Computational indistinguishability

Last time I promise

Unitary notation

- Writing numbers only using 1
  - 1 -> 1
  - 2- > 11
  - 3 -> 111
  - …
  - $n \rightarrow 1 \ldots 1$ (n times)
- 1^n is shorthand for writing n in unitary

Family of games

- Generalization of section 7.8 in the book (page 278)
- A family (ensemble) of games $X = \{X_n\}_{n\in\mathbb{N}}$ assigns to each natural number n a game X_n

Distinguisher

- Definition of distinguisher
  - Input
    - Game g
    - Index i written in unitary
  - Outputs a bit $b \in \{0,1\}$

Computational indistinguishability

- DEFINITION: Two Family of games $X = \{X_n\}_{n\in\mathbb{N}}$, $\mathrm{Y} = \{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if

For every efficient distinguisher D, there exists a negligible function μ

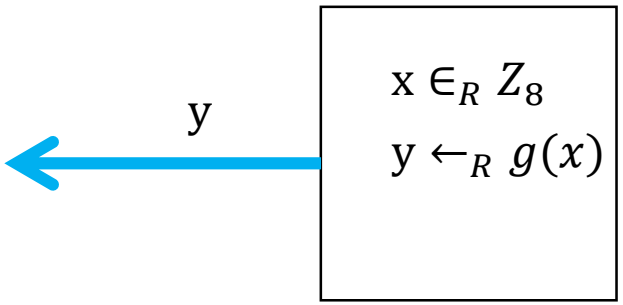
$$|\Pr[D(X_n, 1^n) = 1] - \Pr[D(Y_n, 1^n) = 1\}| \leq \mu(n)$$

- Equivalent to saying the distinguisher cannot do much better than guessing if he is given X_n or Y_n

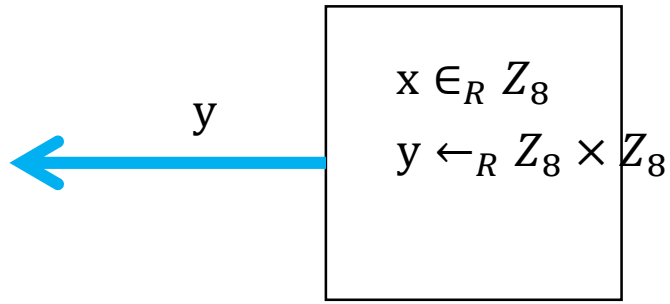
Play the role of the distinguisher

- Let $G : Z_8 \rightarrow Z_8 \times Z_8$ defined as $g(x) \coloneqq (\ 2^x \ mod\ 8\ , x^2\ mod\ 8)$

1) You the distinguisher must guess which of the two games produced the given values
   A. (0,1)
   B. (2,2)



G_0



G_1

Distinguisher

- Answer
  - $(1,1)\ -> G_0$ is your best guess
    - g(3) = (2^3 mod 8,3^2 mod 8) = (1,1)
      - $PR[G_0 = (1,1)] = \frac{1}{8}$
      - $\Pr[G_1 = (1,1,)]$ = 1/64
  - $(2,2) -> Only\ G_1$ can output (2,2).
    - $PR[G_0 = (2,2)] = 0$
    - $\Pr[G_1 = (1,1,)]$ = 1/64

Problem #1

$$
\begin{array}{r}
1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\
\oplus\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1 \\
\hline
1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0
\end{array}
$$

Problem #2

- Known plaintext, ciphertext
  - we attack in a week -> 16 letters
  - DWFTABWDUMAWDXS -> 16 letters
- attack
  - FMAKFINQNFMWQH -> 14
  - We attack by land -> 14
  - We attack by sea -> 13

Problem #3

- Only the size of the message space matters
- Consider $M = \{yes, no\}$
- You can choose keyspace {0,1}
- $Enc(0, yes) = yes$
- $Enc(0, no) = no$
- $Enc(1, yes) = no$
- $Enc(1, no) = yes$

