Authenticated Encryption

Contents

- Ciphertext integrity
- AE definitions
- Chosen Ciphertext Attack
- Constructions
  - Encrypt-then-MAC
  - Encrypt-and-MAC
  - MAC-then-Encrypt

Authenticated Encryption (AE)

- Everything demonstrated so far provides
  - either integrity
  - or confidentiality (security against eavesdropping)
- CPA security does not provide secrecy against active attacks (where an attacker can tamper with ciphertext)

➔ If you require integrity → **MAC**

➔ If you require integrity and confidentiality → **AE**

AE: Desired properties

- An authenticated encryption system $\zeta = (E, D)$ is a cipher where

  as usual $E : K \times M \times N \to C$

  but $D : K \times C \times N \to M \cup \{\perp\}$ $\quad \perp \notin M$

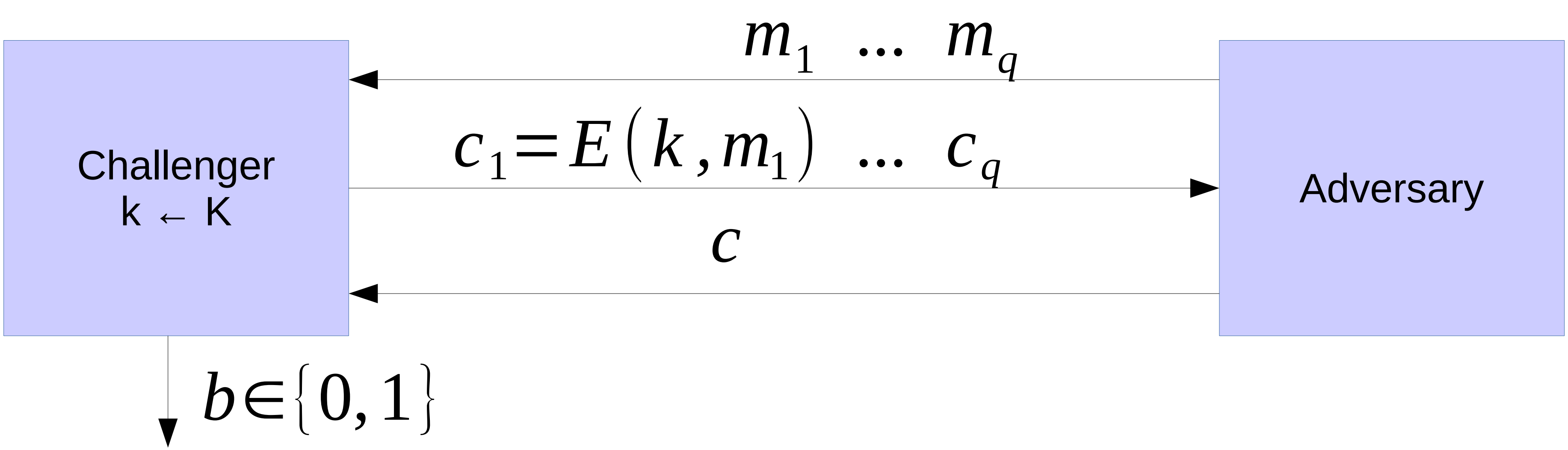
  Nonce

  CT is invalid (rejected)

- Security: the system must provide
  - **semantic security under CPA**, and
  - **ciphertext integrity**
    - an adversary cannot create a new valid CT (such that would decrypt properly)

Ciphertext integrity (def)

Let $\zeta = (E, D)$ be a cipher with message space M



$$b = 1 \quad \text{if } D(k, c) \neq \perp \text{ and } c \notin \{c_1 \dots c_q\}$$
$$b = 0 \quad \text{otherwise}$$

Def: $\zeta = (E, D)$ has **ciphertext integrity** if for all “efficient” adversaries A: $\mathrm{Adv}_{\mathrm{CI}}[A, \zeta]$ is “negligible”.

$$\mathrm{Adv}_{\mathrm{CI}}[A, \zeta] = \Pr[\text{Chal. outputs } 1]$$

Authenticated Encryption

- Def: A cipher $\zeta = (E, D)$ **provides authenticated encryption (AE)** if it is
  1) semantically secure under CPA, and
  2) has ciphertext integrity.

- Do the following ciphers provide AE:
  - AES-CBC,
  - AES-CTR,
  - RC4?
- Why?

Authenticated Encryption

- Implication 1: Authenticity

  - An attacker cannot create a new valid $c \notin \{c_1 \ldots c_q\}$
  - If message decrypts properly $(D(k, c) \neq \perp)$, it must have come from someone who knows secret key *k*
    - But it could be a replay

