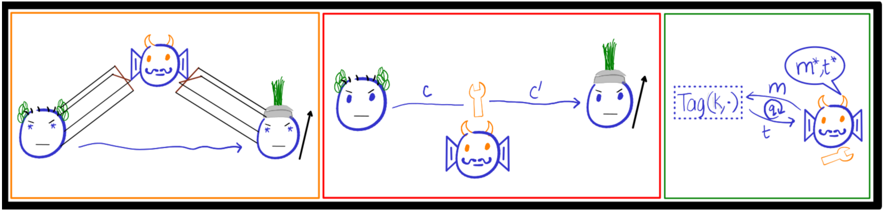
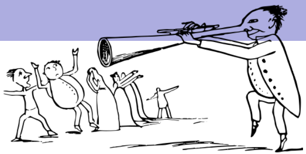


CS409m: Introduction to Cryptography

Lecture 10 (10/Sep/25)

Instructor: Chethan Kamath

Announcements



- Quiz 1: submit cribs by end of today (10/Sep, 23:59)
  - Drop by CC305 after lecture to view your answer sheet

- Assignment 3 (ungraded) released on Monday (08/Sep)

- Mid-sem feedback at the end of the lecture

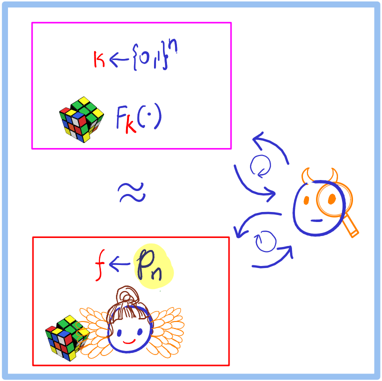
Recall from Previous Lecture...

- Task: secure comm. of *multiple long messages* with shared keys
- Threat model: IND-CPA

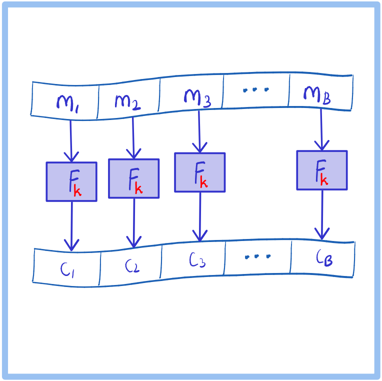
Recall from Previous Lecture . . .

- Task: secure comm. of *multiple long messages* with shared keys
- Threat model: IND-CPA

Block cipher



Modes of Operation



Recall from Previous Lecture...

$|\text{key}| = |\text{Message block}| := n \quad \#\text{Message blocks} := B$

	Baseline	ECB	CBC	OFB	CTR	Ideal
$\lvert$Ciphertext$\rvert$	$2nB$	nB	$nB + n$	$nB + n$	$nB + n$	$nB + n$
#Random coins	nB	0	n	n	n	n
Paralellisable?	✓	✓	×	×	✓	✓
IND-CPA-secure?	✓	×	✓	✓	✓	✓
Assumption on F	PRF	N.A.	PRP	PRF	PRF	PRF

Recall from Previous Lecture...

$|\text{key}| = |\text{Message block}| := n \quad \#\text{Message blocks} := B$

	Baseline	ECB	CBC	OFB	CTR	Ideal
$\lvert$Ciphertext$\rvert$	$2nB$	nB	$nB + n$	$nB + n$	$nB + n$	$nB + n$
#Random coins	nB	0	n	n	n	n
Paralellisable?	✓	✓	×	×	✓	✓
IND-CPA-secure?	✓	×	✓	✓	✓	✓
Assumption on F	PRF	N.A.	PRP	PRF	PRF	PRF

- Careful with n and IV:
  - After $\approx 2^{n/2}$ encryptions, IV will repeat with constant probability
  - CTR/OFB mode breaks if IV repeated; CBC mode "recovers"

Plan for Today's Lecture

- Task: secure comm. of *multiple long* messages with shared keys
- Threat model: ind. against chosen-ciphertext attack (IND-CCA)

Plan for Today's Lecture

- Task: secure comm. of *multiple long* messages with shared keys
- Threat model: ind. against chosen-ciphertext attack (IND-CCA)

NEW IND-CCA

© VAwebteam-Wikipedia

Message Authentication Code NEW

© svgsilh.com/23776

Plan for Today's Lecture

- Task: secure comm. of *multiple long* messages with shared keys
- Threat model: ind. against chosen-ciphertext attack (IND-CCA)

NEW **IND-CCA** | Message Authentication Code NEW

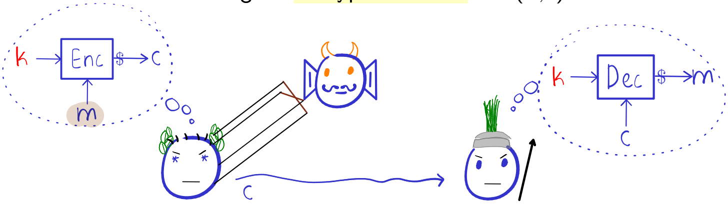
© VAwebteam-Wikipedia

© svgsilh.com/23776

Next Lecture: IND-CPA+MAC $\Rightarrow$ IND-CCA

Recall: Chosen-Plaintext Attack (CPA)

- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$

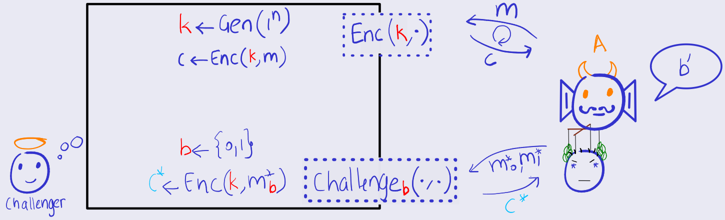


Recall: Chosen-Plaintext Attack (CPA)

- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$

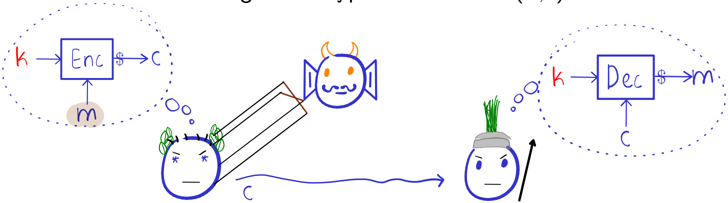
Definition 1 (IND-CPA, Lecture 08)

An SKE $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is CPA-secure if for *every* PPT attacker A
$|\Pr[b' = b] - 1/2|$ is negligible in following game.

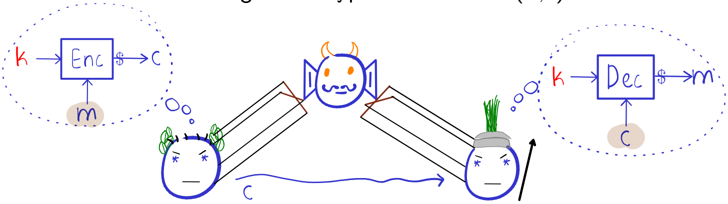


Chosen-Ciphertext Attack (CCA) ...

- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$

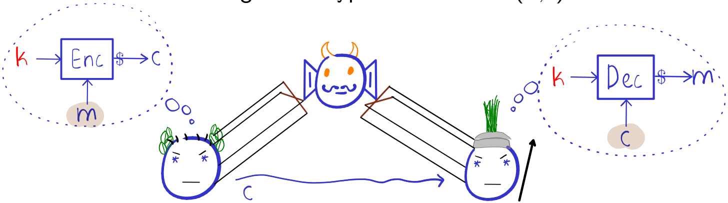


- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$



  - Can also obtain decryption of ciphertexts of its choice
    - Modelled using a decryption oracle $\mathrm{Dec}(k, \cdot)$

- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathsf{Enc}(k, \cdot)$

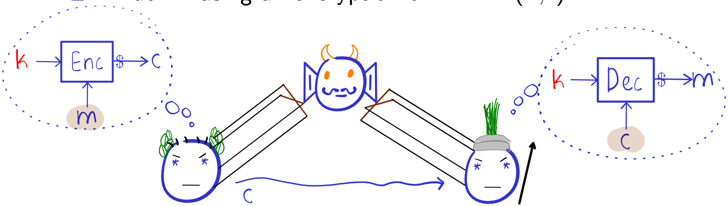


  - Can also obtain decryption of ciphertexts of its choice
    - Modelled using a decryption oracle $\mathsf{Dec}(k, \cdot)$

? Why is decryption oracle useful to the attacker?

Chosen-Ciphertext Attack (CCA)...

- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$



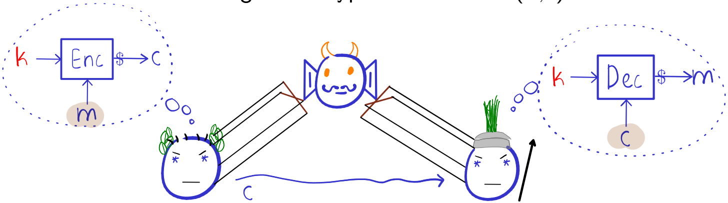
  - Can also obtain decryption of ciphertexts of its choice
    - Modelled using a decryption oracle $\mathrm{Dec}(k, \cdot)$

? Why is decryption oracle useful to the attacker?
- E.g., could obtain decryption of tampered/mauled ciphertexts
- We'll see one example soon: padding-oracle attack ⚠

? Is the decryption oracle justified?

Chosen-Ciphertext Attack (CCA)...

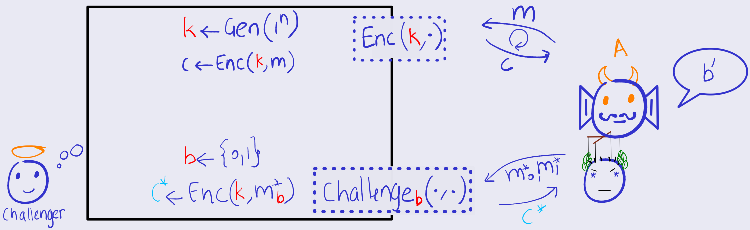
- *Active* attacker:
  - Can influence Caesar's messages
    - Modelled using an encryption oracle $\mathrm{Enc}(k, \cdot)$



  - Can also obtain decryption of ciphertexts of its choice
    - Modelled using a decryption oracle $\mathrm{Dec}(k, \cdot)$
- ? Why is decryption oracle useful to the attacker?
    - E.g., could obtain decryption of tampered/mauled ciphertexts
    - We'll see one example soon: padding-oracle attack ⚠
- ? Is the decryption oracle justified? Yes:
    - E.g. 1: Server sends error message on receiving invalid ciphertext
    - E.g. 2: Receiver could be infected by computer virus

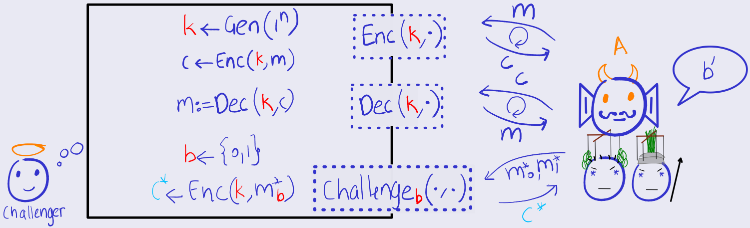
Definition 2 (IND-CCA)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA-secure if for *every* PPT attacker A $|\Pr[b' = b] - 1/2|$ is negligible in following game.



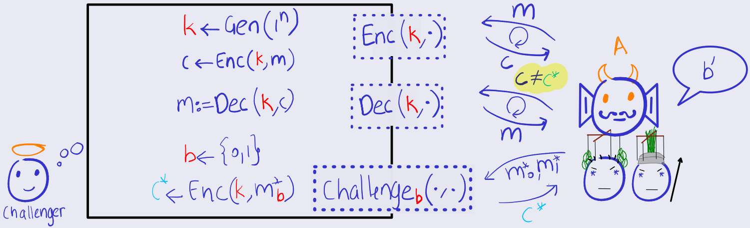
Definition 2 (IND-CCA)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA-secure if for *every* PPT attacker A $|\Pr[b' = b] - 1/2|$ is negligible in following game.



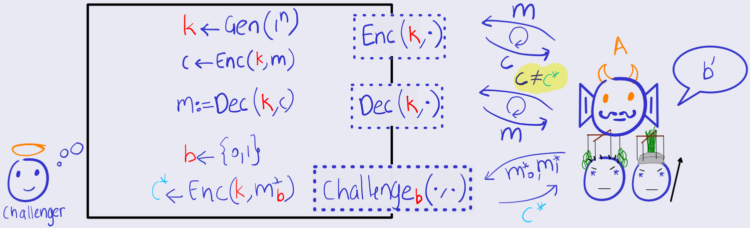
Definition 2 (IND-CCA)

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA-secure if for *every* PPT attacker A $|\Pr[b' = b] - 1/2|$ is negligible in following game.



Definition 2 (IND-CCA)

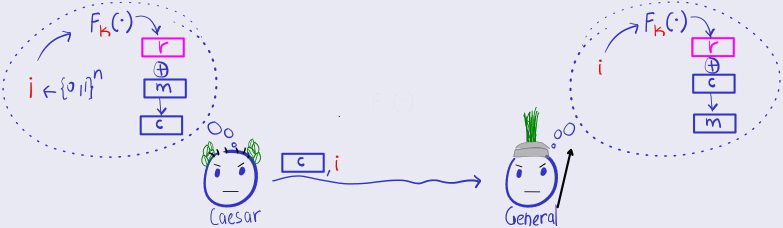
An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA-secure if for *every* PPT attacker A $|\Pr[b' = b] - 1/2|$ is negligible in following game.



