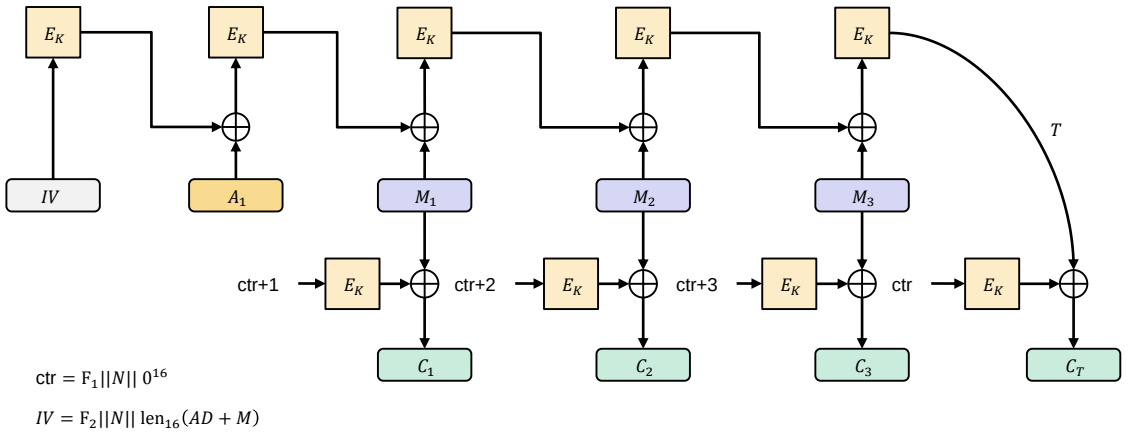
$IV = \text{F}_2||N||\ \text{len}_{16}(AD + M)$

CCM – properties

- **Theorem:** AE secure if E is a secure PRF
- Slow; needs 2 block cipher calls per message block
- Sequential; not online
- Clunky message encoding
- Royalty-free
  - Designed specifically as an alternative to OCB for use in WiFi (WPA2)
- Widely used
  - Default encryption algorithm in WPA2

" CCM is the 1989 Volvo station wagon of AEAD modes. It'll get you to your destination reliably, just not in a hurry."

Matthew Green



Summary

- Authenticated encryption: privacy + intergrity in one primitive
- AEAD: AE + *associated* data: data that gets integrity protection, but not privacy protection (e.g. protocol headers)
- AEAD examples:
  - GCM
  - CCM
  - OCB
- AEAD is *not* a secure channel!
  - Does not provide *replay* protection
  - Secure channels from AEAD: add counters/nonces/timers
- Next week: hash functions