- Implication 2: Security against **chosen ciphertext attack (CCA)**

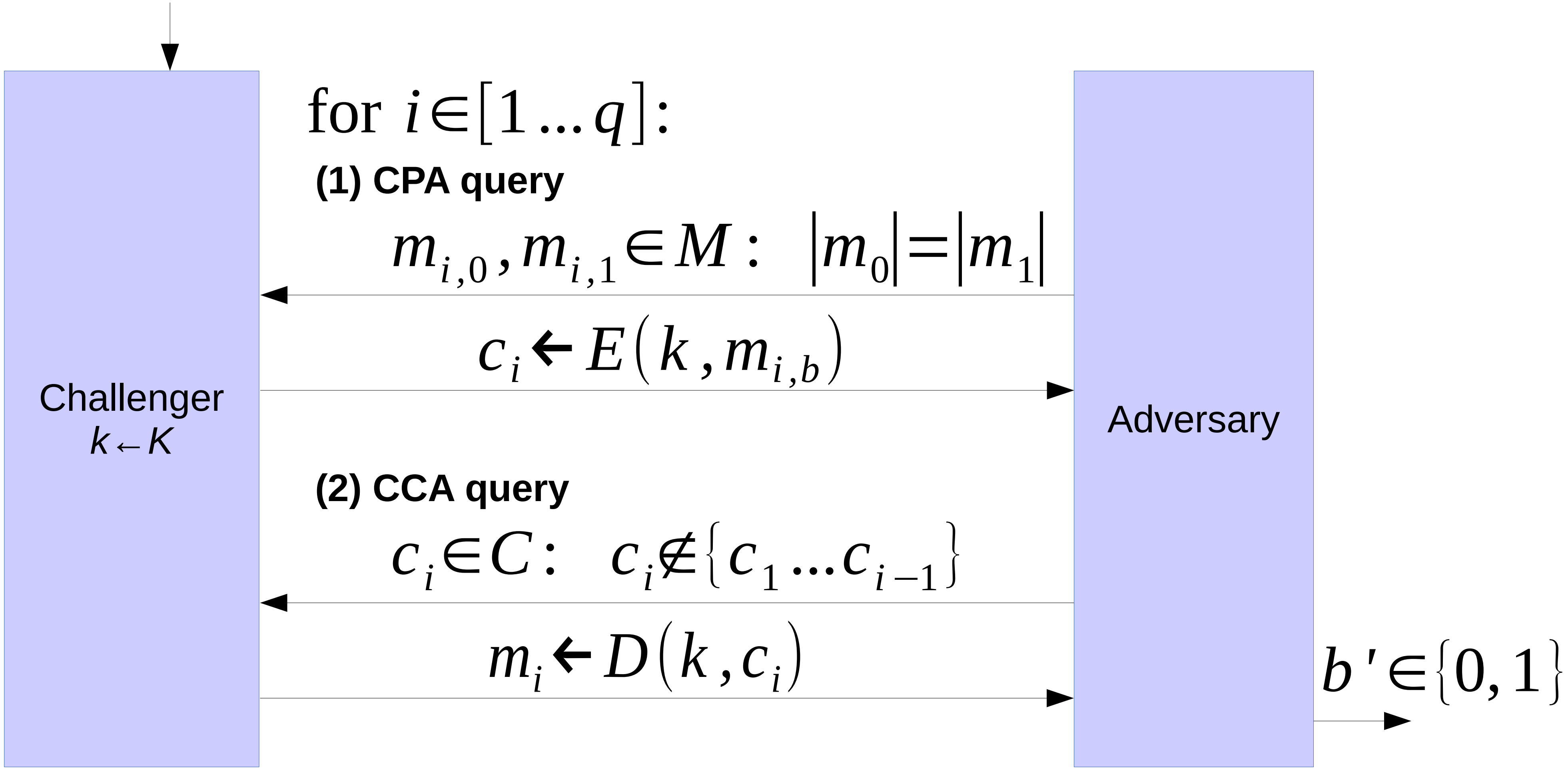
Adaptation of: Dan Boneh, Cryptography I, Stanford.

Chosen ciphertext security

- Adversary’s power: **CPA** and **CCA**
  - Can encrypt any message of her choice
  - Can decrypt any message of her choice *other than some challenge*
  - (still conservative modeling of real life)
- Adversary’s goal: **break semantic security**
  - Learn about the PT from the CT

Chosen ciphertext security (def)

- Let $\zeta=(E,D)$ be a cipher defined over (K,M,C)
- For $b\in\{0,1\}$ define experiments EXP(b) as



Chosen ciphertext security (def)

- Def. Cipher $\zeta=(E,D)$ is CCA secure if for all efficient adversaries A $\mathrm{Adv}_{\mathrm{CCA}}[A,\zeta]$ is negligible.

  $$\mathrm{Adv}_{\mathrm{CCA}}[A,\zeta] := \left|\Pr[\mathrm{EXP}(0)=1]-\Pr[\mathrm{EXP}(1)=1]\right|$$

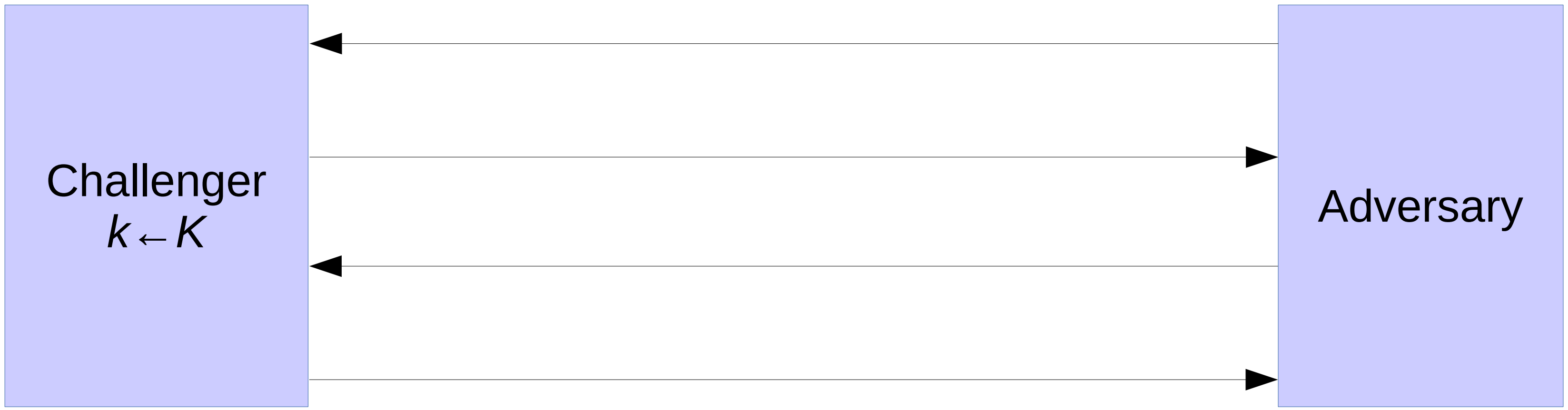
- Thm. A cipher that provides AE is also CCA secure.

- Implication. AE provides confidentiality against an active adversary that can decrypt some ciphertexts.

