Integrity

Contents

- Introduction
- MAC Definition
  - PRF
  - Secure PRF → Secure MAC
- ECBC-MAC
- Cryptographic hash functions
  - Collision resistance
  - MACs from CR
  - Merkle-Damgard iterative construction
- HMAC

Introduction

- Integrity: maintaining accuracy and completeness of data
- Goal
  - Prevent adversary from modifying data
  - More feasible: detect if data has been altered
- Examples
  - Protecting files on disks
  - Assuring installation of correct software
  - Assuring the delivered packet has not been tempered with in traffic

Message Authentication Code

Compute TAG $t = S(k, m)$

Verify TAG $1 = V(k, m, t)$

$MAC\ I = (S, V)$ defined over (K, M, T) is a pair of algs.:

$$S : K \times M \rightarrow T$$

$$V : K \times M \times T \rightarrow \{0,1\}$$

$$|M| \gg |T|$$

such that

$$\forall k \in K, m \in M: \ V(k, m, S(k, m)) = 1$$

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Is a shared secret required?

- Is all these secrecy required?
- Could we not just simply use
  - MD-5 or
  - SHA-{1,2,3} or
  - CRC?

Secure MAC

- Attacker's power: **Chosen message attack**
  - For $m_1 \ldots m_q$ attacker is given $t_i = S(k, m_i)$
- Attacker's goal: **Existential forgery**
  - Produce a **new** valid (m, t) s. t.

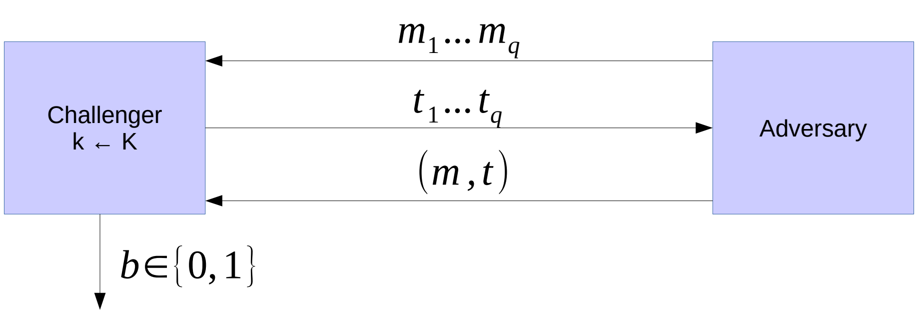
$$(m,t) \notin \{(m_{1,} t_1) \ldots (m_q, t_q)\}$$

Implications

→ attacker cannot produce a valid tag for a new message

→ given (m,t) attacker cannot produce (m, t') for $t \neq t'$

Secure MAC (def)



$$b = 1 \quad \text{if } V(k, m, t) = 1 \text{ and } (m,t) \notin \{(m_{1,} t_1) \ldots (m_q, t_q)\}$$
$$b = 0 \quad \text{otherwise}$$

$I = (S, V)$ is a **secure MAC** if for all "efficient" adversaries A

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

is "negligible".

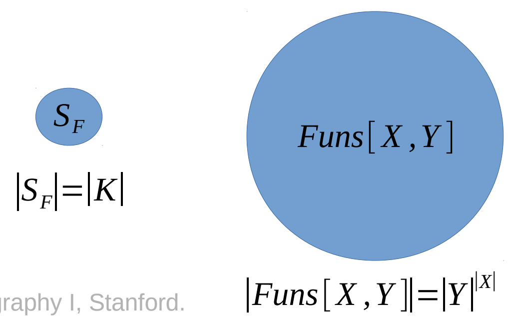
Secure MAC

- Negligible?
  - Assume less than 2^{-80}
- Suppose a *S(k, m)* computes 10-bit tags
  - Is such a MAC secure, why?