Problem 4

- p = 23
- m = 11
- k = (12,5)

- $mac\big((12{,}5), 11\big) = 11 \cdot 12 + 5 \ (mod\ 23) = 22$

Problem 5

- p = 23
- m = (11,4)
- k = (14,2)

- $mac\big((14,2),(11,4)\big) = 11 \cdot 14 + 4 \cdot 14^2 + 2 \ (mod\ 23) = 20$

Problem 6

- p = 32
- m = 0
- t = 14

- $(m', t') \leftarrow (16,14)$
- $mac\big((k_1, k_2), 0\big) = k_2 \ \Rightarrow k_2 = 14$
- m$ac\big((k_1, 14), 16\big) = 16 \cdot k_1 + 14 \ (mod\ 32)$
- If k_1 is even then $16 \cdot k_1 \ (mod\ 32) = 0 \ \Rightarrow mac\big((k_1, 14), 16\big) = 14$
- $\Pr[k_1 \ \ is\ even] = \frac{1}{2}$
- Forgery accepted with probability ½ $> \frac{1}{32}$

Problem 6

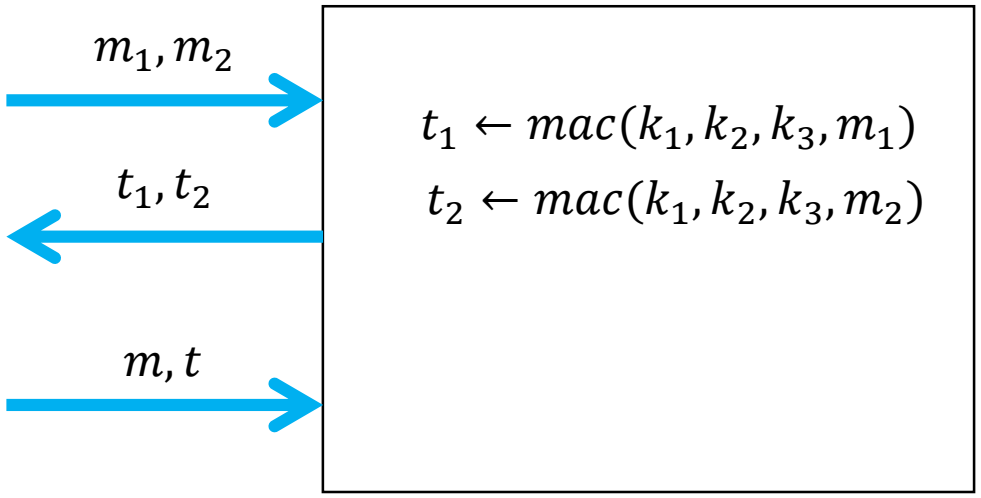
- p = 32
- m = 0
- t = 14

- $(m', t') \leftarrow (16,14)$
- $mac\big((k_1, k_2), 0\big) = k_2 \ \Rightarrow k_2 = 14$
- m$ac\big((k_1, 14), 16\big) = 16 \cdot k_1 + 14 \ (mod\ 32)$
- If k_1 is even then $16 \cdot k_1 \ (mod\ 32) = 0 \ \Rightarrow mac\big((k_1, 14), 16\big) = 14$
- $\Pr[k_1 \ \ is\ even] = \frac{1}{2}$
- Forgery accepted with probability ½ $> \frac{1}{32}$

Problem 7

- Evan is insecure
  - Let $m = 0$
    - $mac_1(k_1, k_2, k_3, 0) = k_2$
    - $mac_2(k_1, k_2, k_3, 0) = k_2$
    - $mac_1(k_1, k_2, k_3, 0) = mac_2(k_1, k_2, k_3, 0)$

Problem 7 : one definition of two-time mac



Win if $(m \neq m_1$ and $m \neq m_2) \wedge ((mac_1(k_1, k_2, k_3, m) = t) \vee (mac_2(k_1, k_2, k_3, m) = t))$