- Limitations
  - AE does not prevent replay attacks
  - Does not account for side channels attacks (timing)

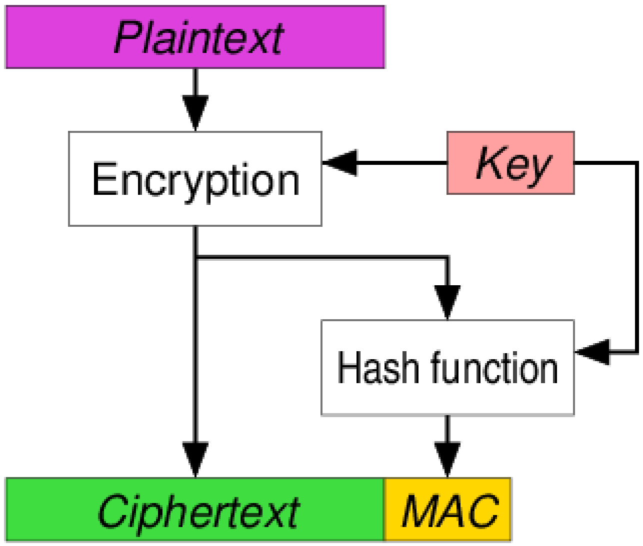
Ex: AES-CTR is not CCA secure

- Recall
  - AES-CTR is effectively a stream cipher
  - Malleability of stream ciphers



Encrypt then MAC

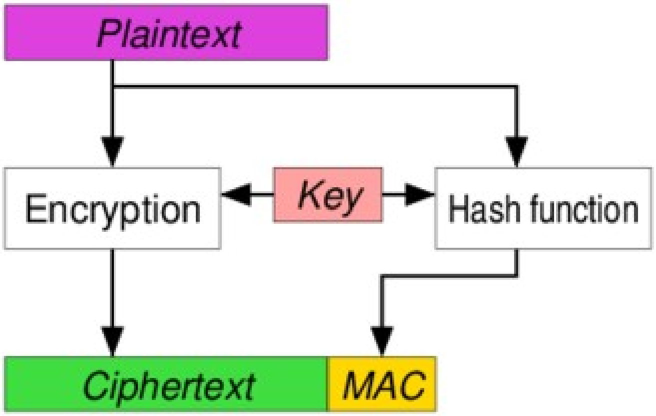
- MAC computed over cipher text
- Used in IPsec, always provides AE
  - Use separate and independent keys



https://en.wikipedia.org/wiki/Authenticated_encryption

Encrypt and MAC

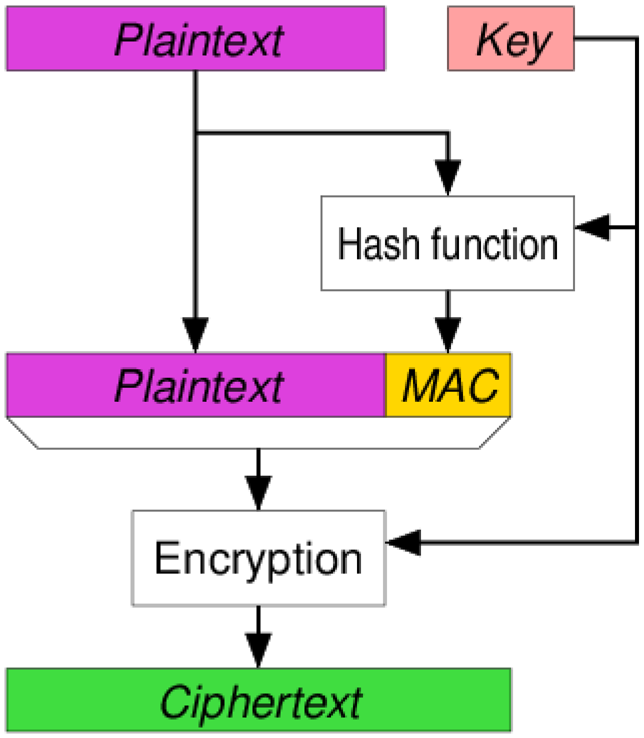
- MAC computed over plain text and sent unencrypted
- Used in SSH
- Use separate and independent keys



