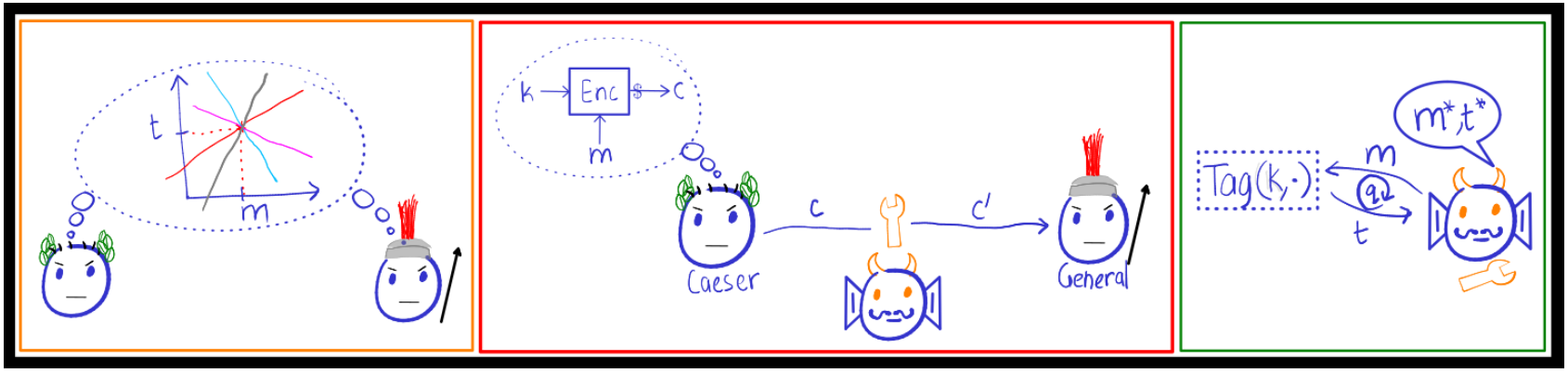


CS783: Theoretical Foundations of Cryptography

Lecture 7 (20/Aug/24)

Instructor: Chethan Kamath

Recall from Last Few Lectures...

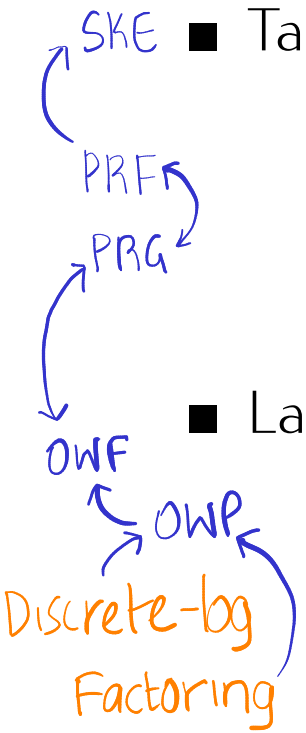
SKE

- Task 1: secret communication in presence of eavesdroppers
  - Later considered the stronger chosen-plaintext attackers
  - In both cases, Task 1 reduced to constructing a PRF
  - Constructing PRF reduces to constructing PRG

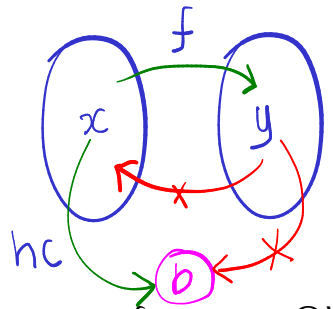
PRF

PRG

Recall from Last Few Lectures...



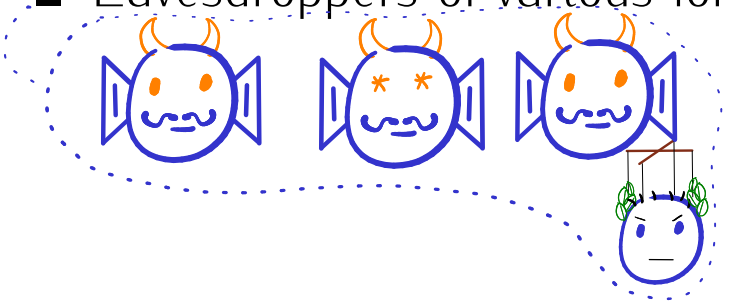
- Task 1: secret communication in presence of eavesdroppers
  - Later considered the stronger chosen-plaintext attackers
  - In both cases, Task 1 reduced to constructing a PRF
  - Constructing PRF reduces to constructing PRG
- Last lecture:
  - Defined OWF and OWP
  - Defined hard-core predicates



    1. Constructed hard-core predicates for any OWP
    2. Hard-core predicate for OWF/OWP → PRG
  - 1+2 ⇒ OWP → PRG
  - Can be relaxed to OWF → PRG, and thus OWF↔PRG

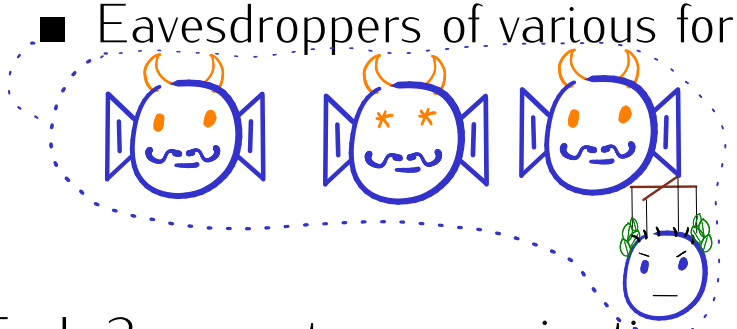
Plan for This Lecture...

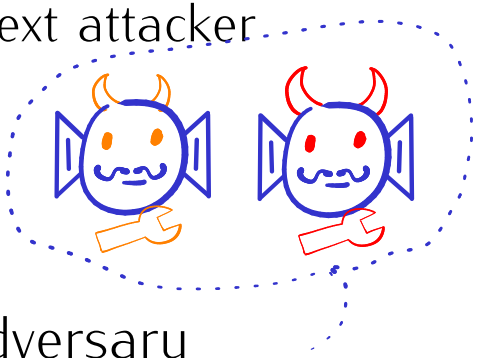
- So far: adversaries who are passive
  - Eavesdroppers of various forms, chosen-plaintext attacker



Plan for This Lecture...

- So far: adversaries who are passive
  - Eavesdroppers of various forms, chosen-plaintext attacker

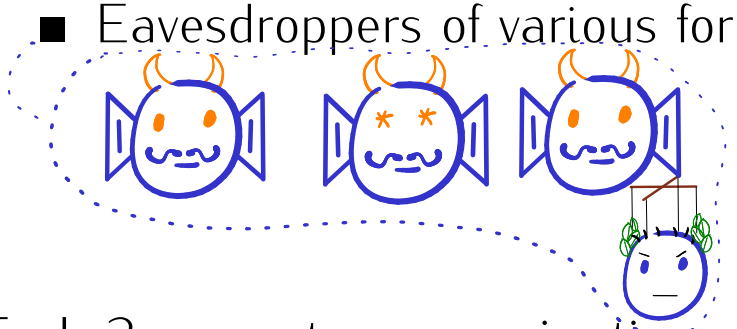


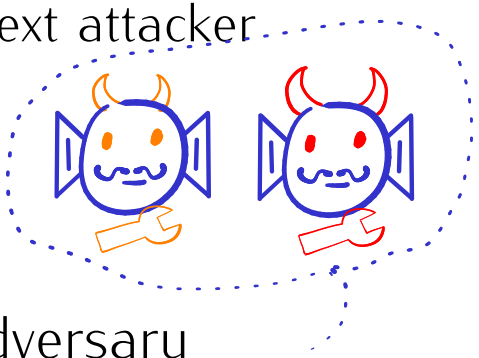


- Task 2: secret communication against *active* adversary

Plan for This Lecture...

- So far: adversaries who are passive
  - Eavesdroppers of various forms, chosen-plaintext attacker

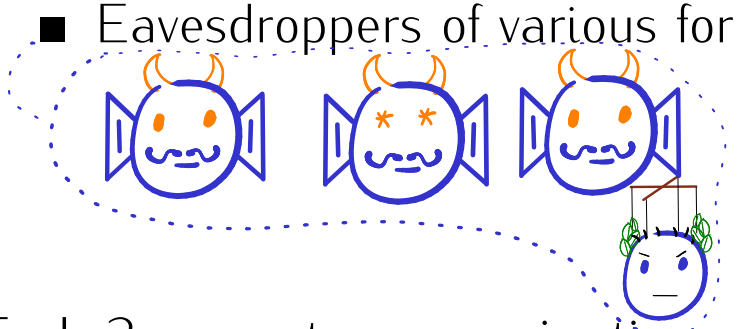


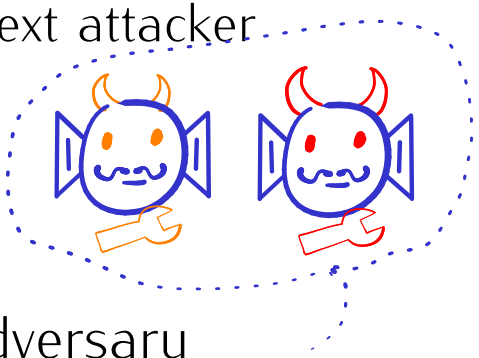


- Task 2: secret communication against *active* adversary
  - Sub-task: How to detect tampering?
    - Message authentication codes (MAC)
    - PRF $\rightarrow$ MAC

Plan for This Lecture...