Problem seven : security

- Dave is secure because k_1 is always hidden: first by k_2 and then k_3

Problem 8: Extending the two-time mac

- $mac_i(m, k_1, \ldots, k_{n+1}) \coloneqq m \cdot k_1 + k_{i+1}$

Relationship between enigma and 2.7

- Excluding possibilities never helps

Book problem 1.5

- Shift: one letter
- Substitution: some subset of the letters (wheel of fortune)
- Vignere: as long as the key

Book problem 1.5

- Shift: one letter
- Substitution: some subset of the letters (wheel of fortune)
- Vignere: as long as the key

Book problem 2.3

- Refute

  Keygen

  - $b_1, b_2 \in_R \{0,1\}$
  - $k' \in_R \{0,1\}^n$

  $Enc((b_1, b_2, k), m)$

  $(b_1 \wedge b_2, k \oplus m)$

  Dec(b_1, b_2, k),(d,c))

  $\mathrm{m} \leftarrow c \oplus m$

Book problem 2.6

A. No

B. Yes

Book problem 2.7

- NOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOO

Book problem 2.10

- For every key and ciphertext there is at most 1 plaintext which the ciphertext can decrypt too
- Since |keyspace| < |message space|, for every ciphertext there must be at least one message which cannot be encrypted to that ciphertext
- We are using pigeon-hole principle

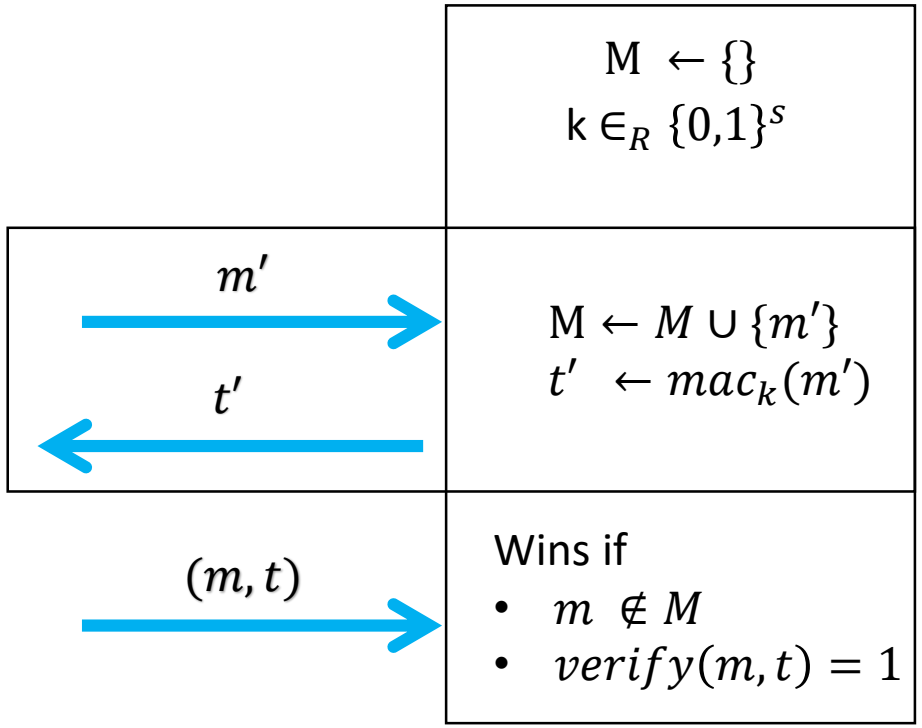
Book problem 2.13

- The same pair of ciphertexts must encrypt to the same plaintext, we get a contradiction by setting $c_1 = c_2, \mathrm{m}_1 \neq \mathrm{m}_2$