https://en.wikipedia.org/wiki/Authenticated_encryption

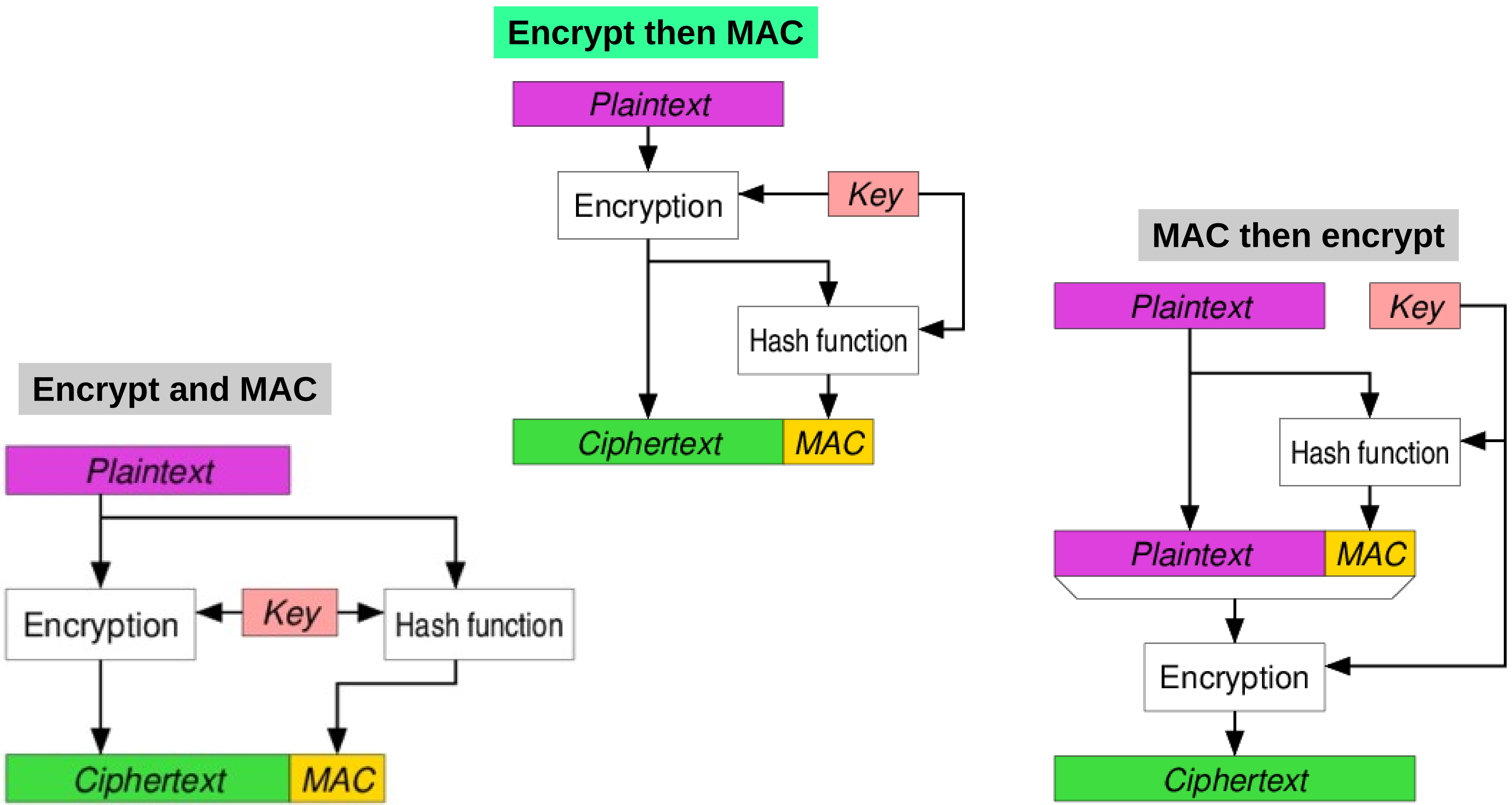
MAC then encrypt

- MAC computed over plain text and then encrypted before sending
- Used in TLS/SSL
- Use separate and independent keys



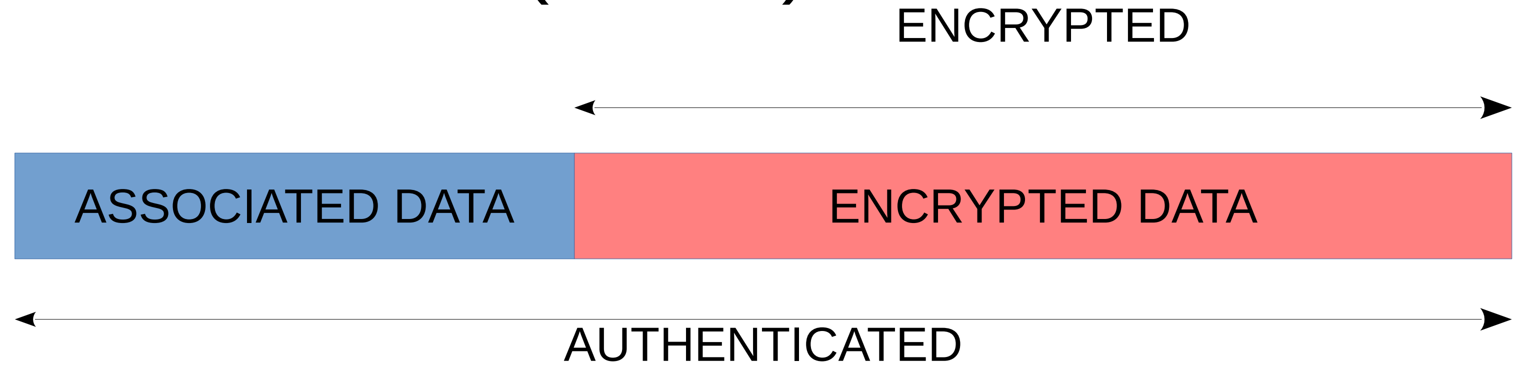
https://en.wikipedia.org/wiki/Authenticated_encryption

Three AE approaches



AE: Standardized solutions

- Galois/Counter Mode (GCM)
  - CTR mode encryption then CW-MAC
  - Made popular by Intel's PCLMULQDQ instruction
- CBC-MAC then CTR mode encryption (CCM)
- EAX
- All support ***authenticated encryption with associated data*** (AEAD)



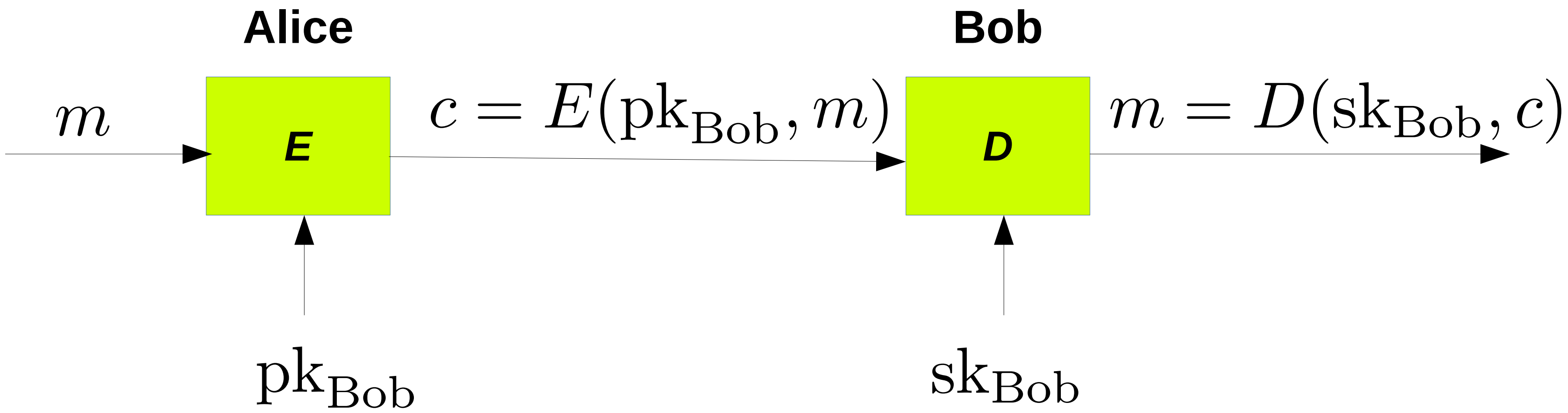
Public key encryption

Index

- Public-key ciphers overview
- Security definitions
  - CPA-security
  - CCA-security
- Trapdoor functions and permutations (TDF, TDP)
  - Encryption schemes from TDF (ISO)
- Example TDP: RSA
  - Definition
  - RSA in practice
  - Security of RSA