Exercise 1 (IND-CCA⇒IND-CPA)

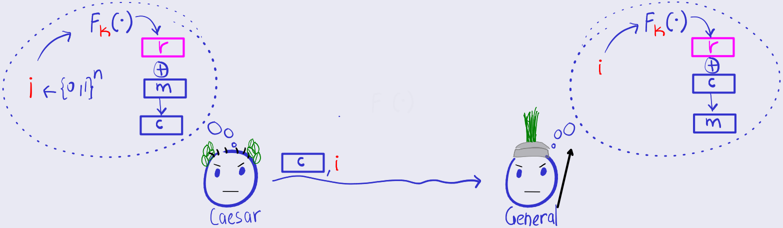
Show that if Π is IND-CCA secure then it is IND-CPA secure

Construction 1 (Lecture 07, PRF ⇒ CPA-SKE)



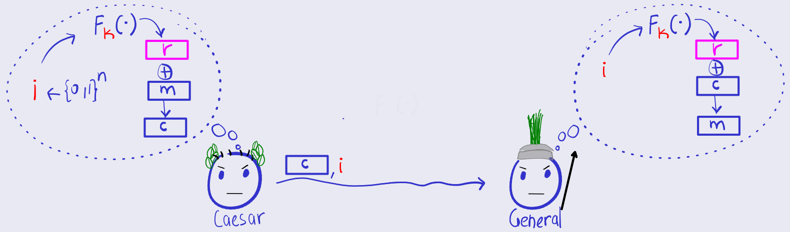
IND-CPA$\not\Rightarrow$IND-CCA!

Construction 1 (Lecture 07, PRF $\Rightarrow$ CPA-SKE)



How to break Construction 1 using decryption oracle?

Construction 1 (Lecture 07, PRF $\Rightarrow$ CPA-SKE)

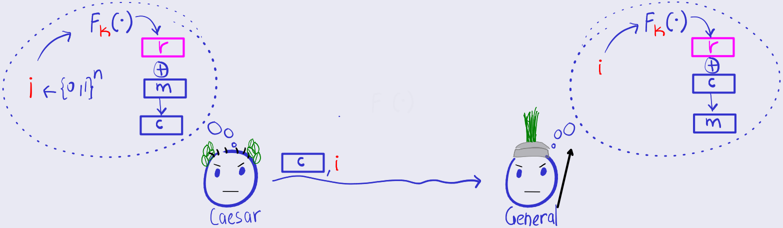


How to break Construction 1 using decryption oracle?

Hint: can you *modify* a ciphertext to get another valid ciphertext?

IND-CPA$\not\Rightarrow$IND-CCA!

Construction 1 (Lecture 07, PRF $\Rightarrow$ CPA-SKE)



How to break Construction 1 using decryption oracle?

Hint: can you *modify* a ciphertext to get another valid ciphertext?

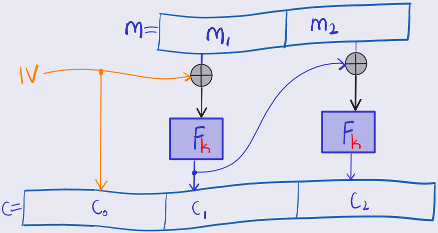
The attack:

1. Challenge on $m_1^* := 0^n$ and $m_2^* := 1^n$ to obtain $c^* := (c_1^*, c_2^*)$
2. Query decryption oracle on $(c_1^*, c_2^* \oplus 1\|0^{n-1})$ to obtain m^*
3. Output $b' := 0$ if $m^* = 1\|0^{n-1}$, and $b' := 1$ otherwise

Decryption Oracle IRL: Oracle-Padding Attack...

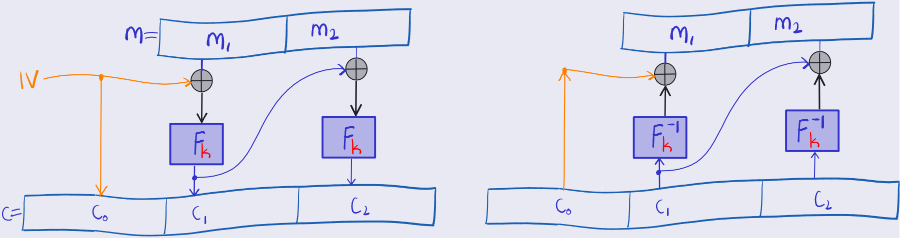
2

Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)



Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)

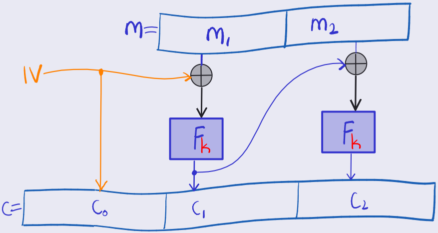
2



Decryption Oracle IRL: Oracle-Padding Attack...

Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)

$\to 2$

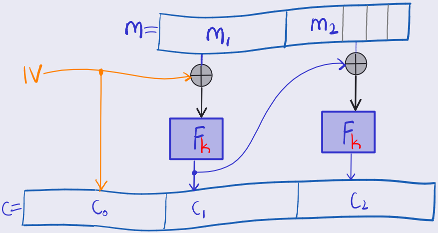


What if $|m| \neq n \cdot B$ bytes for some B? Say m is s bytes *short*

Decryption Oracle IRL: Oracle-Padding Attack...

$\to 2$

Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)



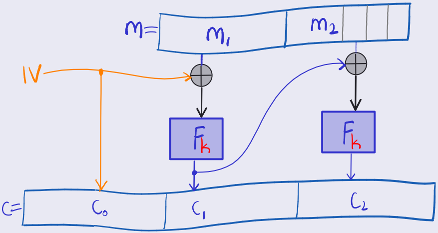
$\to 3$

❓ What if $|m| \neq n \cdot B$ bytes for some B? Say m is s bytes *short*

Decryption Oracle IRL: Oracle-Padding Attack...

Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)

$\to 2$



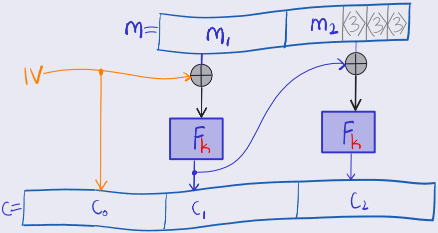
$\to 3$

? What if $|m| \neq n \cdot B$ bytes for some B? Say m is s bytes *short*

- We need to "pad" m with an s byte string. How?

2

Construction 3 (Lecture 09, CBC mode for $|m| = n \cdot B$ bytes)



3

? What if $|m| \neq n \cdot B$ bytes for some B? Say m is s bytes *short*

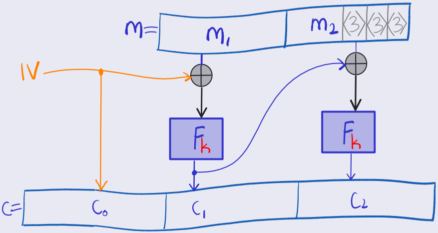
- We need to "pad" m with an s byte string. How?
- PKCS#7 std.: If $\langle s \rangle$ is byte representation of s, then padding is

$$\underbrace{\langle s \rangle \| \cdots \| \langle s \rangle}_{s \text{ times}}$$

Decryption Oracle IRL: Oracle-Padding Attack...