- Use $\mathrm{m}^2 \cdot k_1 + m \cdot k_2 + k_3 \ (mod\ p)$

Computational message authentication code

Mac forgery game

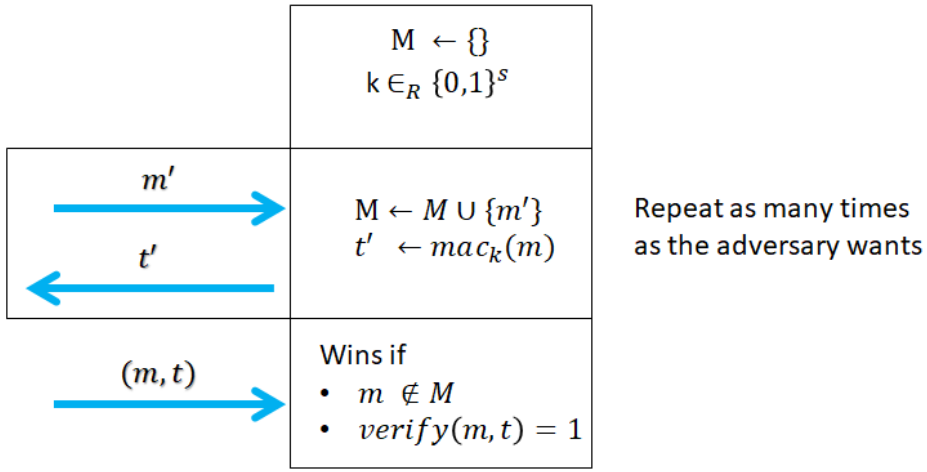


Repeat as many times
as the adversary wants

Mac forgery game

- Allow the adversary to learn tags for as many message as he wants

- A mac scheme is secure if
  - $\Pr[adv\ wins\ the\ forgery\ game] \leq negl(s)$

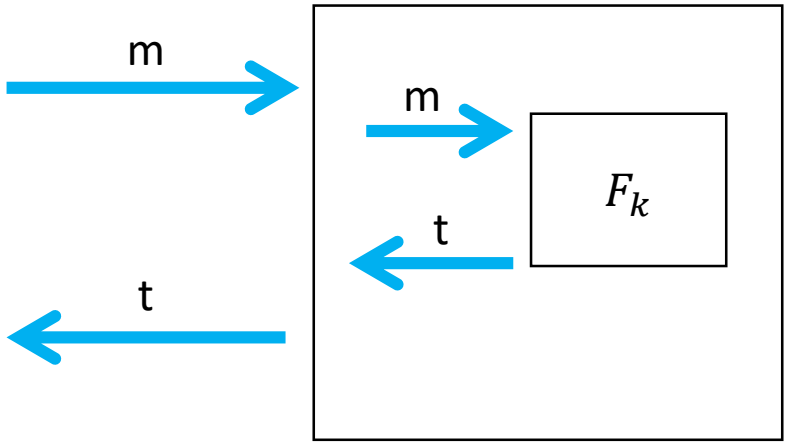


Does encryption imply authentication

- Let’s take as example the one-time pad
- What happens if the adversary flips a bit of the ciphertext
- Lesson**: Encryption does not imply authentication**

Fixed-length mac from PRF

- Keygen
  - $k \in_R \{0,1\}^n$
- Authenticate



Pitfalls of authenticating arbitrary length message

- $\widetilde{auth}_k(m_1, \ldots, m_n) \coloneqq auth(m_1), \ldots, auth(m_n)$

Pitfalls of authenticating arbitrary length message

- $\widetilde{auth}_k(m_1, \ldots, m_n) \coloneqq auth(m_1), \ldots, auth(m_n)$

- $\widetilde{auth}_k(m_1, \ldots, m_n) \coloneqq auth(1, m_1), \ldots, auth(n, m_n)$

Pitfalls of authenticating arbitrary length message

- $\widetilde{auth}_k(m_1, \ldots, m_n) \coloneqq auth(m_1), \ldots, auth(m_n)$
  - Block replacement attack

- $\widetilde{auth}_k(m_1, \ldots, m_n) \coloneqq auth(1, m_1), \ldots, auth(n, m_n)$
  - Different message attack
  - truncation

Arbitrary-length mac scheme

Based on secure fixed length authentication scheme (has secret key)

We use such a scheme in a black-box way.