Public key encryption

- Each party uses a key pair: $k = (\mathrm{pk}, \mathrm{sk})$
- Public key is given to everyone, secret is kept hidden



Public key encryption: usage

- Communication session set-up
  - A process where Alice and Bob agree upon a shared secret
- Non-interactive applications
  - E.g. email
  - Typically, PKs are long-lived, symmetric keys are ephemeral
  - (But the sender needs to know recipient's PK in advance – need PKI)

Public key encryption: def

Def. A public-key encryption system is triple of algs. (G, E, D)

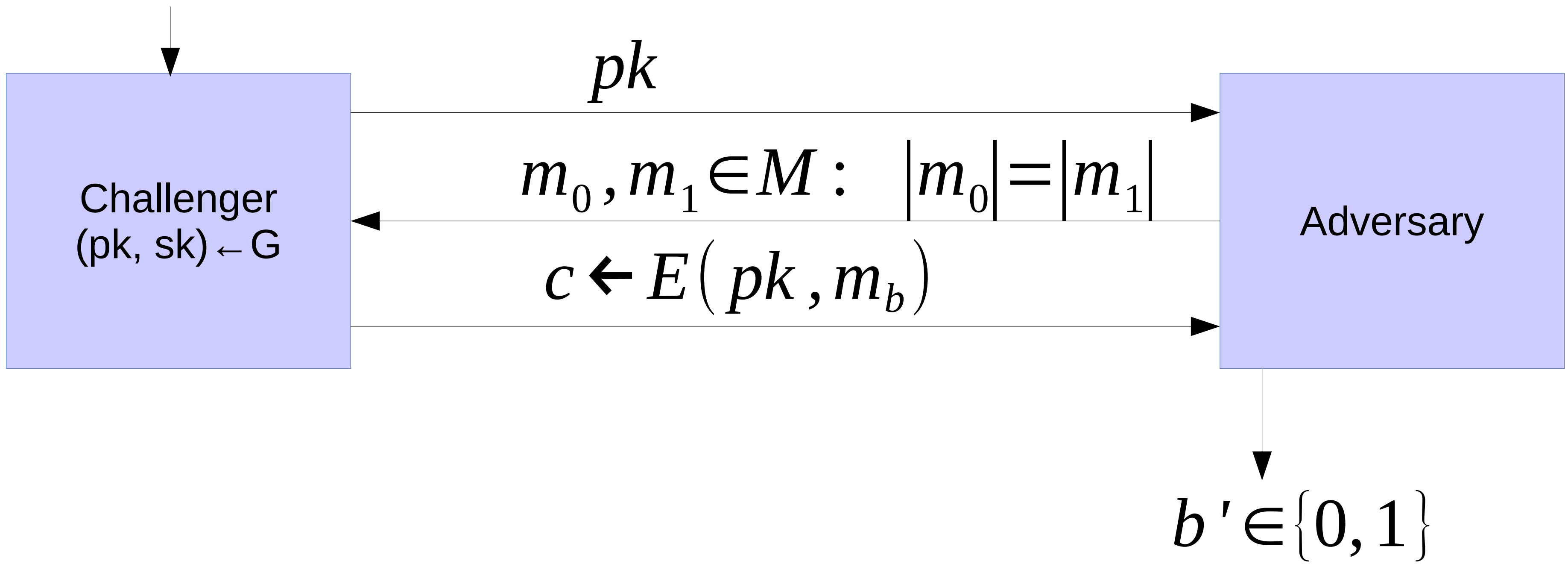
- $G()$ rand. alg. generates key pairs (pk, sk)
- $E(pk, m)$ <u>rand. alg.</u> takes $m \in M$ and returns $c \in C$
- $D(sk, c)$ det. alg. takes $c \in C$ and returns $m \in M$ or $\perp$

such that $\forall (pk, sk)$ output by G:

$$\forall m \in M : D(sk, E(pk, m)) = m$$

Semantic security (def)

Let $\zeta = (G, E, D)$ be a public key encryption system.
For $b \in \{0, 1\}$ define experiments EXP(0), EXP(1)



Def: $\zeta = (G, E, D)$ is **semantically secure** (aka IND-CPA) if for all eff. adversaries $A : \mathrm{Adv}_{SS}[A, \zeta]$ is negligible.

$$\mathrm{Adv}_{SS}[A, \zeta] := \left| \Pr[\mathrm{EXP}(0) = 1] - \Pr[\mathrm{EXP}(1) = 1] \right|$$