- So far: adversaries who are passive
  - Eavesdroppers of various forms, chosen-plaintext attacker





- Task 2: secret communication against *active* adversary
  - Sub-task: How to detect tampering?
    - Message authentication codes (MAC)
    - PRF → MAC
  - How to model secrecy against tampering adversary?
    - Chosen-ciphertext attack (CCA)
    - CPA-secret SKE + MAC → CCA-secure SKE

Plan for This Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

General *template*:

1. Identify the sub-task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Detect tampering

chosen-message attacker

Existentially unforgeable

Plan for This Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

General *template*:

secret communication in presence of active adversary

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

General *template*:

secret communication in presence of active adversary

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?

chosen ciphertext attacker

indistinguishability

3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

secret communication in presence of active adversary

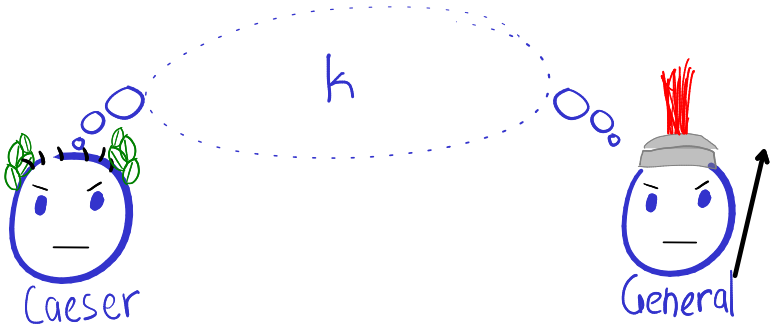
chosen ciphertext attacker

indistinguishability

$\sim$ MAC + CPA-SKE $\rightarrow$ CCA-SKE

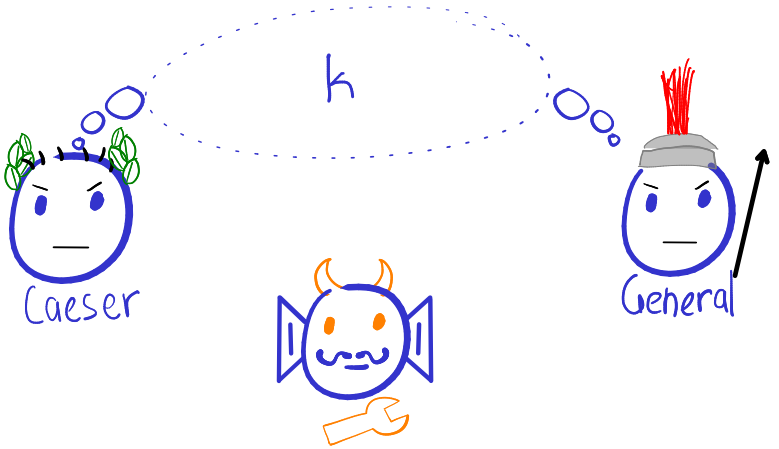
What Exactly Is the Security Goal?...

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



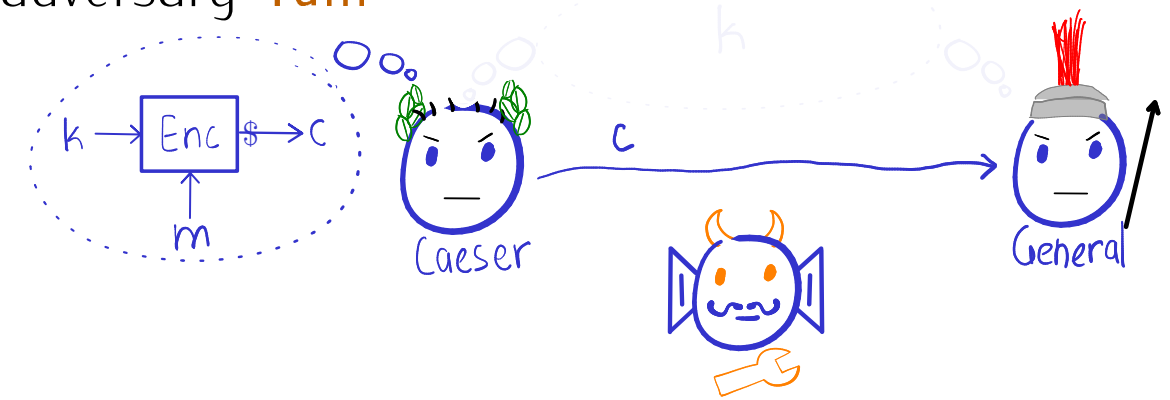
What Exactly Is the Security Goal?...

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



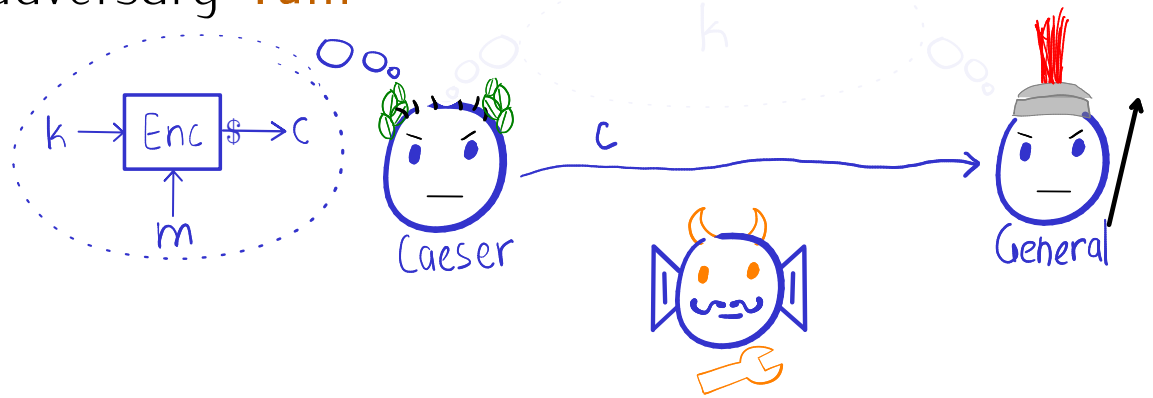
What Exactly Is the Security Goal?...

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



What Exactly Is the Security Goal?...

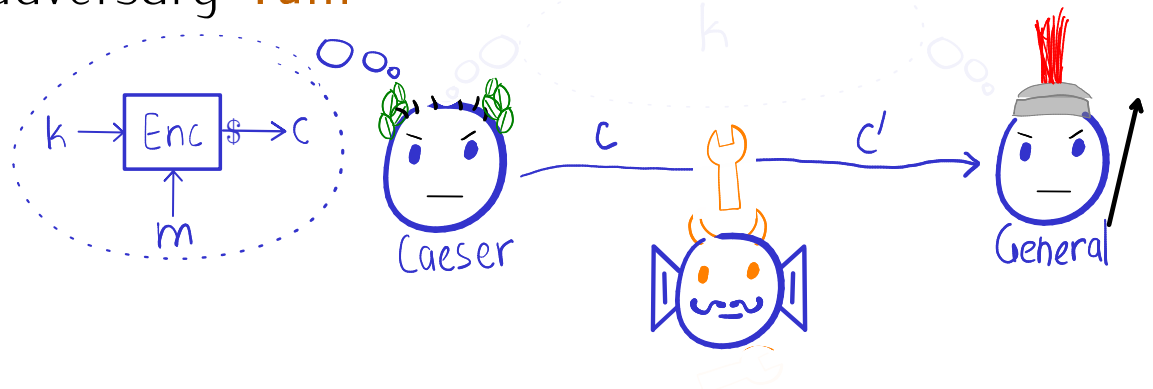
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?

What Exactly Is the Security Goal?...