$Auth(m)$

1. $m_1, \dots, m_d \leftarrow m$ (decompose m into d blocks)
2. $r \in_R \{0,1\}^s$
3. $w_{len} \leftarrow (r, 0, d)$
4. $w_i \leftarrow (r, 1, i, m_i)$
5. $t_{len} \leftarrow auth_k(w_{len})$
6. $t_i := auth_k(w_i)$
7. $t_{msg} \leftarrow t_1, \dots, t_d$
8. $t \leftarrow (r, t_{\text{len}}, t_{msg})$

- Disavantage: requires communicating one mac tag per block

CBC-mac (fixed-length extension)

$Auth(m)$

1. $m_1, \ldots, m_d \leftarrow m$
2. $t_0 \leftarrow 0^n$
3. For $i = 1, \ldots, d$
   a. $t_i \leftarrow F_k(t_{i-1} \oplus m_i)$
4. Output t_d

Computational mac from PRF and one-time mac

- otm is the one-time mac

- $keygen(\{1\}^s)$
  - $k \in_R \{0,1\}^s$

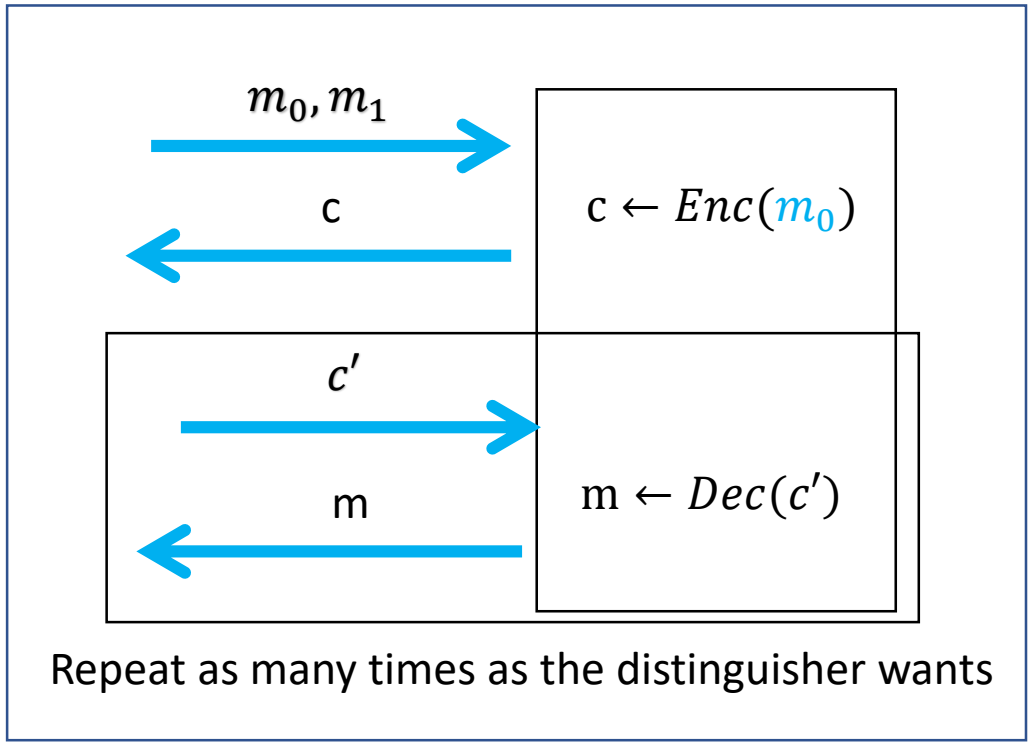
- $auth_k(m; r_1, r_2)$
  - $r_1, r_2 \in_R \{0,1\}^n$
  - $k_1 \leftarrow F_k(r_1)$
  - $k_2 \leftarrow F_k(r_2)$
  - $\bar{k} \leftarrow (k_1, k_2)$
  - t $\leftarrow (r_1, r_2, otm_{\bar{k}}(m))$