(Recall) Secure PRF

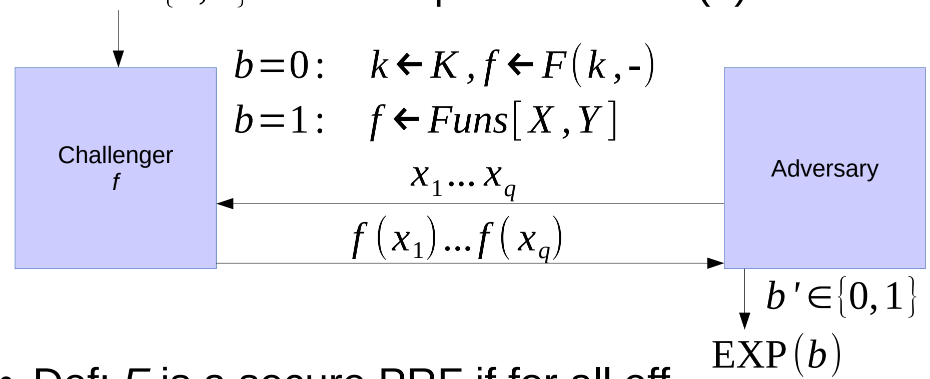
- Let $F : K \times X \to Y$ be a PRF
  - $Funs[X, Y]$ the set of all functions from X to Y
  - $S_F = \{F(k, -) : \forall k \in K\} \quad \subseteq Funs[X, Y]$

  Intuitively
  - A PRF is secure if a random function in $Funs[X, Y]$ is indistinguishable from a random function in S_F



(Recall) Secure PRF (def.)

- For $b \in \{0, 1\}$ define experiment EXP(b) as



- Def: *F* is a secure PRF if for all eff. adversaries A $\text{Adv}_{\text{PRF}}[A, F]$ is negligible.

$$\text{Adv}_{\text{PRF}}[A, F] := \left|\Pr[\text{EXP}(0) = 1] - \Pr[\text{EXP}(1) = 1]\right|$$

Secure PRF → Secure MAC

- For a PRF $F : K \times X \to Y$ define MAC $I_F = (S, V)$

$$S(k, m) := F(k, m)$$

$$V(k, m, t) := \begin{cases} 1 & t = F(k, m) \\ 0 & \text{otherwise} \end{cases}$$

- **Thm**. If F is a secure PRF and $1/|Y|$ is negligible (i.e. $|Y|$ is sufficiently large), then I_F is a secure MAC.

Truncating MACs based on PRFs

- Lemma: Suppose $F : K \times X \to \{0, 1\}^n$ is a secure PRF. So is $F_t(k, m) := F(k, m)[1 \dots t]$ for all $1 \le t \le n$

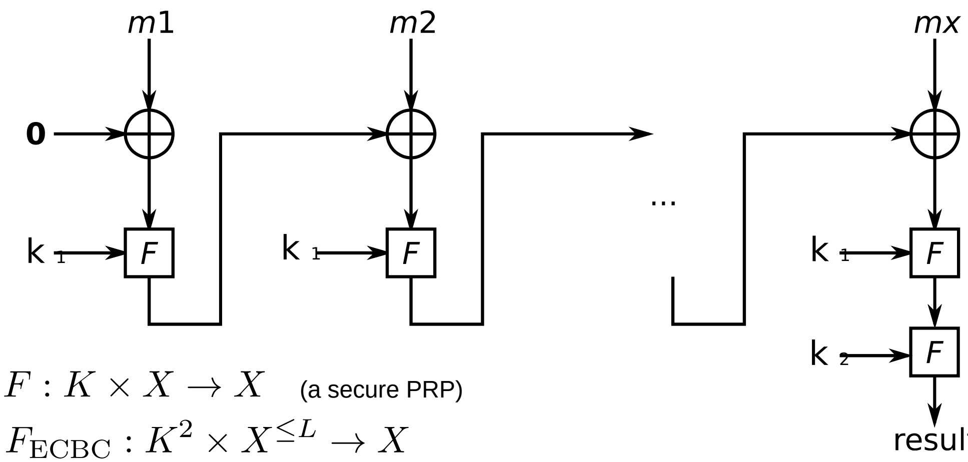
- If (S, V) is a MAC based on a secure PRF that outputs *n*-bit tags, then the truncated MAC that outputs *w* bits is also secure.
  - As long as 2^{-w} is still negligible

Examples of secure MAC

- AES (or any secure PRF)
  - A secure MAC for 16-byte (128-bit) messages
- Longer messages?
  - **CBC-MAC**
  - **HMAC**
- Both convert a small-PRF into a big-PRF

Adaptation of: Dan Boneh, Cryptography I, Stanford.