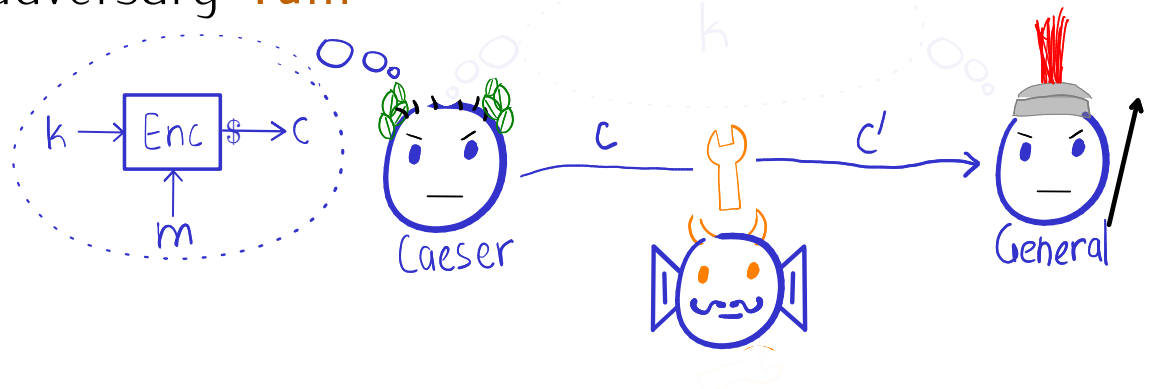
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



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

What Exactly Is the Security Goal?...

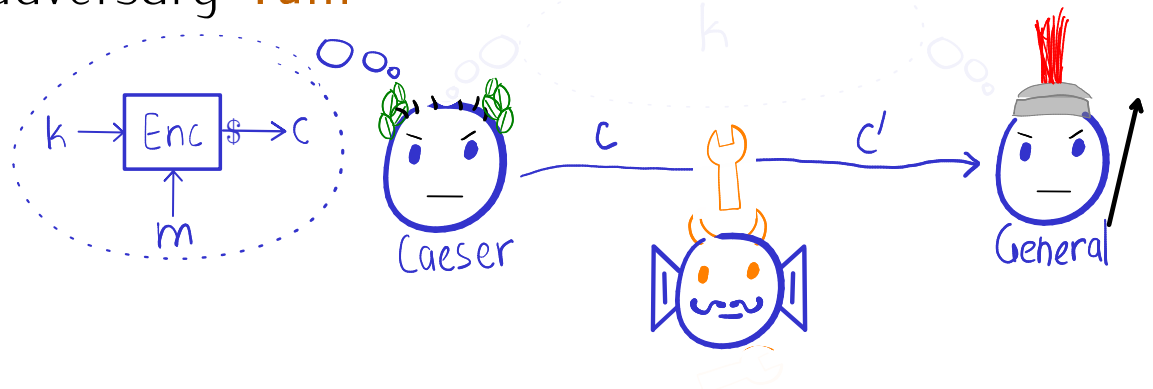
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caeser's sends to the General (integrity)
     - All schemes we've seen so far (OTP, computational OTP, CPA-secret construction) are malleable!

What Exactly Is the Security Goal?...

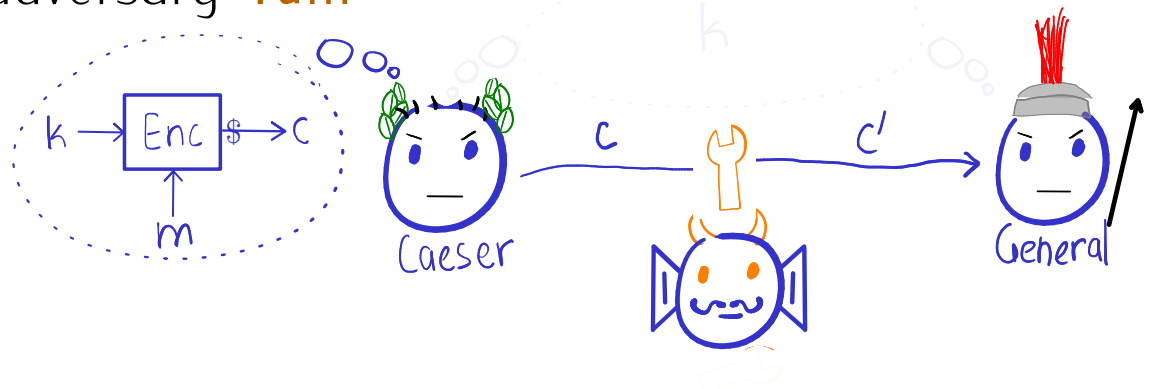
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caeser's sends to the General (integrity)
     - All schemes we've seen so far (OTP, computational OTP, CPA-secret construction) are malleable!
     - Why not use error-detecting codes?

What Exactly Is the Security Goal?...

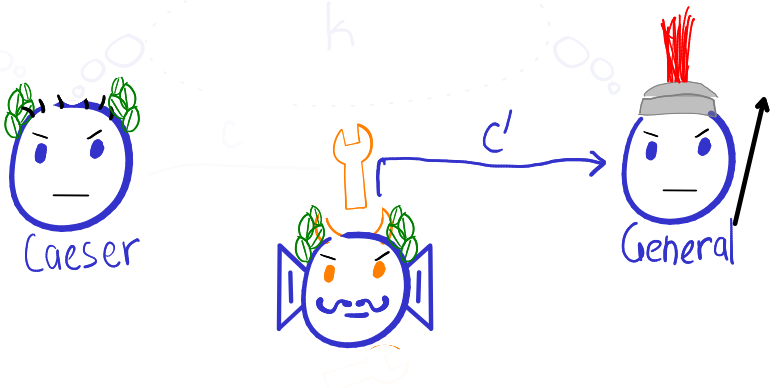
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caeser's sends to the General (integrity)
     - All schemes we've seen so far (OTP, computational OTP, CPA-secret construction) are malleable!
     - Why not use error-detecting codes? Doesn't stand up to adversarial errors (only stochastic errors).

What Exactly Is the Security Goal?...

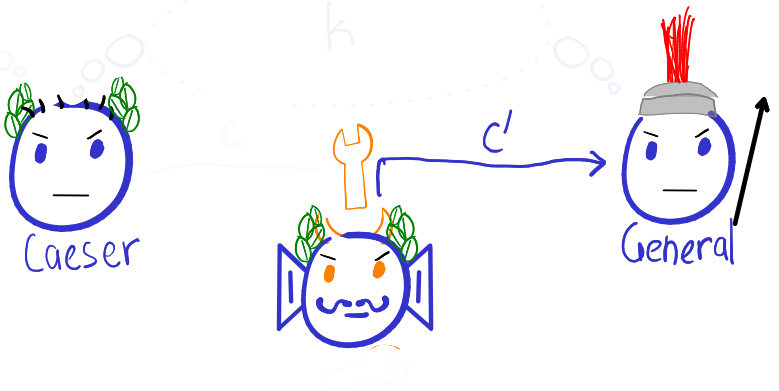
- The setting: Caeser and his general share a key $k \in \{0,1\}^n$ and want to secretly communicate in presence of an *active* adversary Tam



- What can Tam do?
  1. Modify what Caeser's sends to the General (integrity)
     - All schemes we've seen so far (OTP, computational OTP, CPA-secret construction) are malleable!
     - Why not use error-detecting codes? Doesn't stand up to adversarial errors (only stochastic errors).
  2. Try to impersonate Caeser by injecting messages (authenticity)

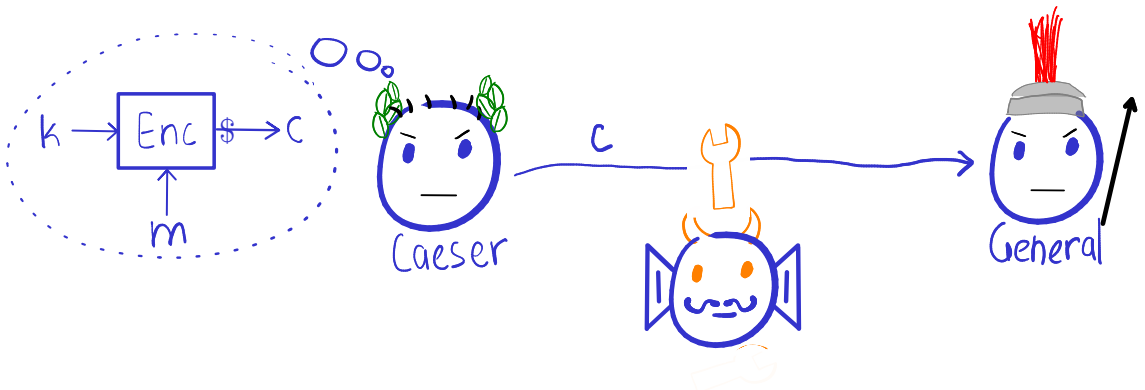
What Exactly Is the Security Goal?...