2

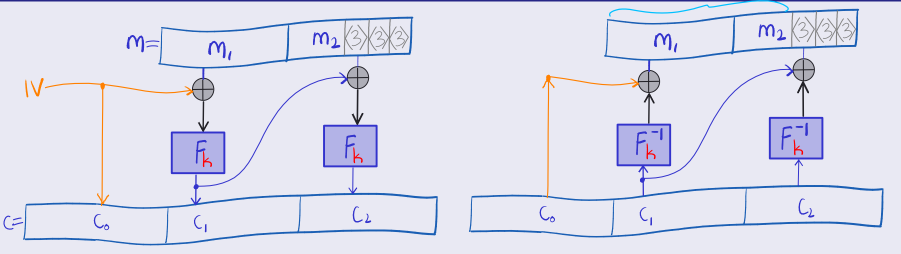
Construction 1 (CBC mode with padding)



- To encrypt m: encrypt $m \| \underbrace{\overset{3}{\langle s \rangle} \| \cdots \| \langle s \rangle}_{s \text{ times}}$ in CBC mode

Decryption Oracle IRL: Oracle-Padding Attack…

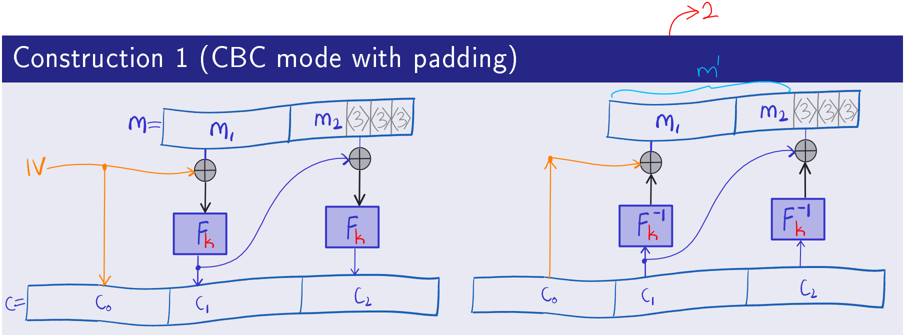
Construction 1 (CBC mode with padding)



- To encrypt m: encrypt $m \| \underbrace{\langle s \rangle \| \cdots \| \langle s \rangle}_{s \text{ times}}$ in CBC mode
- To decrypt c:
  1. Decrypt c in CBC mode to obtain message of form $m' \| \langle s' \rangle \| \cdots \| \langle s' \rangle$
  2. If last s' bytes are all $\langle s' \rangle$ then o/p m'
     Else o/p "bad padding"

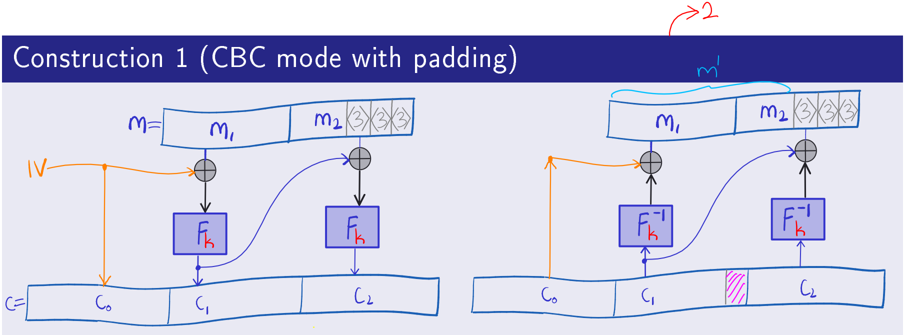
Decryption Oracle IRL: Oracle-Padding Attack...

Construction 1 (CBC mode with padding)



❓ How to break Construction 1 using decryption oracle?

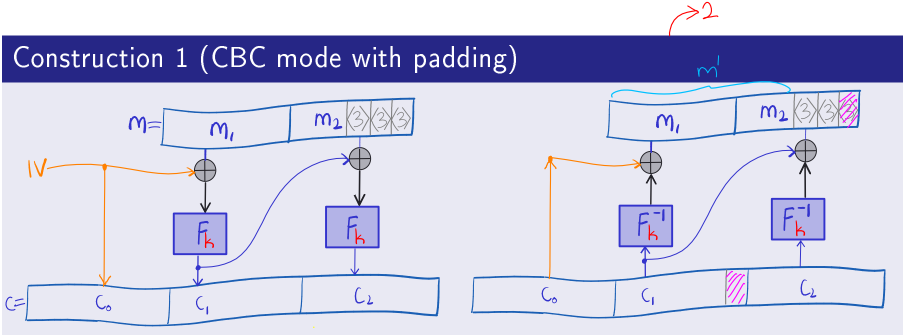
Construction 1 (CBC mode with padding)



How to break Construction 1 using decryption oracle?

Hint: What happens if you modify last byte of c_1?

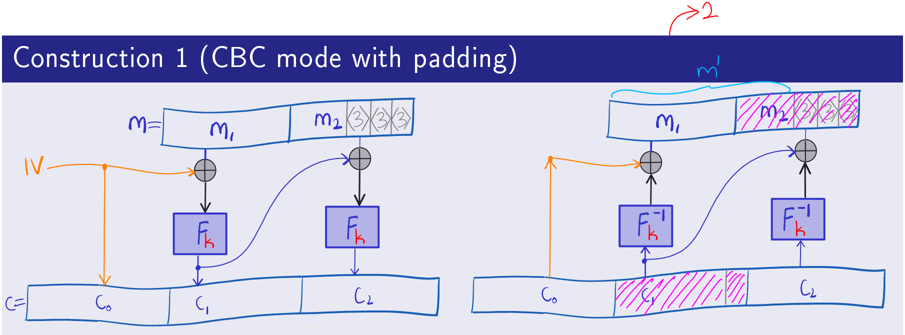
Construction 1 (CBC mode with padding)



How to break Construction 1 using decryption oracle?

Hint: What happens if you modify last byte of c_1?

Construction 1 (CBC mode with padding)



How to break Construction 1 using decryption oracle?

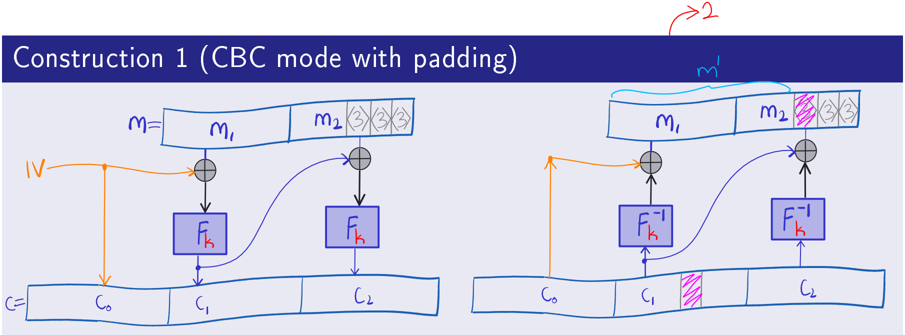
Hint: What happens if you modify last byte of c_1?

- Note that $m_2 = \mathsf{F}_k^{-1}(c_2) \oplus c_1$

Observation: for any Δ, $c_1' := c_1 \oplus \Delta \implies$ decryption of (c_0, c_1', c_2) yields (m_1', m_2') where $m_2' = m_2 \oplus \Delta$

- ...