Authenticated encryption

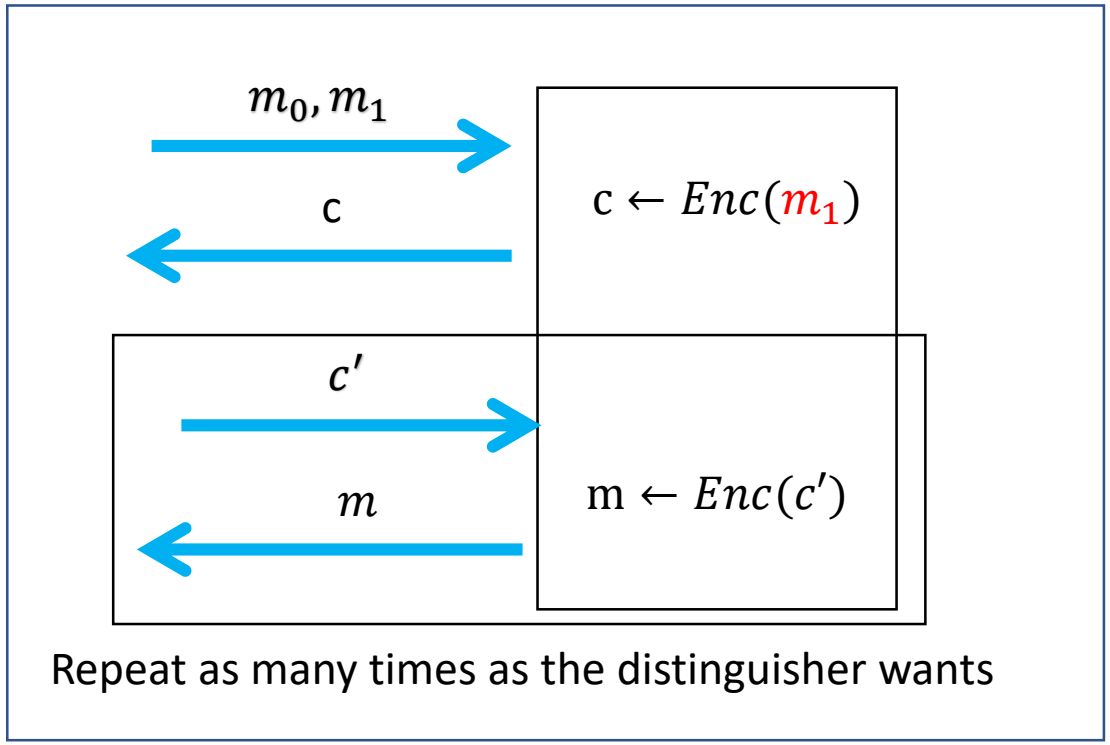
- Authenticated encryption
  - Authenticated (adversary cannot forge a ciphertext)
  - Encrypted (adversary cannot learn message)