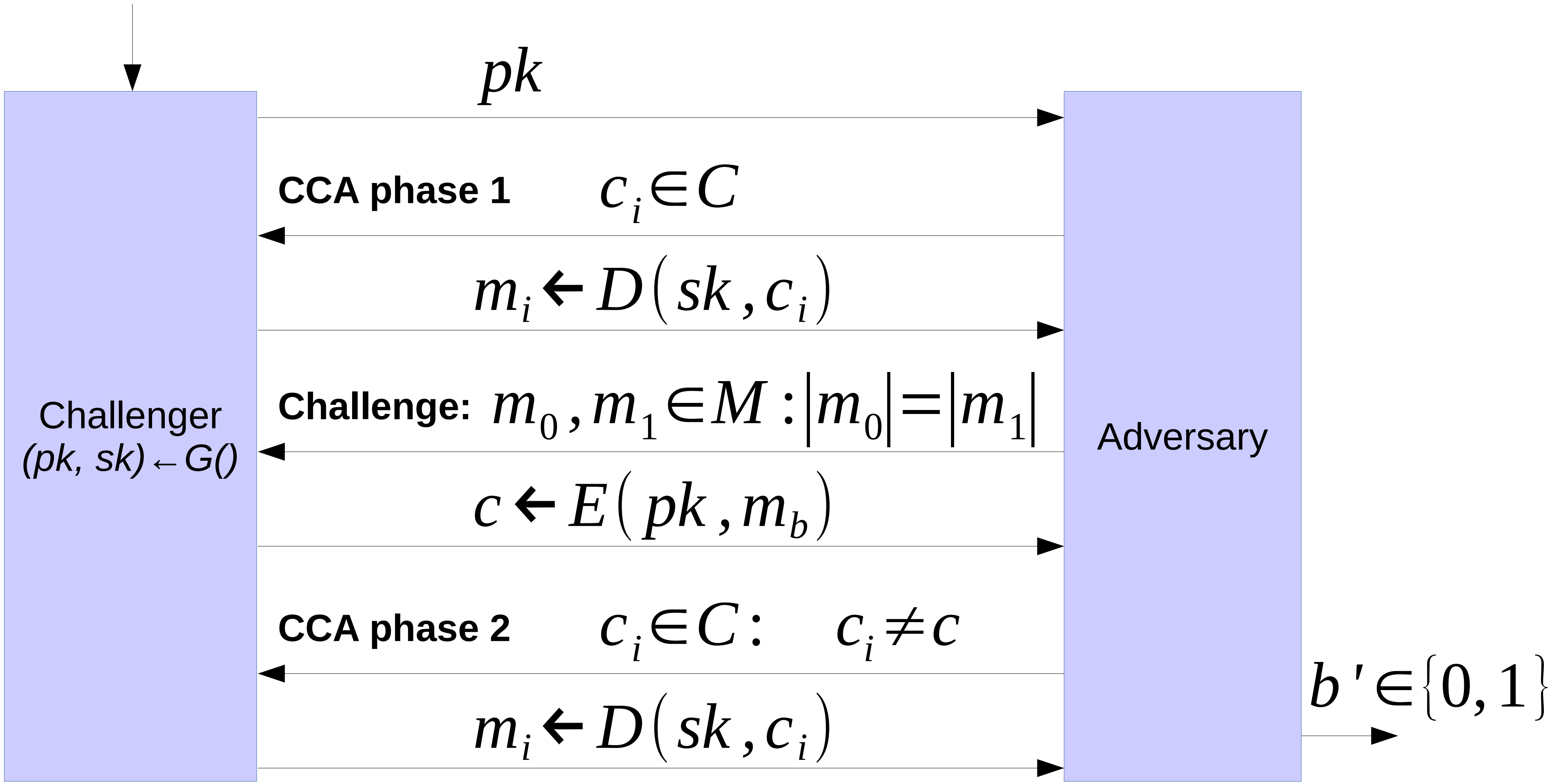
Adaptation of: Dan Boneh, Cryptography I, Stanford.

Relation to symmetric cipher security

- For symmetric ciphers, we had 2 security definitions
  - One-time security (key used only once) and many-time security (key used many times; CPA)
  - One-time security does not imply many-time security (OTP is broken if used more than once)
- Public key encryption
  - One-time security $\rightarrow$ many-time security (CPA)
    - Because the adversary can encrypt herself (she knows pk)
  - Public key encryption **must be randomized**

(pub-key) Chosen Ciphertext Security (def)

$\zeta = (G, E, D)$ a pub-key enc. over (M, C). For $b \in \{0,1\}$ define experiments EXP(b):



Adaptation of: Dan Boneh, Cryptography I, Stanford.

CCA security

- Def. $\zeta = (G, E, D)$ is CCA secure (aka. IND-CCA) if for all efficient adversaries A: $\mathrm{Adv}_{\mathrm{CCA}}[A, \zeta]$ is negligible.

$$\mathrm{Adv}_{\mathrm{CCA}}[A, \zeta] := \left|\Pr[\mathrm{EXP}(0)=1] - \Pr[\mathrm{EXP}(1)=1]\right|$$

- Recall: A secure symmetric cipher provides AE, when it has CPA security and ciphertext integrity
  - Attacker cannot create new ciphertexts (implies CCA security)
- In pub-key setting
  - Attacker knows pk $\rightarrow$ **can** create new ciphertexts
  - Instead: we directly require CCA security
- Next step: Constructing CCA secure pub-key encryption

Trapdoor function (TDF)

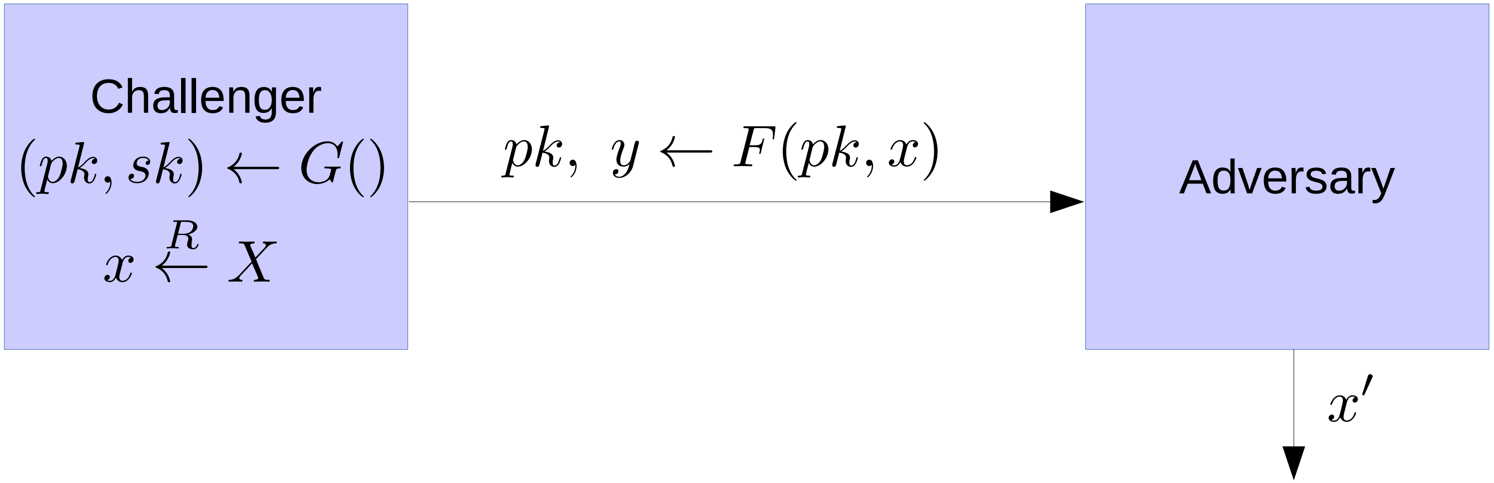
- **Def.** A trapdoor function $X \rightarrow Y$ is a triple of eff. algorithms (G, F, F^{-1})
  - **G()**: rand. alg. for creating (pk, sk)
  - **F(pk, -)**: <u>det. alg.</u> that defines $X \rightarrow Y$
  - **F^{-1}(sk, -)**: det. alg. that defines $Y \rightarrow X$ [inverts F(pk, -)]