Construction 1 (CBC mode with padding)



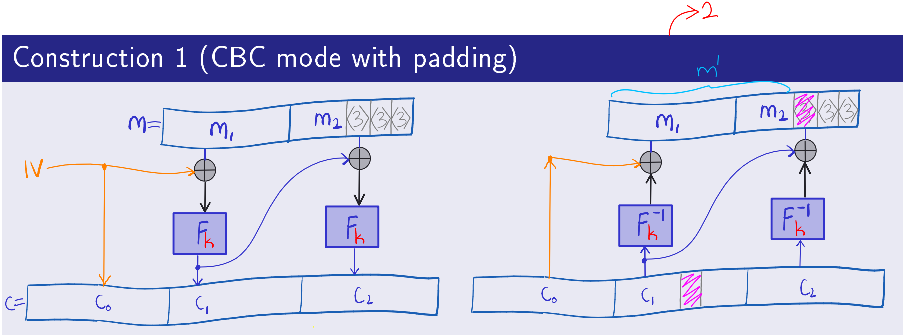
- ...

⚠ Attack to recover s. For each $i \in [1, n]$:

1. Set $c_1^{(i)}$ as c_1 with i-th byte modified (arbitrarily)
2. Query decryption oracle with $(c_0, c_1^{(i)}, c_2)$
3. If oracle returns "bad padding", output $n - i$ and halt

Decryption Oracle IRL: Oracle-Padding Attack...

Construction 1 (CBC mode with padding)



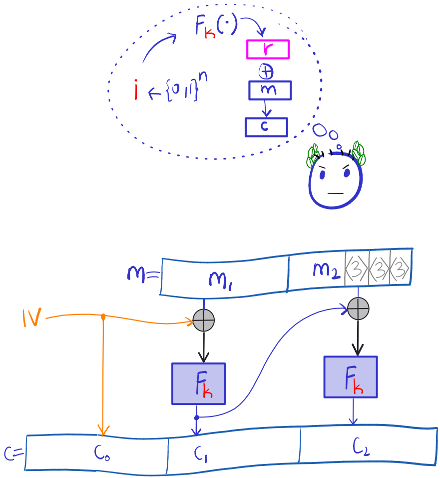
- ...

⚠ Attack to recover s. For each $i \in [1, n]$:

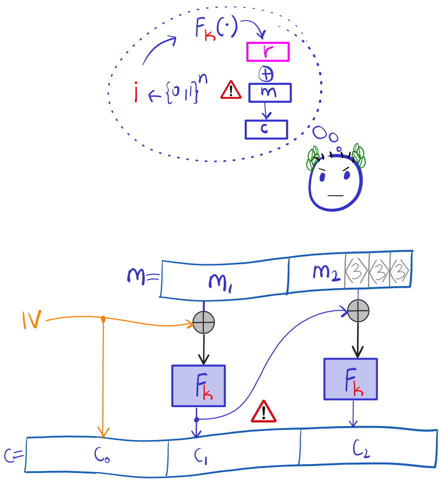
1. Set $c_1^{(i)}$ as c_1 with i-th byte modified (arbitrarily)
2. Query decryption oracle with $(c_0, c_1^{(i)}, c_2)$
3. If oracle returns "bad padding", output $n - i$ and halt

? How to recover rest of message? Lab Exercise 2, Problem 4

What Made These Schemes Vulnerable? ⚠



What Made These Schemes Vulnerable? ⚠



- Ciphertext is *malleable*! Prevent mauling using MAC

Plan for Today's Lecture

- Task: secure comm. of *multiple long* messages with shared keys
- Threat model: ind. against chosen-ciphertext attack (IND-CCA)

NEW IND-CCA

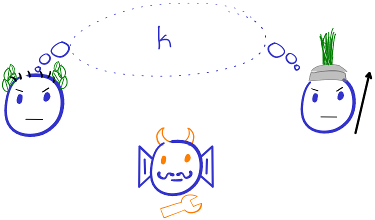
© VAwebteam-Wikipedia

Message Authentication Code NEW

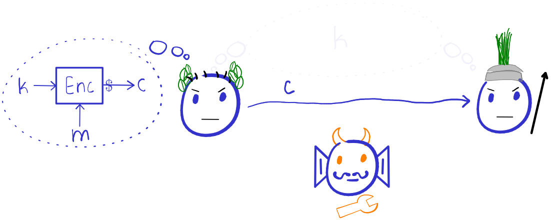
© svgsilh.com/23776

Next Lecture: IND-CPA+MAC $\Rightarrow$ IND-CCA

- The setting: Caesar and his general share key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam

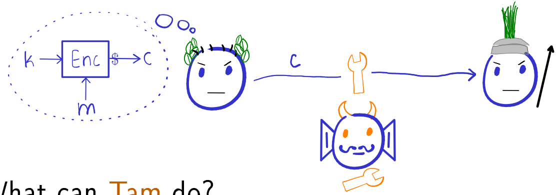


- The setting: Caesar and his general share key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



What Exactly Is the Security Goal?...

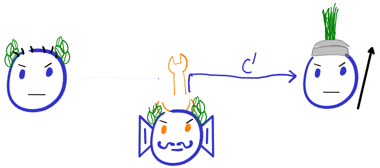
- The setting: Caesar and his general share key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caesar sends to the General (integrity)

What Exactly Is the Security Goal?...

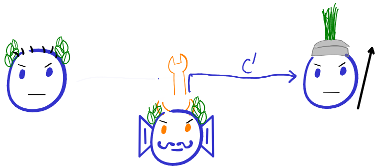
- The setting: Caesar and his general share key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caesar sends to the General (integrity)
     ⚠ All schemes we've seen so far are malleable and allow this!
  2. Try to impersonate Caesar by injecting messages (authenticity)

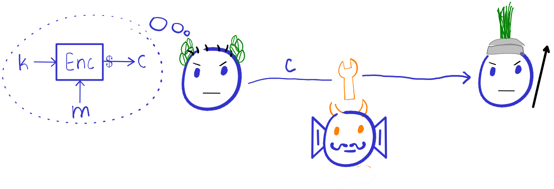
What Exactly Is the Security Goal?

- The setting: Caesar and his general share key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



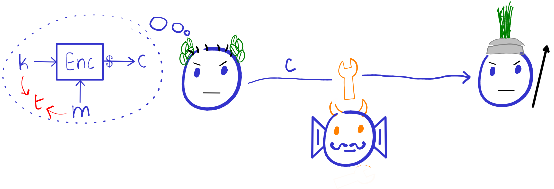
- What can Tam do?
  1. Modify what Caesar sends to the General (integrity)
     ⚠ All schemes we've seen so far are malleable and allow this!
  2. Try to impersonate Caesar by injecting messages (authenticity)
- We cannot prevent this: the hope is to *detect* when it happens

What Exactly Is the Security Goal?...



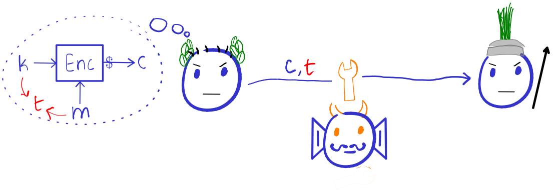
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



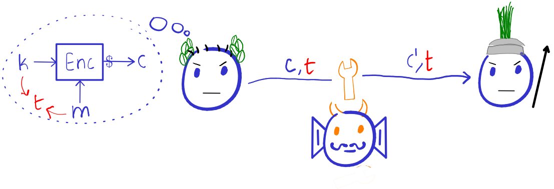
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...