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



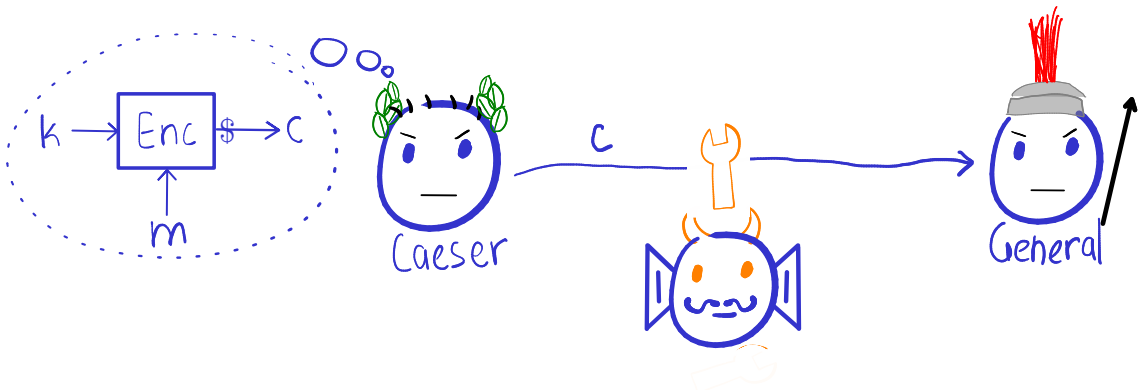
- What can Tam do?
  1. Modify what Caeser's sends to the General (integrity)
     - All schemes we've seen so far (OTP, computational OTP, CPA-secret construction) are malleable!
     - Why not use error-detecting codes? Doesn't stand up to adversarial errors (only stochastic errors).
  2. Try to impersonate Caeser by injecting messages (authenticity)
- We cannot prevent this: the hope is to *detect* when it happens

What Exactly Is the Security Goal?...



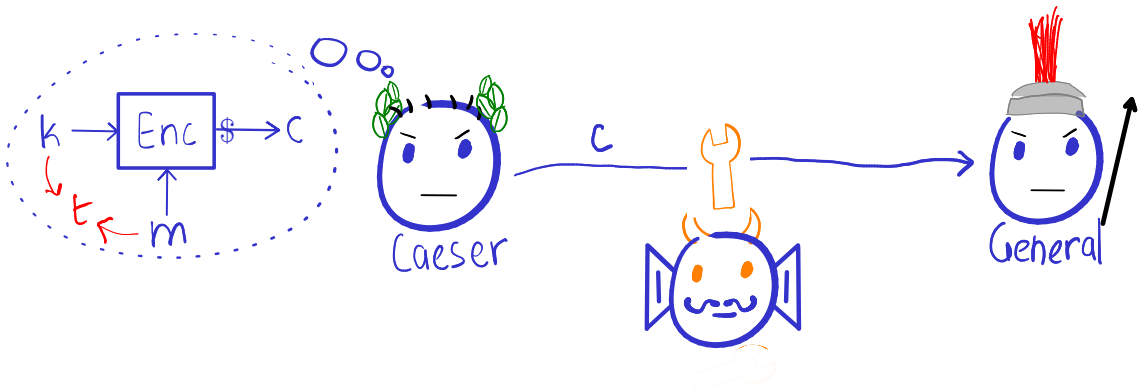
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



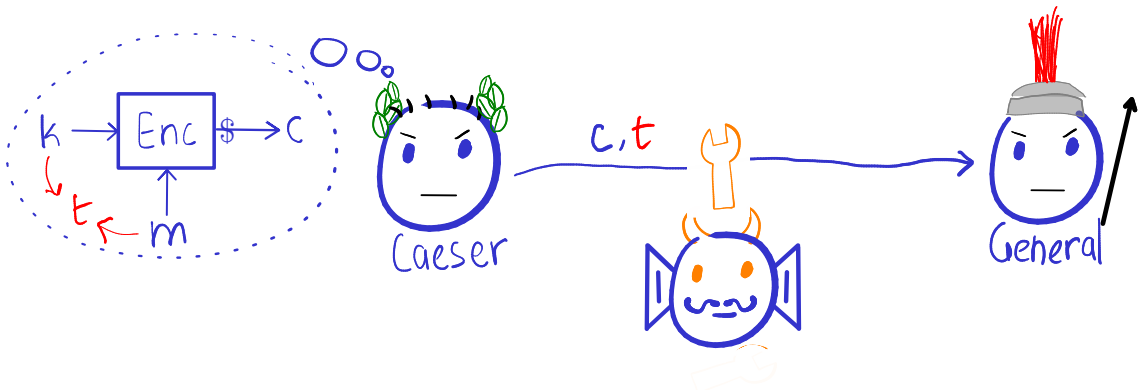
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



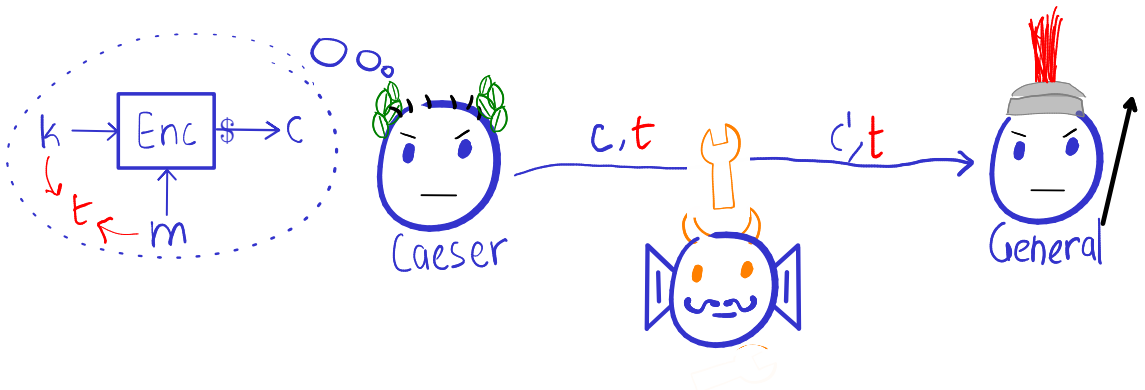
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



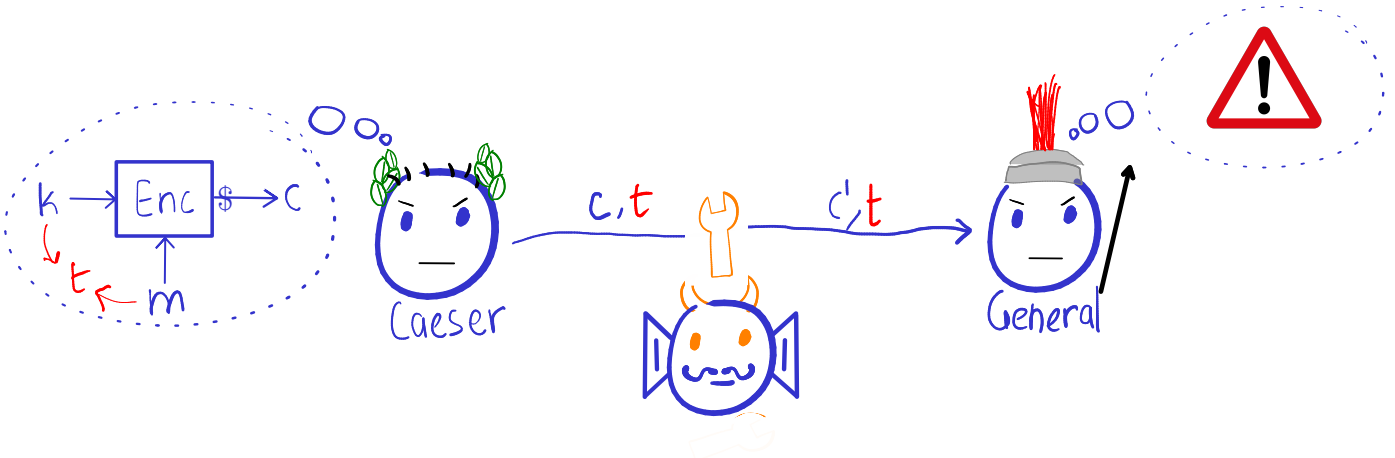
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



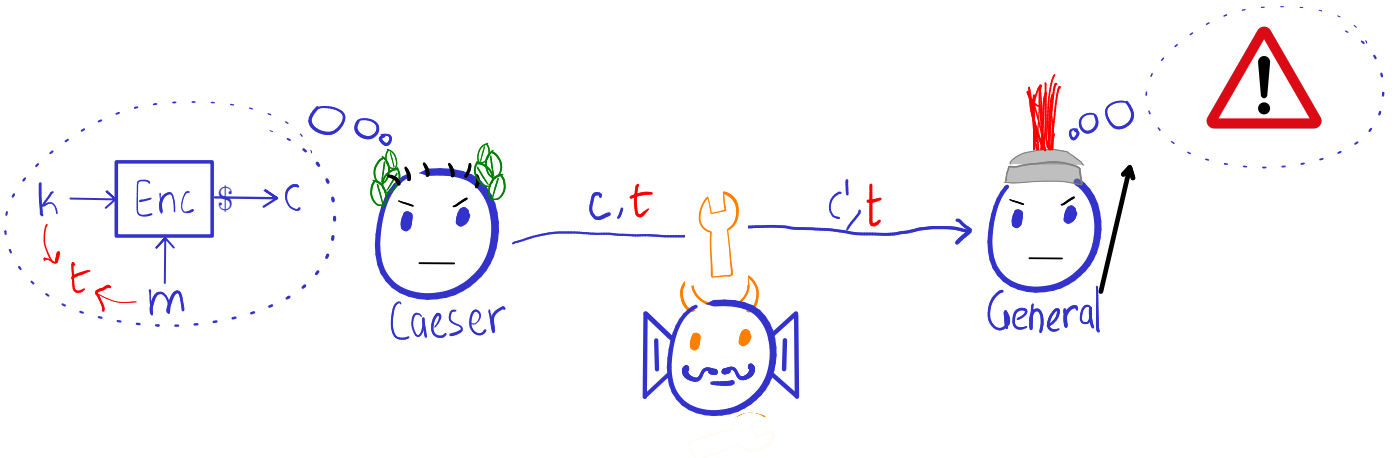
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