ECBC-MAC



$F : K \times X \to X$ (a secure PRP)

$F_{\mathrm{ECBC}} : K^2 \times X^{\leq L} \to X$

$$X^{\leq L} = \bigcup_{i=1}^{L} X^i$$

https://en.wikipedia.org/wiki/CBC-MAC

Hash-MAC (HMAC)

- Built from *collision resistance*
- Let $H : M \to T$ be a hash function $|M| \gg |T|$
- A **collision** for H is a pair $m_0, m_1 \in M$ such that:
  $$H(m_0) = H(m_1) \text{ and } m_0 \neq m_1$$
- Function H is **collision resistant** if for all *explicit* "eff." algs. A $\mathrm{Adv}_{\mathrm{CR}}[A, H]$ is negligible.

$$\mathrm{Adv}_{\mathrm{CR}}[A, H] := \Pr[A \text{ outputs collision for } H]$$

- Example: SHA-256

Adaptation of: Dan Boneh, Cryptography I, Stanford.

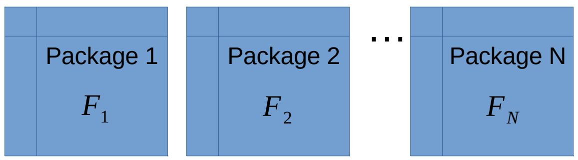
MAC from CR

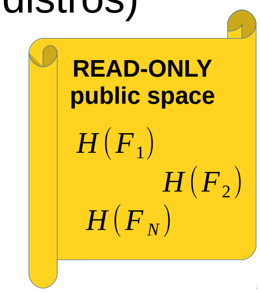
- Let $I = (S, V)$ be a MAC for short messages over (K, M, T) (e.g. AES)
- Let $H : M^{\mathrm{BIG}} \to M$
- Def: $I^{\mathrm{BIG}} = (S^{\mathrm{BIG}}, V^{\mathrm{BIG}})$ over (K, M^{BIG}, T) as:
  $$S^{\mathrm{BIG}}(k, m) := S(k, H(m))$$
  $$V^{\mathrm{BIG}}(k, m, t) := V(k, H(m), t)$$
- **Thm**. If I is a secure MAC and H is collision resistant, then I^{BIG} is a secure MAC.
- Example: $S(k, m) := \mathrm{AES}_{\text{2-block-CBC}}(k, \text{SHA-256}(m))$

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Example: Integrity using CR hash

- Protecting software packages (Linux distros)





- User downloads a package and verifies it using hashes in public space
  - If *H* is collision resistant, the attacker cannot modify packages without being detected
- We require <u>no shared secret</u>, but we need a <u>read-only public space</u>

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Generic attack on CR

- Let $H: M \rightarrow \{0,1\}^n$ be a hash function $|M| \gg 2^n$
- Generic algorithm to find a collision
  1)Chose $\sqrt{2^n} = 2^{\frac{n}{2}}$ random messages: $m_1 \ldots m_{2^{n/2}} \in M$ distinct w.h.p.
  2)For $i = 1 \ldots 2^{n/2}$: compute $t_i = H(m_i)$
  3)Look for a collision $(t_i = t_j)$. If not found, go to 1.

- How many iterations before we find a collision?

Adaptation of: Dan Boneh, Cryptography I, Stanford.

The birthday paradox

- **Thm.** Let $r_1 \ldots r_n \in [1 \ldots B]$ be independent and identically distributed integers. If we sample $n = 1.2 \times \sqrt{B}$ samples from interval $[1 \ldots B]$ then the probability of finding a collision is

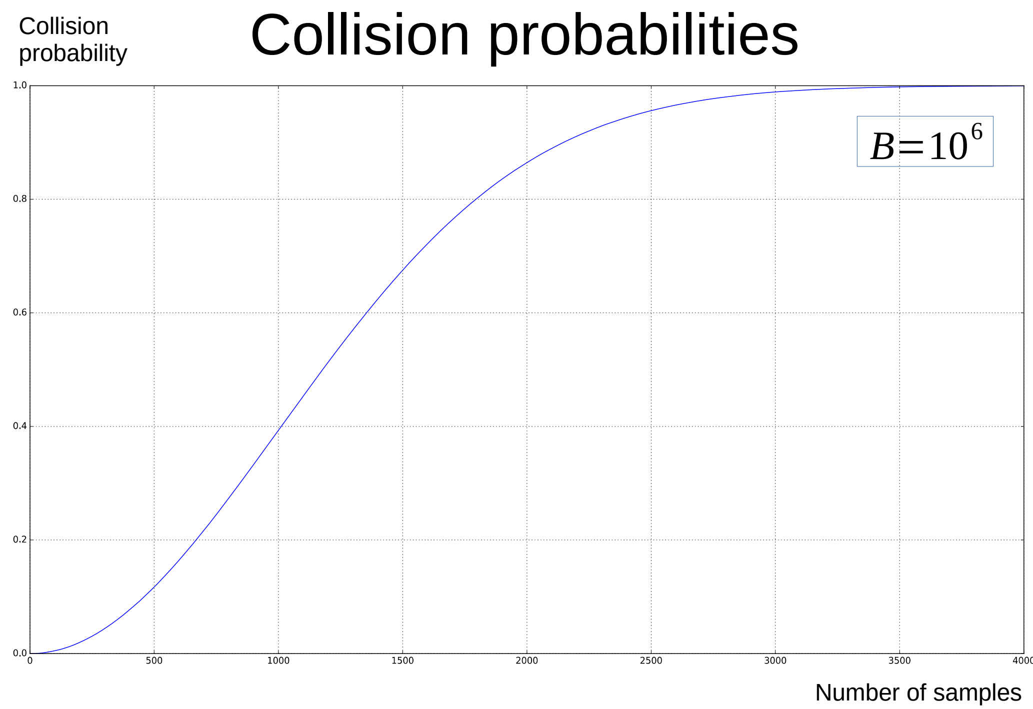
$$\Pr[\exists i \neq j : r_i = r_j] \geq 0.5$$