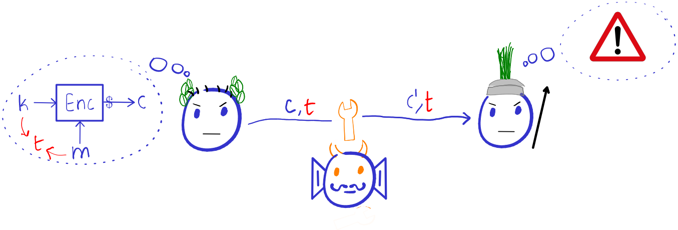


- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...

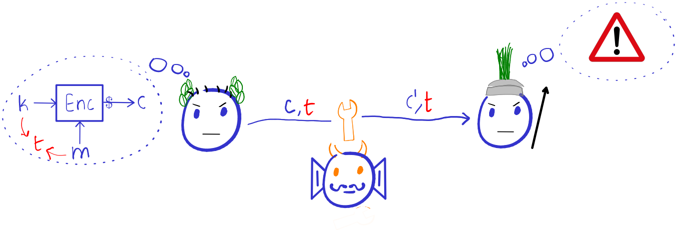


- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext



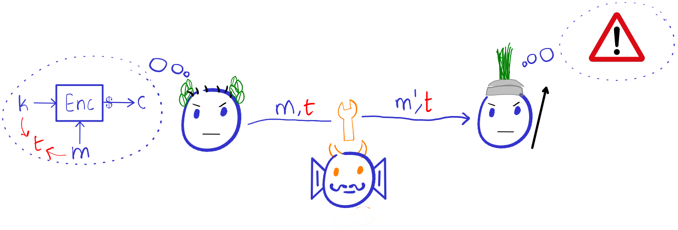
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext 

What Exactly Is the Security Goal?...



- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext
  - Message-authentication code (MAC)
    - Think of it as "cryptographic" version of error detection!

What Exactly Is the Security Goal?...



- How do we ensure integrity and authenticity? 
  - Append "additional information" t with the ciphertext
  - Message-authentication code (MAC)
    - Think of it as "cryptographic" version of error detection!
- For now, let's forget about secrecy and focus on detecting tampering
  - Why? Modularity
  - Lecture 11: MAC + CPA-secure SKE $\Rightarrow$ CCA-secure SKE

Syntax of Message-Authentication Code (MAC)

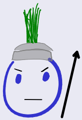
Definition 1 (Message-Authentication Code (MAC))

An MAC M is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ with the following syntax:

Syntax of Message-Authentication Code (MAC)

Definition 1 (Message-Authentication Code (MAC))

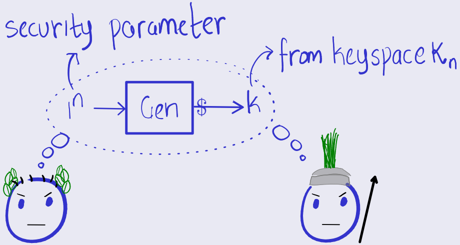
An MAC M is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ with the following syntax:



Syntax of Message-Authentication Code (MAC)

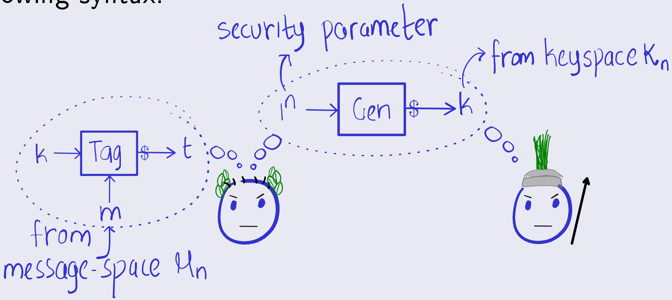
Definition 1 (Message-Authentication Code (MAC))

An MAC M is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ with the following syntax:



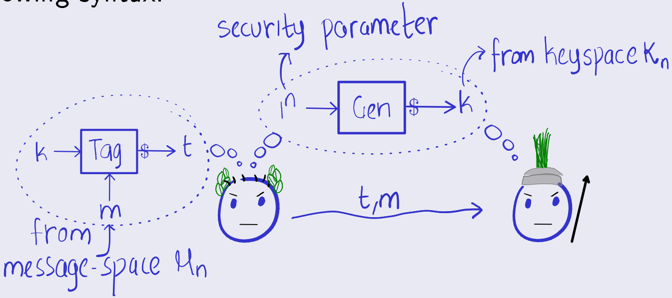
Definition 1 (Message-Authentication Code (MAC))

An MAC M is a triple of efficient algorithms (Gen, Tag, Ver) with the following syntax:



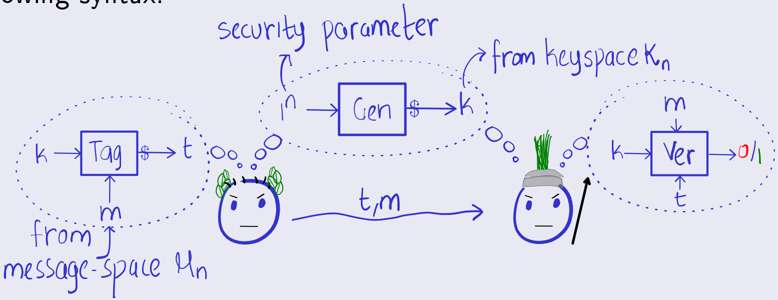
Definition 1 (Message-Authentication Code (MAC))

An MAC M is a triple of efficient algorithms (Gen, Tag, Ver) with the following syntax:



Definition 1 (Message-Authentication Code (MAC))

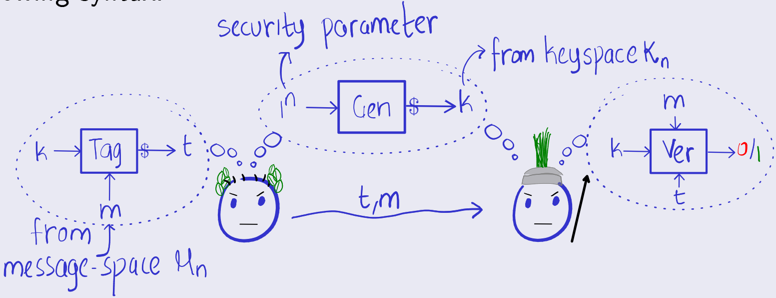
An MAC M is a triple of efficient algorithms (Gen, Tag, Ver) with the following syntax:



Syntax of Message-Authentication Code (MAC)

Definition 1 (Message-Authentication Code (MAC))

An MAC M is a triple of efficient algorithms (Gen, Tag, Ver) with the following syntax:



- Correctness of verification: for every $n \in \mathbb{N}$, message $m \in \mathcal{M}_n$,

$$\Pr_{k \leftarrow \mathrm{Gen}(1^n), t \leftarrow \mathrm{Tag}(k,m)}[\mathrm{Ver}(k,t,m) = 1] = 1$$

How to Define Security?...