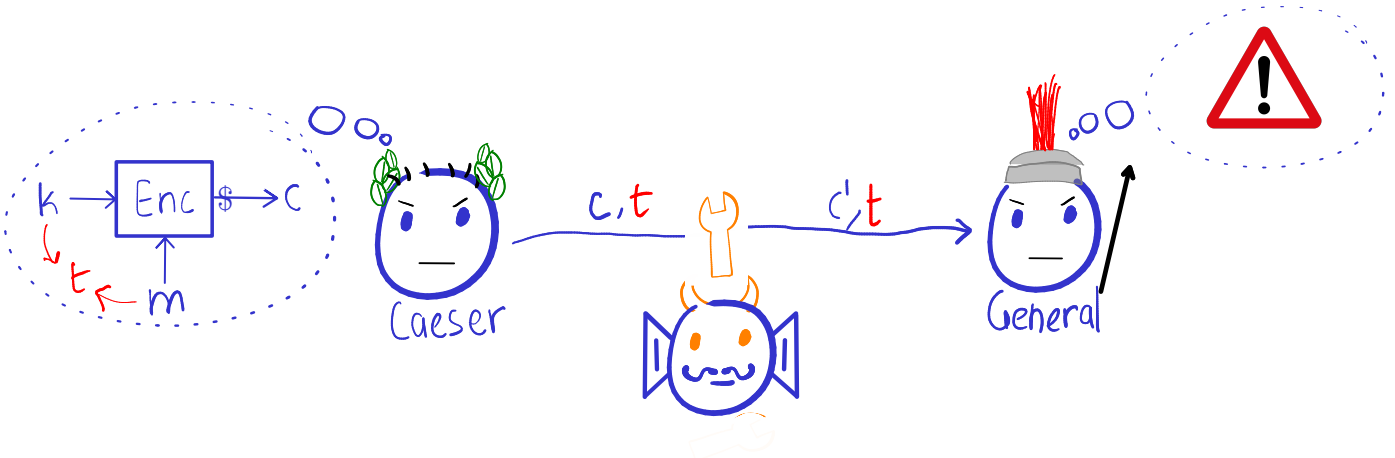
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext

What Exactly Is the Security Goal?...



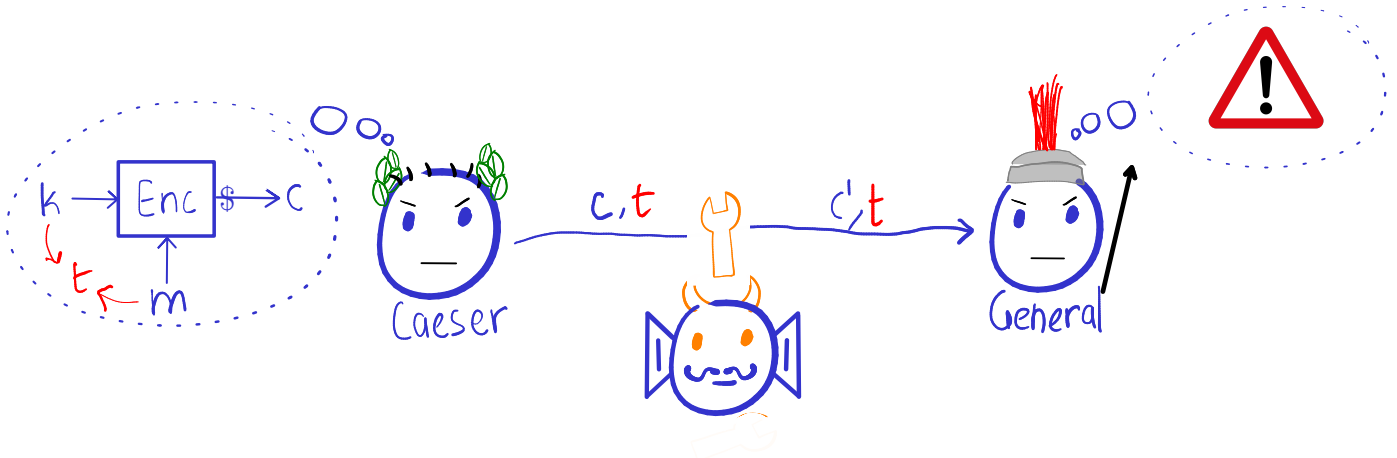
- How do we ensure integrity and authenticity?
  - Append "additional information" t with the ciphertext
  - Message-authentication code (MAC)
    - Think of it as "cryptographic" version of error detection!

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

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.
  - Later: MAC + *any* CPA-secret SKE → "secret communication" against Tam

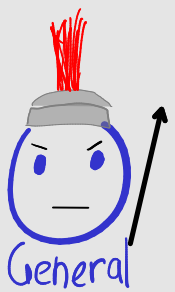
Plan for this Lecture

1. Message-Authentication Code (MAC)
2. Constructing MACs
3. Chosen-Ciphertext Attack

Syntax of Message-Authentication Code (MAC)

Defintion 1 (Message-Authentication Code (MAC))

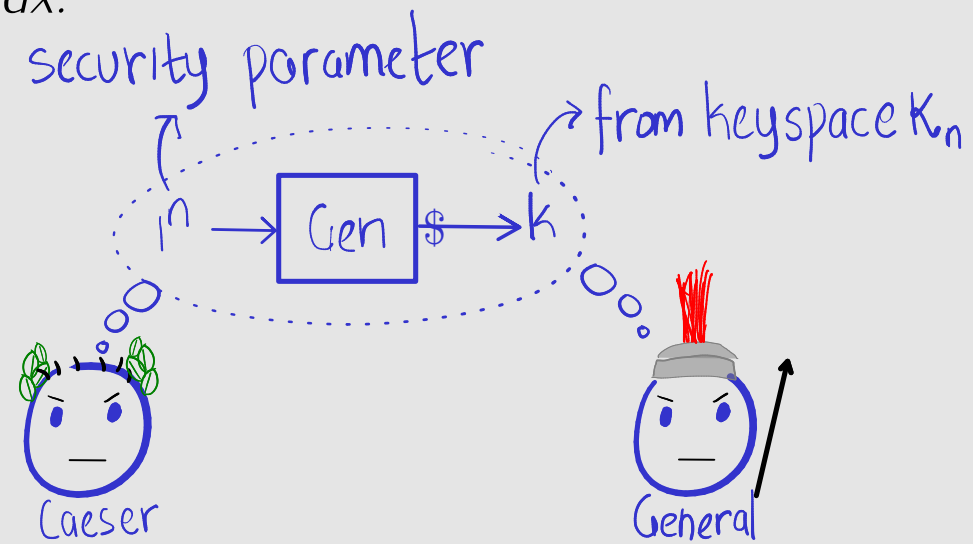
An MAC M *is a triple of efficient algorithms* (Gen, Tag, Ver) *with the following syntax:*



Syntax of Message-Authentication Code (MAC)

Defintion 1 (Message-Authentication Code (MAC))