- Approximation of collision probability given *n* samples with Taylor series

$$p(n) \approx 1 - e^{\frac{-n(n-1)}{2B}}$$

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Collision probabilities



Generic attack on CR

- Let $H: M \to \{0,1\}^n$ be a hash function $|M| \gg 2^n$
- Generic algorithm to find a collision
  1) Chose $\sqrt{2^n} = 2^{\frac{n}{2}}$ random messages: $m_1 \ldots m_{2^{n/2}} \in M$ distinct w.h.p.
  2) For $i = 1 \ldots 2^{n/2}$: compute $t_i = H(m_i)$
  3) Look for a collision $(t_i = t_j)$. If not found, go to 1.

- How many iterations before we find a collision?
  - ~ 2
  - Running time $O(2^{\frac{n}{2}})$

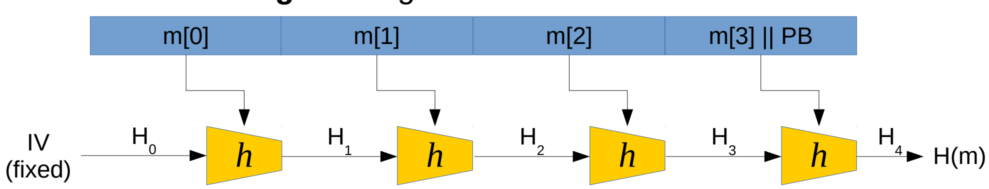
Example CR hash functions

Function	Digest (tag) size [bits]	Generic attack time
MD-5	128	2^{64}
SHA-1*	160	2^{80}
SHA-256	256	2^{128}
SHA-512	512	2^{256}
Whirpool	512	2^{256}

* Found collision by performing $2^{63.1}$ evaluations https://shattered.it

Merkle-Damgard construction

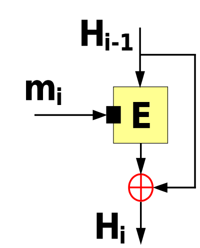
- <u>Goal:</u> given CR function for **short** messages, construct CR function for **long** messages



- CR for short messages (compression function) $h: T \times X \to T$
- CR for long messages $H: X^{\leq L} \to T$
- PB: padding block 10..0 || msg len (in bits) — 64-bit
  - If no space for PB, add an extra block
- **Thm.** If *h* is CR, so is *H.*

Compression functions

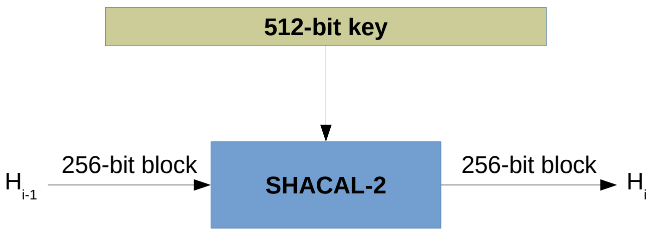
- Built from block ciphers $E: K \times \{0,1\}^n \to \{0,1\}^n$
- Several constructions
  - **Davies-Meyer**
    $h(H, m) := E(m, H) \oplus H$
  - Matyas–Meyer–Oseas
  - Miyaguchi–Preneel