Chosen-ciphertext game

Distinguisher loses automatically if $c = c'$

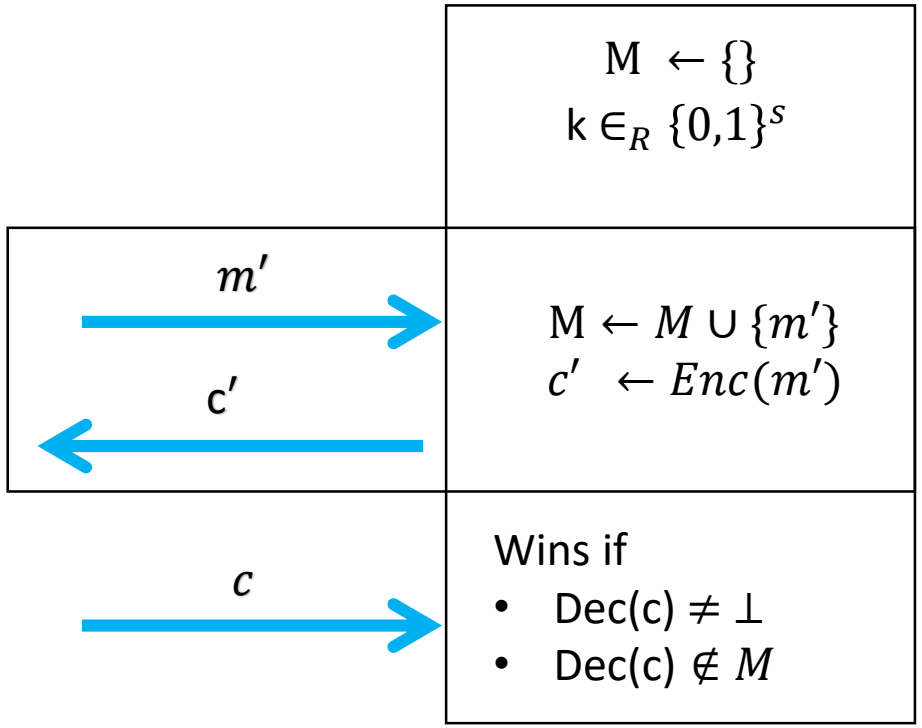


G_0



G_1

Unforgeability game



Authenticated encryption

- An Encryption scheme (Gen, Enc, Dec) is an authenticated encryption scheme if
  - Unforgeable
  - CCA-secure

Three Candidates for AE from mac + enc

- We assume
  - $\overline{Enc}$ is a secure encryption scheme
  - $\overline{Auth}$ is a secure message authentication code

Encrypt-and-mac	encrypt-then-mac	Mac-then-encrypt
$Enc(m)$ $c' \leftarrow \overline{Enc}(m)$ $t \leftarrow \overline{Auth}$ (m) $c \leftarrow (c', t)$	$Enc(m)$ $c' \leftarrow \overline{Enc}(m)$ $t \leftarrow \overline{Auth}$ (c') $c \leftarrow (c', t)$	$Enc(m)$ $t \leftarrow \overline{Auth}$ (m) $w \leftarrow (m, t)$ $c \leftarrow \overline{Enc}(w)$

Authenticated encryption with associated (public) data

- $Enc(data, m) \rightarrow (data, c, t)$
- Correctness:
  - $Dec(data, Enc(data, m)) = (data, m)$
- Authentication
  - Impossible to create a fresh pair $(data', c', t')$ such that:
    - $(data, c')$ has been seen before
- Indistinguishability

Galois-counter mode

- $Enc(data, m)$
  - $m_1, \dots, m_d \leftarrow m$
  - $A_1, \dots, A_\ell \leftarrow data$
  - $H \leftarrow E_k(IV, 0)$
  - $c_1, \dots, c_d \leftarrow CTR(m_1, \dots, m_n ; counter = (IV \,||\, 1))$
  - $\tau_0 \leftarrow 0$
  - $$\tau_i \leftarrow \begin{cases} (\tau_{i-1} \oplus A_i) \cdot H, & i \in \{1, d\} \\ (\tau_{i-1} \oplus C_i) \cdot H, & i \in \{d+1, d+\ell\} \\ (\tau_{i-1} \oplus (d \,||\, \ell)) \cdot H, & i = d + \ell + 1 \end{cases}$$
  - $\mathrm{c} \leftarrow (d, \ell, data, c_1, \dots, c_d, \tau_{d+\ell+1})$