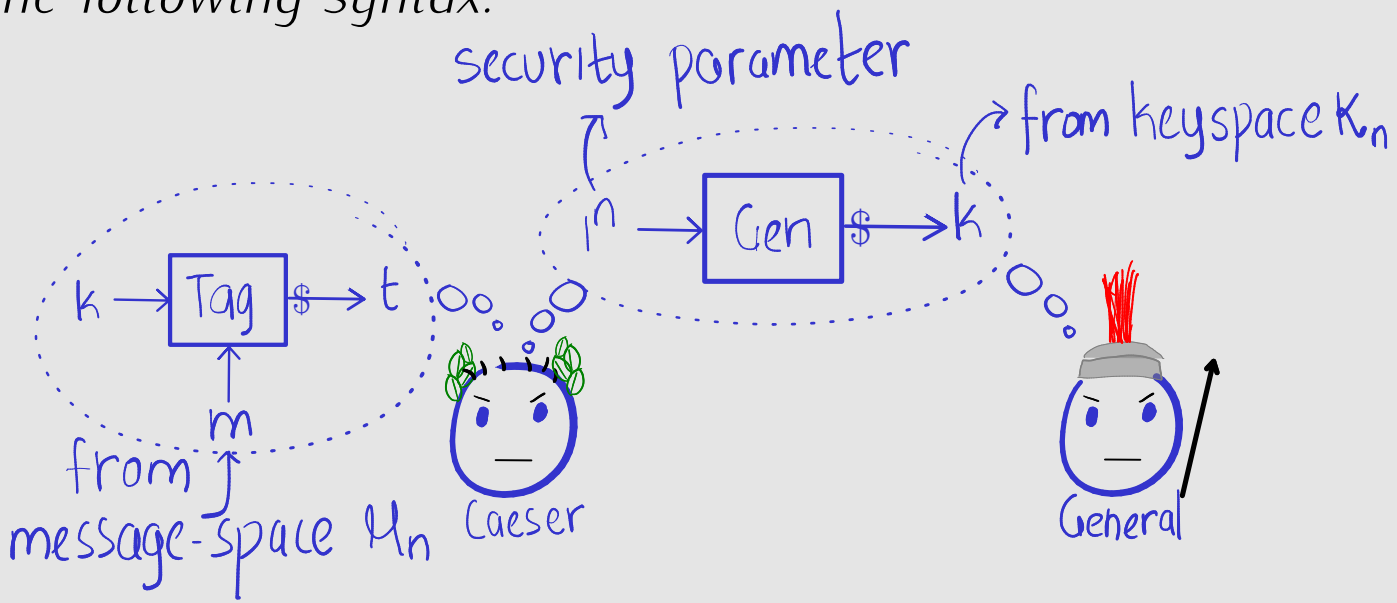
An MAC M *is a triple of efficient algorithms* (Gen, Tag, Ver) *with the following syntax:*



Syntax of Message-Authentication Code (MAC)

Defintion 1 (Message-Authentication Code (MAC))

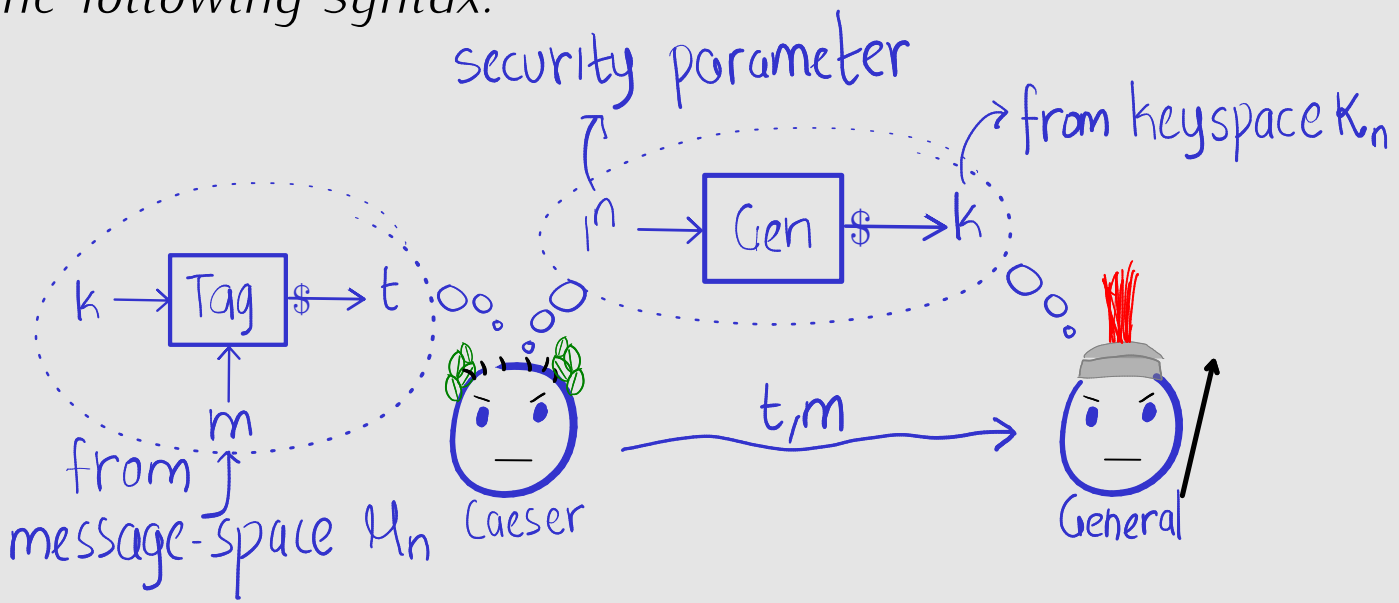
An MAC M *is a triple of efficient algorithms* (Gen, Tag, Ver) *with the following syntax:*



Syntax of Message-Authentication Code (MAC)

Defintion 1 (Message-Authentication Code (MAC))

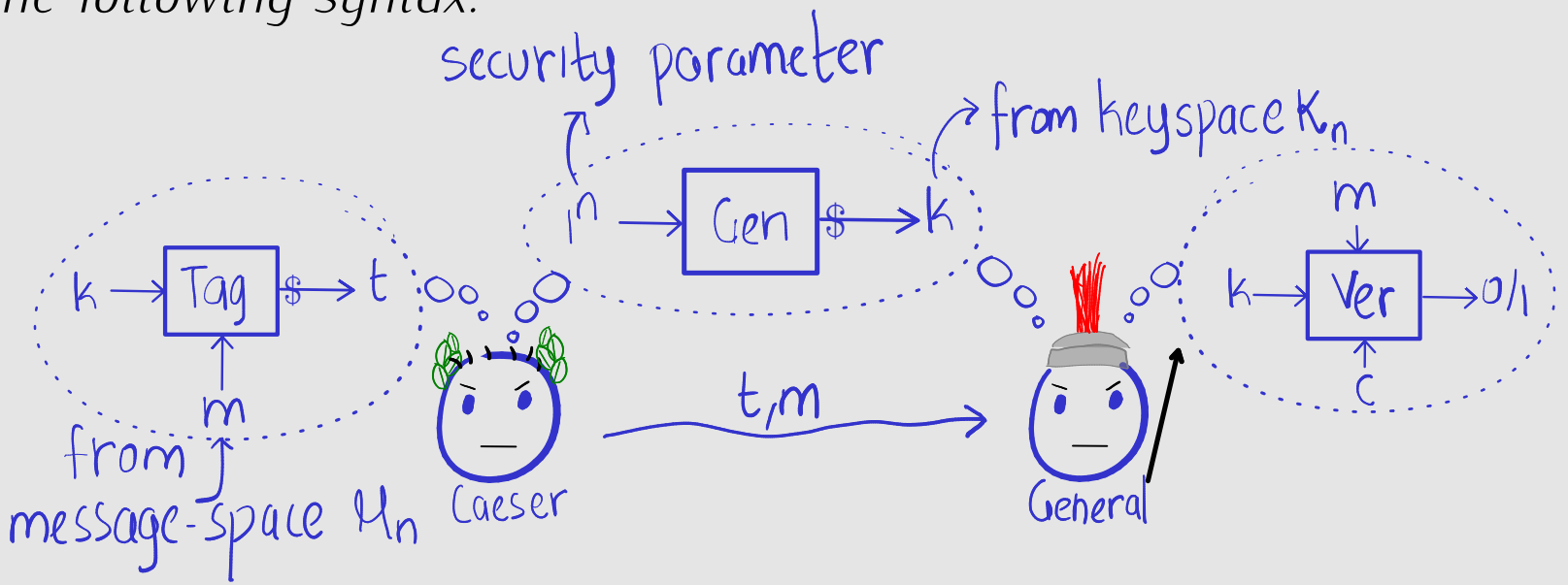
An MAC M *is a triple of efficient algorithms* (Gen, Tag, Ver) *with the following syntax:*



Syntax of Message-Authentication Code (MAC)

Defintion 1 (Message-Authentication Code (MAC))