https://en.wikipedia.org/wiki/One-way_compression_function

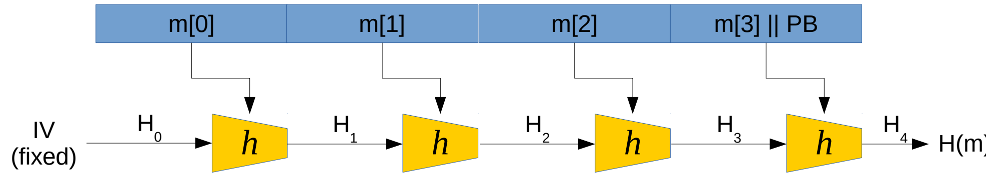
Example: SHA-256

- Merkle-Damgard iterative construction
- Davies-Meyer compression function
  - Block cipher: SHACAL-2



MAC from M-D hash func.

- Can we construct a MAC directly from *H*? (e.g SHA-256)
- Naive attempt $S(k,m) := H(k \| m)$
  - Is it secure?



  - If you knew $H(k \| m)$ could you compute $H(k \| m \| \mathrm{PB} \| w)$ for any w ? How?
  - Length-extension attack

Standardized solution: HMAC

- Most commonly used on the Internet
  - https://tools.ietf.org/html/rfc2104
- Given CR hash function *H,* define a MAC as

$$S(k,m) := H(k \oplus \mathrm{opad} \quad \| \quad H(k \oplus \mathrm{ipad} \| m))$$

  - Built from a black-box implementation of SHA-256
  - Assumed to be a secure PRF
  - TLS 1.2 requires support of HMAC-SHA1-96 (TLS 1.3 does not)

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 <u>integrity</u>
  - or <u>confidentiality</u> (security against eavesdropping)
- CPA security does not provide secrecy against active attacks (where an attacker can tamper with ciphertext)
  - ➔ If you require <u>integrity</u> $\rightarrow$ **MAC**
  - ➔ If you require <u>integrity and confidentiality</u> $\rightarrow$ **AE**

Adaptation of: Dan Boneh, Cryptography I, Stanford.

AE: Desired properties

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

  as usual $\quad E : K \times M \times N \rightarrow C$

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

  Nonce

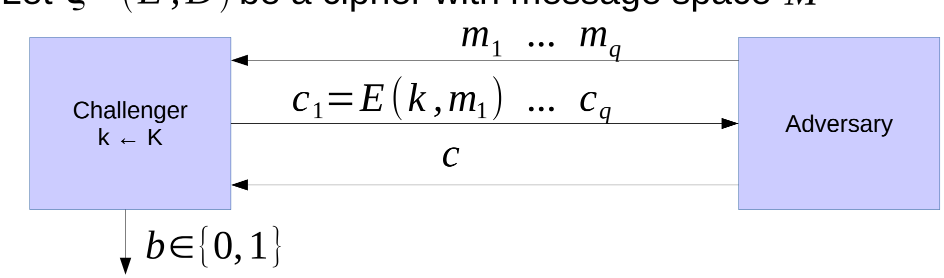
  CT is invalid (rejected)

- 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)

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Ciphertext integrity (def)

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



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

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

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

Adaptation of: Dan Boneh, Cryptography I, Stanford.