For every (pk, sk) returned by **G**

$$F^{-1}[\ sk, F(pk, x)\ ] = x$$

Secure TDFs

- TDF (G, F, F^{-1}) is secure if F(pk, -) is *one-way*
  - It can be evaluated but not inverted without sk



- Def. (G, F, F^{-1}) is a secure TDF if for all eff. algs. A: $\mathrm{Adv}_{\mathrm{OW}}[A, F] := \Pr[x = x']$ is negligible.

Pub-key encryption from TDFs

(ISO 18033-2 standard)

- Building blocks
  - (G, F, F^{-1}) – secure TDF $X \to Y$
  - (E_S, D_S) – symmetric AE cipher over (K, M, C)
  - $H: X \to K$ – a hash function
- Pub-key enc. system **(G, E, D)**
  - Key generation **G**: same as **G** in TDF

E(pk, m):

$x \xleftarrow{R} X$, $\quad y \leftarrow F(pk, x)$

$k \leftarrow H(x)$, $\quad c \leftarrow E_s(k, m)$

return (y, c)

D(sk, (y, c)):

$x \leftarrow F^{-1}(sk, y)$

$k \leftarrow H(x)$, $\quad m \leftarrow D_s(k, c)$

return m

Pub-key encryption from TDFs
(ISO 18033-2 standard)

$F(pk, x)$	$E_S(H(x), m)$

Thm. If **(G, F, F^{-1})** is a secure TDF, if **(E_s, D_s)** provides AE, and if **H: X → K** is a "random oracle", then **(G, E, D)** is CCA^{ro} secure.

An incorrect use of TDF:

E(pk, m) := F(pk, m)

D(sk, c) := F^{-1}(sk, c)

Such construction results in a deterministic encryption scheme: cannot be semantically secure

Trapdoor permutation (TDP)

- TDP is a triple of eff. algorithms (G, F, F^{-1})
  - G(): generates (pk, sk); pk defines a function $X \rightarrow X$
  - F(pk, x): evaluates the function at x
  - F^{-1}(sk, y): inverts the function at y using sk

- **Secure** TDP

  The function F(pk, -) is one-way without the sk

Arithmetic modulo composites

Let $N = p \cdot q$ where p, q are primes

$$\mathbb{Z}_N = \{0, 1, ..., N-1\}$$

$$\mathbb{Z}_N^* = \{\text{invertible elements in } Z_N\}$$

Facts $\quad x \in \mathbb{Z}_N$ is invertible $\iff \gcd(x, N) = 1$

$$|\mathbb{Z}_N^*| = \varphi(N) = (p-1)(q-1) = N - p - q + 1$$

Euler's theorem

$$\forall x \in \mathbb{Z}_N^* : x^{\varphi(N)} = 1 \mod N$$

RSA trapdoor permutation

- G():
  - Choose random primes p, q (~1024 bits); $N = p \cdot q$
  - Choose integers e, d such that $e \cdot d = 1 \mod \varphi(N)$
  - Return $pk = (N, e),\ sk = (N, d)$
- F(pk, x): $\mathbb{Z}_N^* \to \mathbb{Z}_N^* : \mathrm{RSA}(x) = x^e \mod N$
- F^{-1}(sk, y):

$$
\begin{aligned}
y^d &= \mathrm{RSA}(x)^d && \mod N \\
&= x^{ed} && \mod N \\
&= x^{k \cdot \varphi(N) + 1} && \mod N \\
&= \left(x^{\varphi(N)}\right)^k \cdot x && \mod N \\
&= x
\end{aligned}
$$

RSA trapdoor permutation

RSA assumption: RSA is one-way permutation

For all eff. algs. A:

$$\Pr\left[A(N,e,y) = \sqrt[e]{y}\right] < \text{negligible}$$

$$p, q \leftarrow n\text{-bit primes}$$

$$N = p \cdot q$$

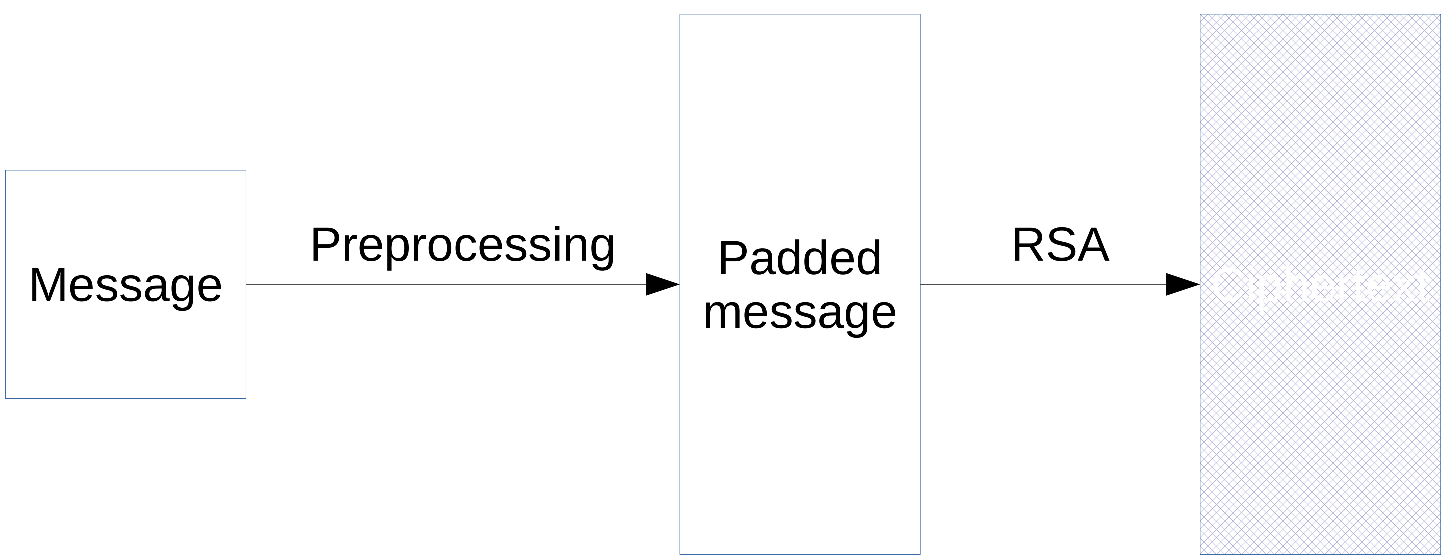
$$y \xleftarrow{R} \mathbb{Z}_N^*$$

Insecure “textbook” RSA

- Encrypting directly with RSA (“textbook” RSA) is insecure
  - $E((N,e),x) := x^e \mod N$
  - $D((N,d),y) := y^d \mod N$
- Problem 1: Ciphertext is **malleable**
  - Given ciphertext $c = E((N,e),m)$ an attacker can create $c' = c \cdot 2^e \mod N$
  - The modified ciphertext c' decrypts to $2m \mod N$
- Problem 2: Encryption is **deterministic**

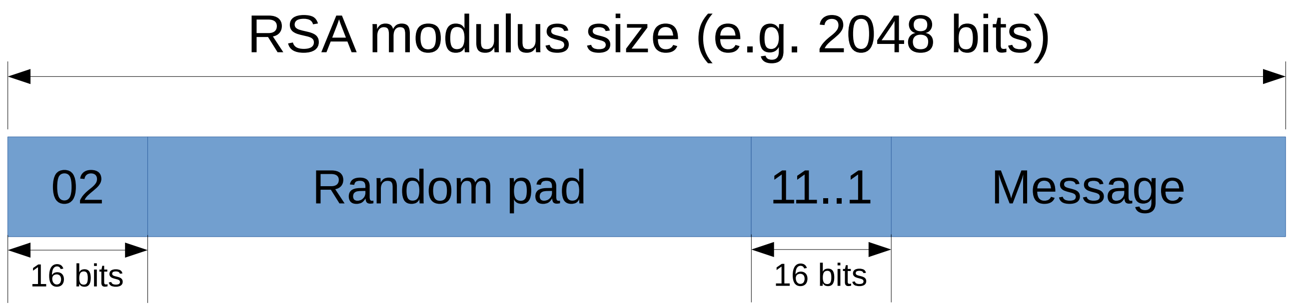
RSA in practice

- RSA in practice (ISO standard rarely used)
  - Expand the message to the RSA modulus size and add random bits
  - Apply the RSA function



Adaptation of: Dan Boneh, Cryptography I, Stanford.

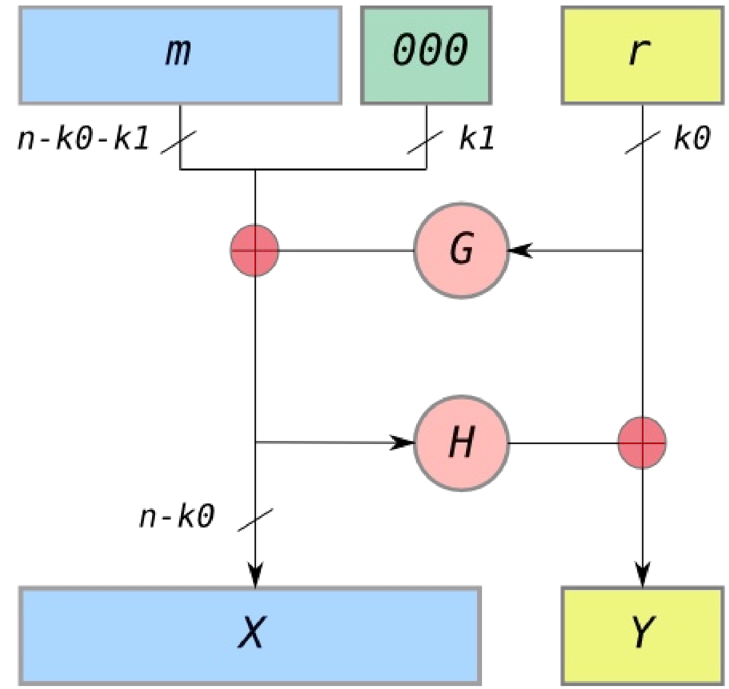
RSA in practice: PKCS1 v1.5



- Resulting value is RSA encrypted
- Widely deployed (HTTPS)
- Attack due to Bleichenbacher (1998)
  - During decryption, the system will signal an error if the decrypted plaintext does not start with 02
  - Enough to completely decrypt the ciphertext
- Solution in RFC 5246
  - set decrypted PT to a random value and *fail later on*
- Generally PKCS1 v1.5 padding should be avoided

RSA in practice: PKCS1: v2.0 (OAEP)

- New preprocessing function: **Optimal asymmetric encryption padding (OAEP)**
- Check pad on decryption
  - Reject CT if invalid
- **Thm.** If RSA is a TDP, then RSA-OAEP is CCA secure if H, G are *random oracles*.
  - In practice we use SHA-256 for H and G



https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

RSA security (informally)

- To invert RSA one-way function, the attacker must extract x from $c = x^e \mod N$
- How difficult is to compute e’th root modulo N? Currently best known algorithm
  - Step 1: Factor N [difficult]
  - Step 2: Compute e’th roots modulo p and q [easy]
- Shor’s algorithm: a quantum algorithm for integer factorization in polynomial time
  - Unknown if quantum computers can be built

RSA security (informally)

- Security of public key system should be comparable to security of symmetric cipher

Cipher key size	**RSA modulus size [in modulo primes]**
80	1024
128	3072
256	15360