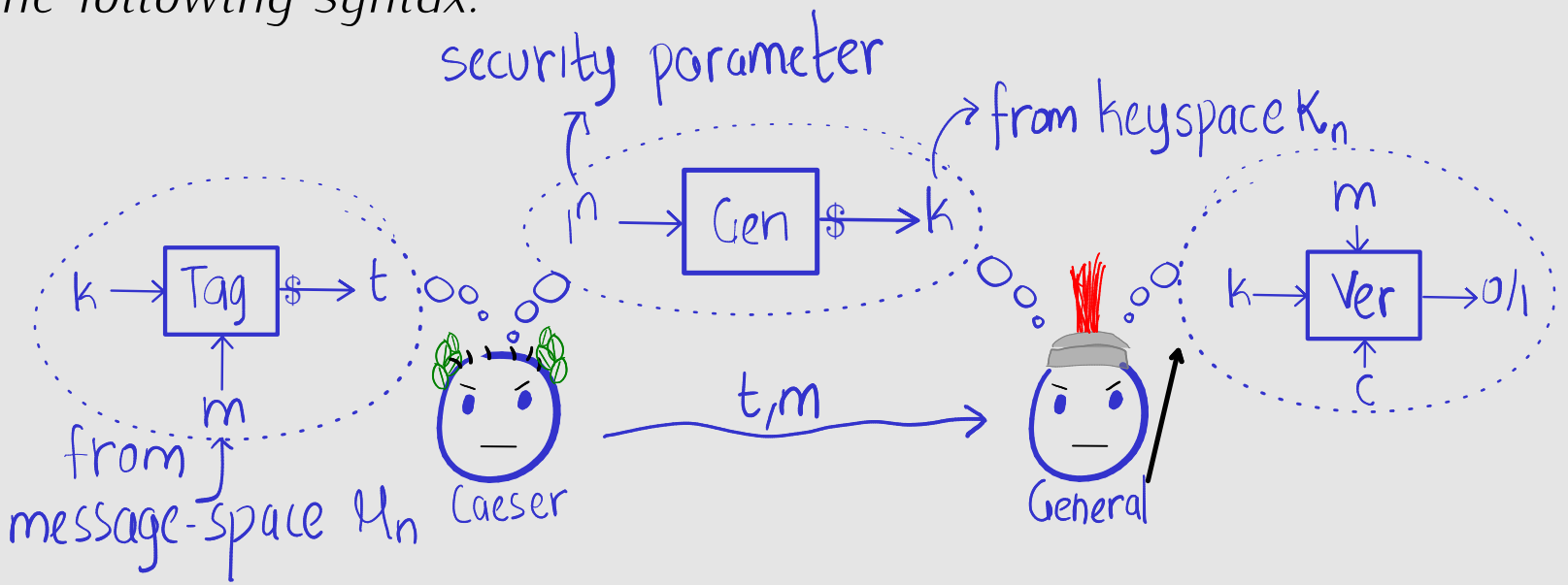
An MAC M *is a triple of efficient algorithms* (Gen, Tag, Ver) *with the following syntax:*



Syntax of Message-Authentication Code (MAC)

Defintion 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* $m \in \mathcal{M}_n$*,*

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

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 a valid *new* tag from previously-seen tags...

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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 a 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 a 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 a 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

Defintion 2 (EU-CMA)

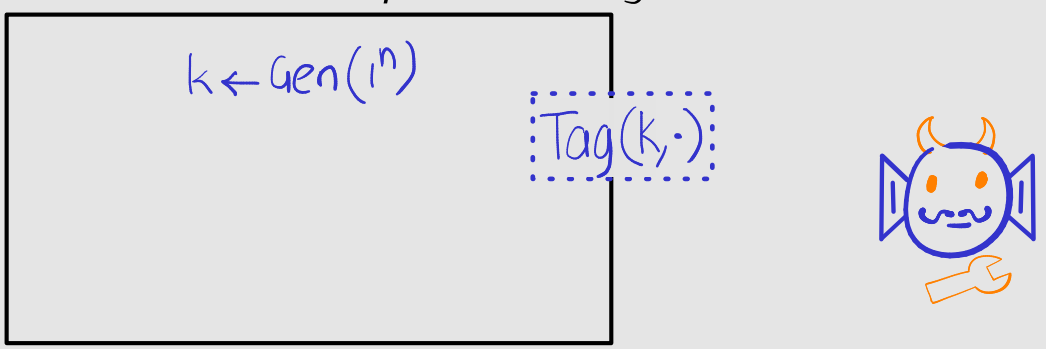
A MAC $M = (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ *is* (ϵ, q)*-EU-CMA secure if no PPT tampering adversary* Tam *that makes at most* q *queries can break* M *as below with probability* ϵ*.*

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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

Defintion 2 (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* ϵ*.*

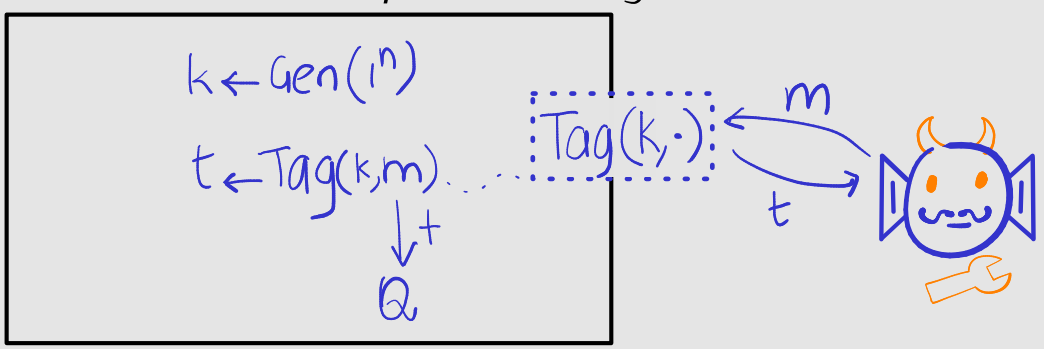


How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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

Defintion 2 (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* ϵ*.*

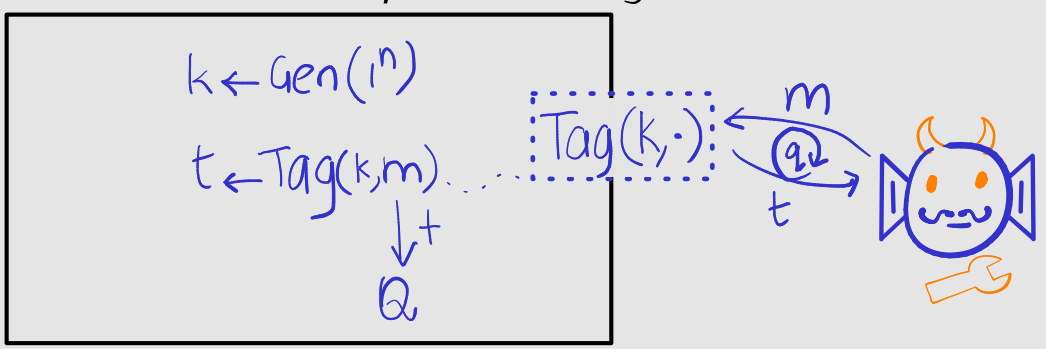


How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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

Defintion 2 (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* ϵ.



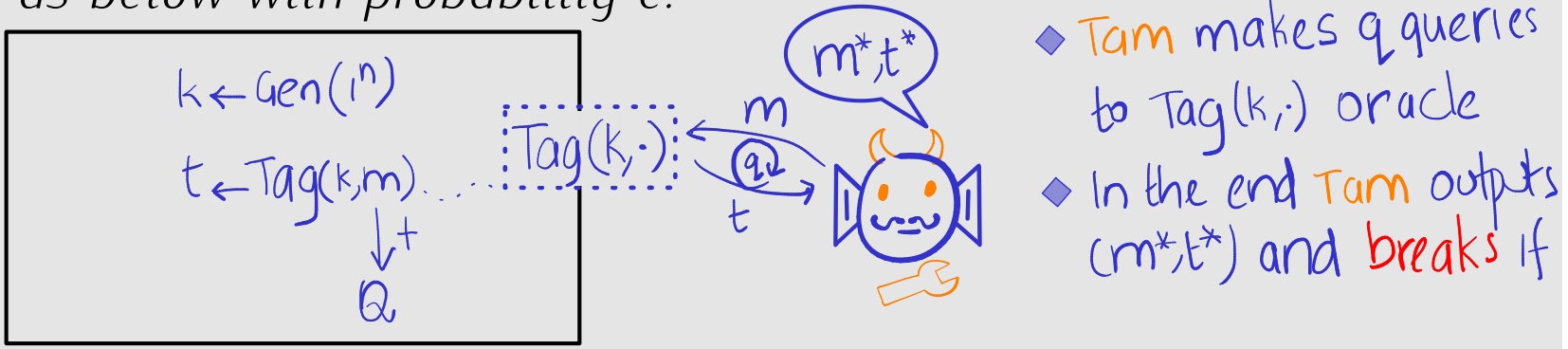
- Tam makes q queries to Tag(k,·) oracle

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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

Defintion 2 (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* ϵ.