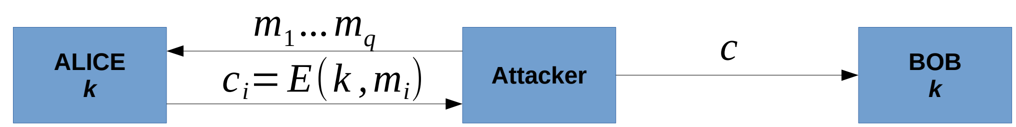
Authenticated Encryption

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

- 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

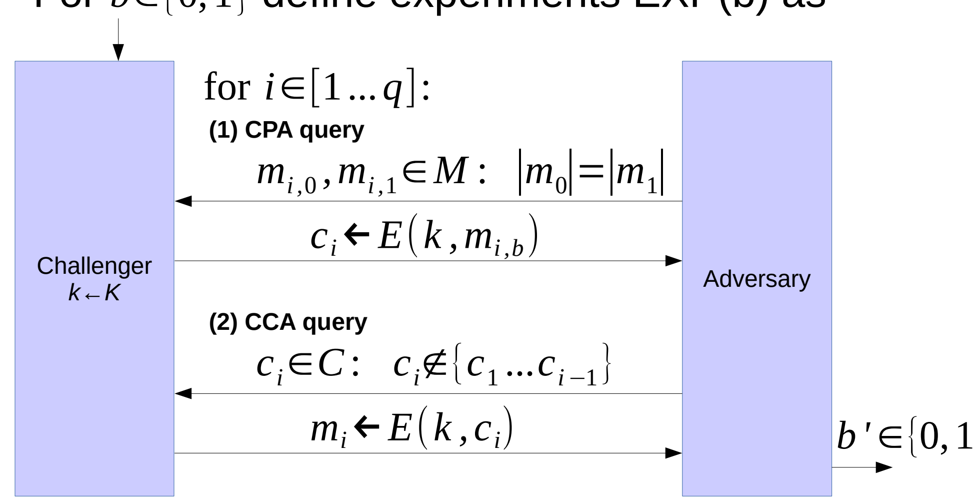
- Implication 2: Security against **chosen ciphertext attack (CCA)**

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

Adaptation of: Dan Boneh, Cryptography I, Stanford.

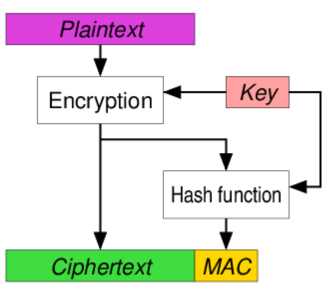
Ex: AES-CTR is not CCA secure

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

Adaptation of: Dan Boneh, Cryptography I, Stanford.

Encrypt then MAC

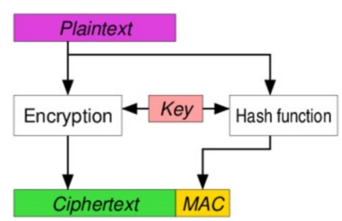
- MAC computed over cipher text
- Used in IPsec, always provides AE
  - Use separate and independent keys



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

Encrypt and MAC

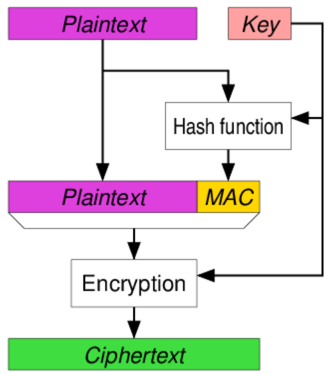
- MAC computed over plain text and sent unencrypted
- Used in SSH
- Use separate and independent keys



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

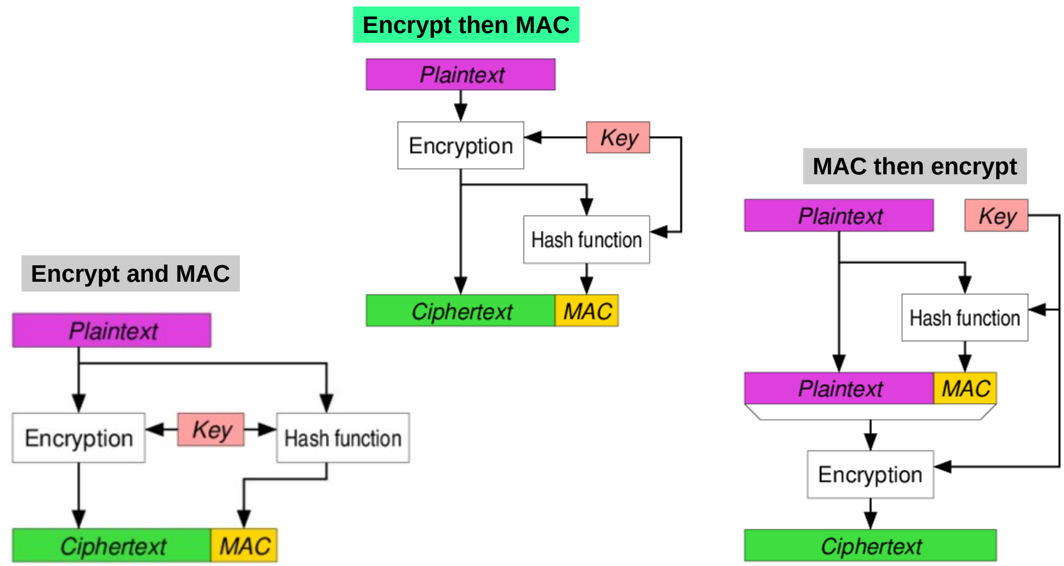
MAC then encrypt

- 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)