Intuitively, what are the security requirements?

How to Define Security?...

Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags...

How to Define Security?...

Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags...
  - ... on messages of its choice

How to Define Security?...

Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags...
  - ... on messages of its choice
- The forged new tag can be on *any* message of Tam's choice

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge valid *new* tag from previously-seen tags...
    - ... on messages of its choice
  - The forged new tag can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

How to Define Security?...

Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags...
  - ... on messages of its choice
- The forged new tag can be on *any* message of Tam's choice

- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

A MAC $M = (\text{Gen}, \text{Tag}, \text{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ

How to Define Security?...

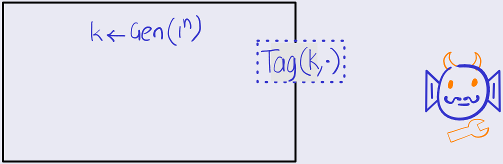
Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags...
  - ... on messages of its choice
- The forged new tag can be on *any* message of Tam's choice

- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

A MAC $M = (\text{Gen}, \text{Tag}, \text{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ

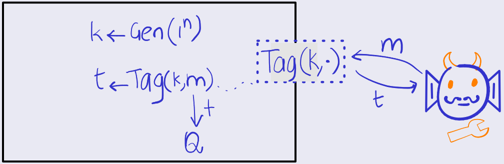


How to Define Security?

- Intuitively, what are the security requirements?
  - Tam must not be able to forge valid *new* tag from previously-seen tags...
    - ... on messages of its choice
  - The forged new tag can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

A MAC $M = (\text{Gen}, \text{Tag}, \text{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ

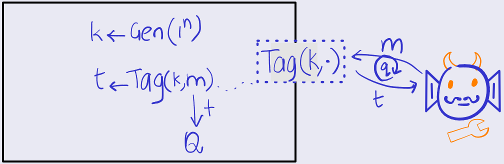


- Intuitively, what are the security requirements?
  - Tam must not be able to forge valid *new* tag from previously-seen tags...
    - ... on messages of its choice
  - The forged new tag can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

A MAC $M = (\text{Gen}, \text{Tag}, \text{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ

- Tam makes q queries to Tag(k,·) oracle

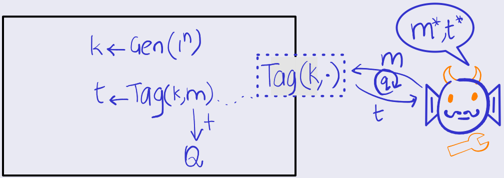


How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge valid *new* tag from previously-seen tags...
    - ... on messages of its choice
  - The forged new tag can be on *any* message of Tam's choice
- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if

How to Define Security?

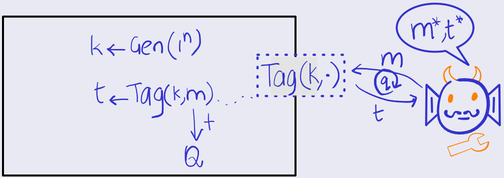
Intuitively, what are the security requirements?

- Tam must not be able to forge valid *new* tag from previously-seen tags…
  - … on messages of its choice
- The forged new tag can be on *any* message of Tam's choice

- Existential Unforgeability Under Chosen-Message Attack

Definition 3 (EU-CMA)

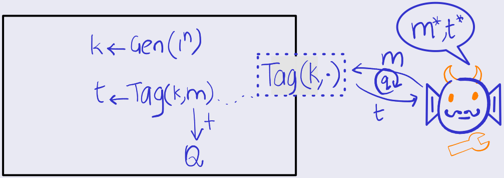
A MAC $M = (\mathrm{Gen}, \mathrm{Tag}, \mathrm{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m^*, t^*) and breaks if
  - i) $m^* \notin Q$
  - ii) $\mathrm{Ver}(k, t^*, m^*) = 1$

Definition 3 (EU-CMA)

A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



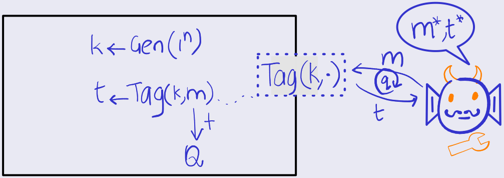
- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  - i) $m^* \notin Q$
  - ii) $\mathsf{Ver}(k, t^*, m^*) = 1$

Definition 3 (EU-CMA)

Typically → negligible

→ polynomial

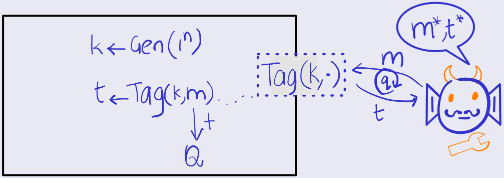
A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to $Tag(k,\cdot)$ oracle
- In the end Tam outputs (m^*, t^*) and breaks if
  - I) $m^* \notin Q$
  - II) $Ver(k, t^*, m^*) = 1$

Definition 3 (EU-CMA) Typically ϵ → negligible, q → polynomial

A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  - I) $m^* \notin Q$
  - II) $\mathsf{Ver}(k, t^*, m^*) = 1$

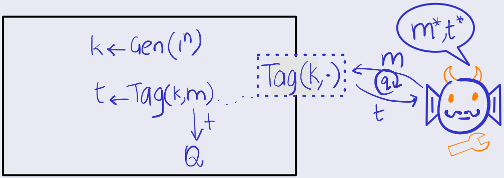
MAC or not?

1. Encrypt to MAC: Given SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, define:
   - $\mathsf{Tag}(k, m) := \mathsf{Enc}(k, m)$
   - $\mathsf{Ver}(k, t, m)$: Compute $m' := \mathsf{Dec}(k, t)$ and accept if $m = m'$

How to Define Security?...

Definition 3 (EU-CMA) Typically negligible, polynomial

A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to $\mathsf{Tag}(k,\cdot)$ oracle
- In the end Tam outputs (m^*,t^*) and breaks if
  - i) $m^* \notin Q$
  - ii) $\mathsf{Ver}(k,t^*,m^*)=1$

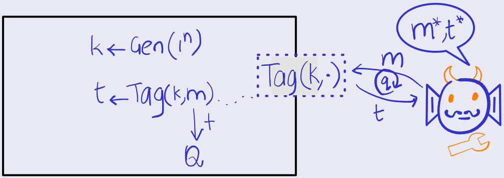
MAC or not?

1. Encrypt to MAC: Given SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, define:
   - $\mathsf{Tag}(k,m) := \mathsf{Enc}(k,m)$
   - $\mathsf{Ver}(k,t,m)$: Compute $m' := \mathsf{Dec}(k,t)$ and accept if $m = m'$
2. Append-0 MAC: Given MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$, define M' as
   - $\mathsf{Tag}'(k,m) := t\|0$, where $t \leftarrow \mathsf{Tag}(k,m)$
   - $\mathsf{Ver}'(k,t\|b,m) := \mathsf{Ver}(k,t,m)$

How to Define Security?...

Definition 3 (EU-CMA) Typically → negligible → polynomial

A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ is (ϵ, q)-EU-CMA secure if no PPT tampering adversary Tam that makes at most q queries can break M as below with probability more than ϵ



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m^*, t^*) and breaks if
  - i) $m^* \notin Q$
  - ii) $\mathrm{Ver}(k, t^*, m^*) = 1$

? MAC or not?

1. Encrypt to MAC: Given SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, define:
   - $\mathsf{Tag}(k, m) := \mathsf{Enc}(k, m)$
   - $\mathsf{Ver}(k, t, m)$: Compute $m' := \mathsf{Dec}(k, t)$ and accept if $m = m'$
2. Append-0 MAC: Given MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$, define M' as
   - $\mathsf{Tag}'(k, m) := t \| 0$, where $t \leftarrow \mathsf{Tag}(k, m)$
   - $\mathsf{Ver}'(k, t \| b, m) := \mathsf{Ver}(k, t, m)$

How to Construct a MAC?

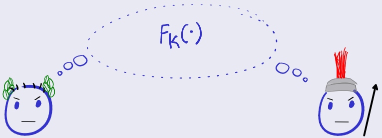
Use a PRF to generate the tag!

Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)

How to Construct a MAC?

Use a PRF to generate the tag!

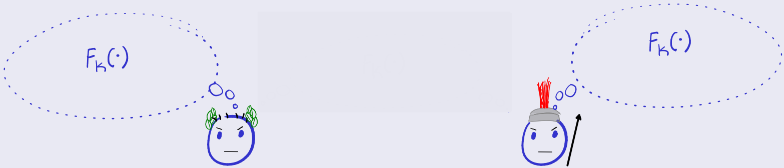
Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



How to Construct a MAC?

Use a PRF to generate the tag!

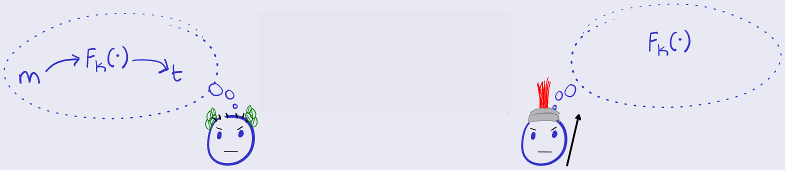
Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



How to Construct a MAC?

Use a PRF to generate the tag!

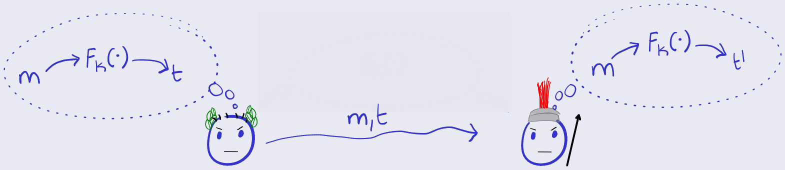
Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



How to Construct a MAC?

Use a PRF to generate the tag!

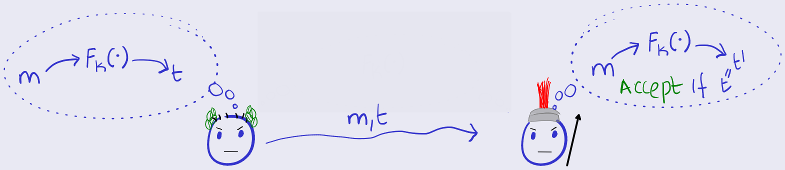
Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



How to Construct a MAC?

Use a PRF to generate the tag!

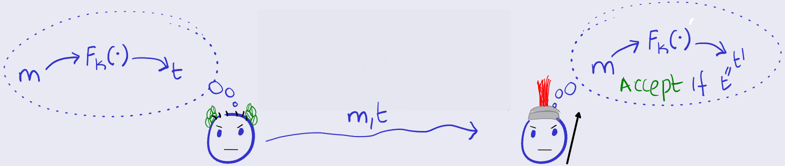
Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



How to Construct a MAC?

Use a PRF to generate the tag!

Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



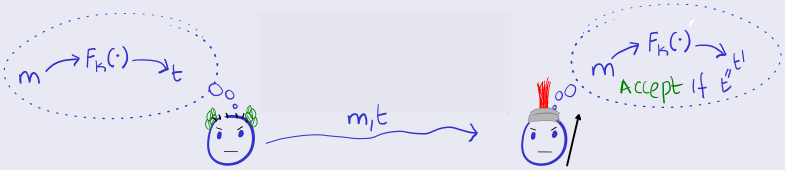
Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^n}$ is a PRF then Construction 2 is EU-CMA-secure against any PPT Tam

How to Construct a MAC?

Use a PRF to generate the tag!

Construction 2 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



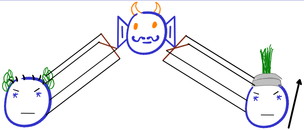
Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\{0,1\}^n}$ is a PRF then Construction 2 is EU-CMA-secure against any PPT Tam

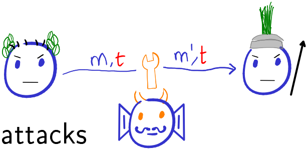
Proof by reduction.

On the whiteboard □

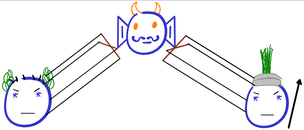
Recap/Next Lecture



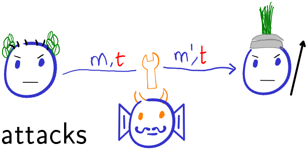
- Saw Chosen-Ciphertext Attack (CCA)
  - Stronger threat model
  - ⚠ CCA IRL: padding oracle attack
    - Affected PKCS#1 v1.5, SSL, IPSEC...



- ☆ Takeaway: ciphertext malleability can lead to attacks
- How to prevent/detect mauling? Use message-authentication codes

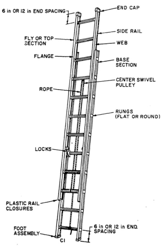


- Saw Chosen-Ciphertext Attack (CCA)
  - Stronger threat model
  - CCA IRL: padding oracle attack
    - Affected PKCS#1 v1.5, SSL, IPSEC...



- Takeaway: ciphertext malleability can lead to attacks
- How to prevent/detect mauling? Use message-authentication codes

- Next lecture
  - How to construct a CCA-secure scheme using MAC
  - Domain-extension for MAC



Further Reading

1. The definition of CCA security can be found in [KL14, §5.1.2]. The notion was introduced by Naor and Yung [NY89]
2. You can read more about oracle-padding attack in [KL14, §5.1.1]. The original attack was due to Bleichenbacher on PKCS#1 v1.5 [Ble98]. Vaudenay came up with the attack on the CBC mode [Vau02].
3. The definition of MAC can be found in [KL14, §4.2]

Daniel Bleichenbacher.

Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1.

In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, August 1998.

Jonathan Katz and Yehuda Lindell.

Introduction to Modern Cryptography (3rd ed.).

Chapman and Hall/CRC, 2014.

Moni Naor and Moti Yung.

Universal one-way hash functions and their cryptographic applications.

In *21st ACM STOC*, pages 33–43. ACM Press, May 1989.

Serge Vaudenay.

Security flaws induced by CBC padding - applications to SSL, IPSEC, WTLS...

In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 534–546. Springer, Berlin, Heidelberg, April / May 2002.