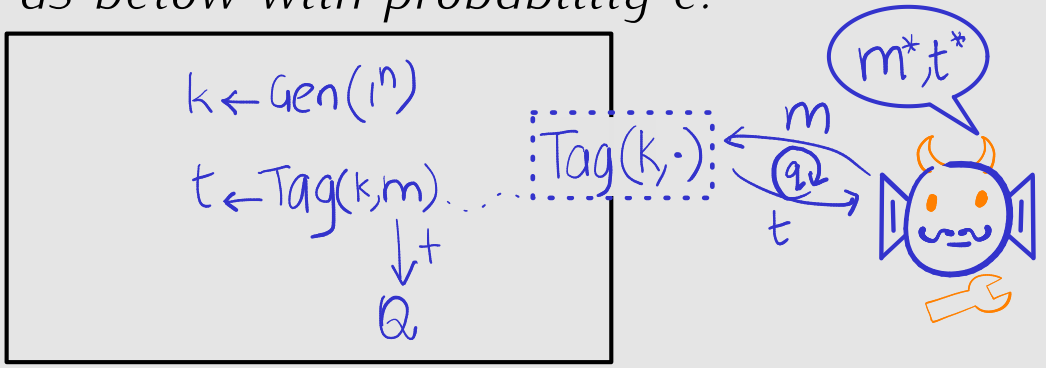
- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if

How to Define Security?...

- Intuitively, what are the security requirements?
  - Tam must not be able to forge a 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

Defintion 2 (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* ϵ.

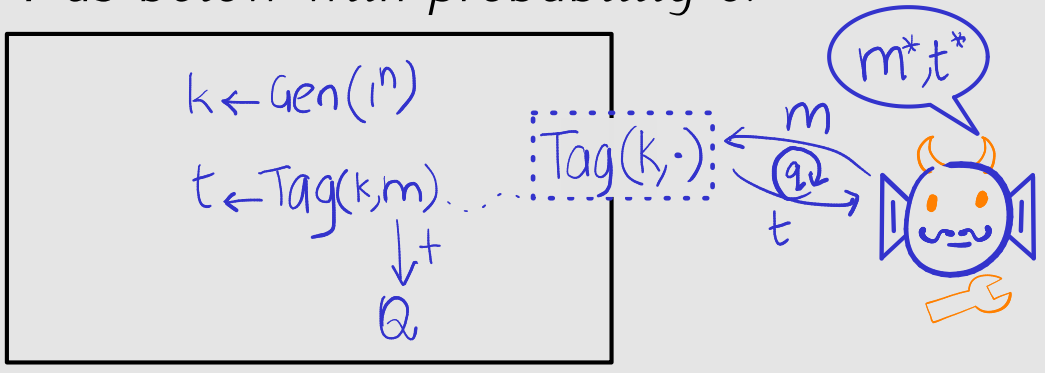


- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) m*∉Q
  ii) Ver(k,t*,m*)=1

How to Define Security?...

Defintion 2 (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* ϵ.

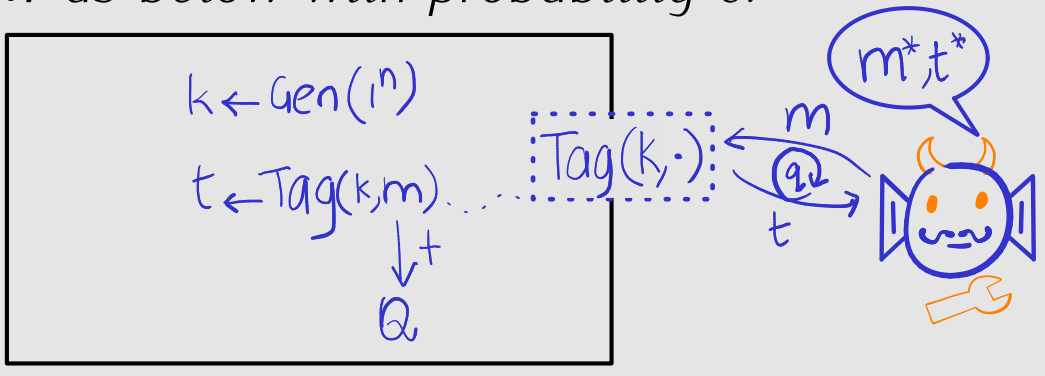


- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) $m^* \notin Q$
  ii) $\mathrm{Ver}(k, t^*, m^*) = 1$

How to Define Security?...

Defintion 2 (EU-CMA) Typically → negligible / polynomial "information-theoretic"

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* ϵ.

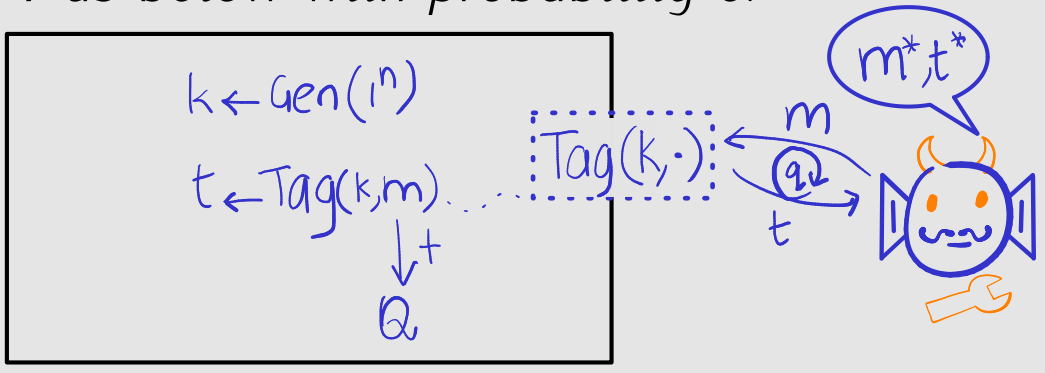


- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) m* ∉ Q
  ii) Ver(k,t*,m*)=1

How to Define Security?...

Defintion 2 (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* ϵ.



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) m* ∉ Q
  ii) Ver(k,t*,m*)=1

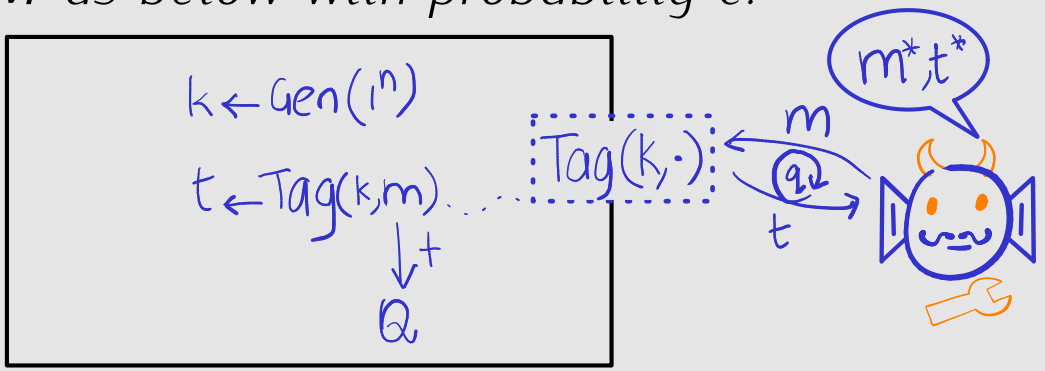
❓ MAC or not?

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?...

Defintion 2 (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* ϵ.



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) m* ∉ Q
  ii) Ver(k,t*,m*)=1

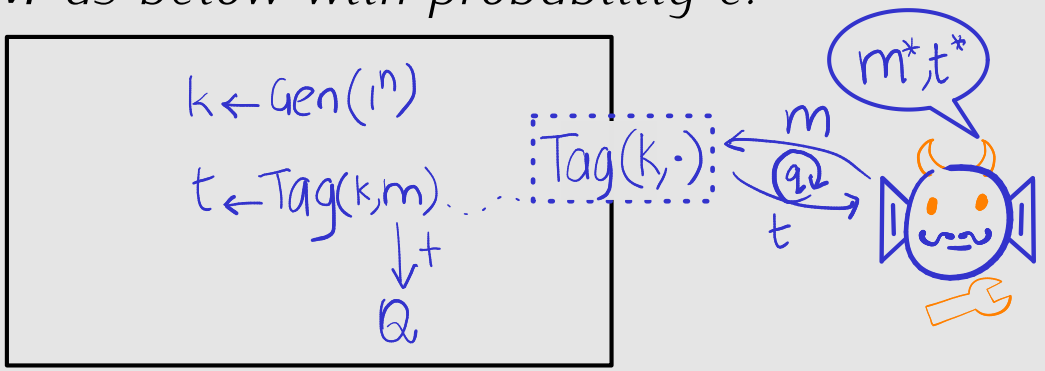
MAC or not?

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
   - $\mathsf{Tag}'(k, m) := t0$, where $t \leftarrow \mathsf{Tag}(k, m)$
   - $\mathsf{Ver}'(k, tb, m) := \mathsf{Ver}(k, t, m)$

How to Define Security?...

Defintion 2 (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* ϵ.



- Tam makes q queries to Tag(k,·) oracle
- In the end Tam outputs (m*,t*) and breaks if
  i) m* ∉ Q
  ii) 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
   - $\mathsf{Tag}'(k, m) := t0$, where $t \leftarrow \mathsf{Tag}(k, m)$
   - $\mathsf{Ver}'(k, tb, m) := \mathsf{Ver}(k, t, m)$

Plan for this Lecture

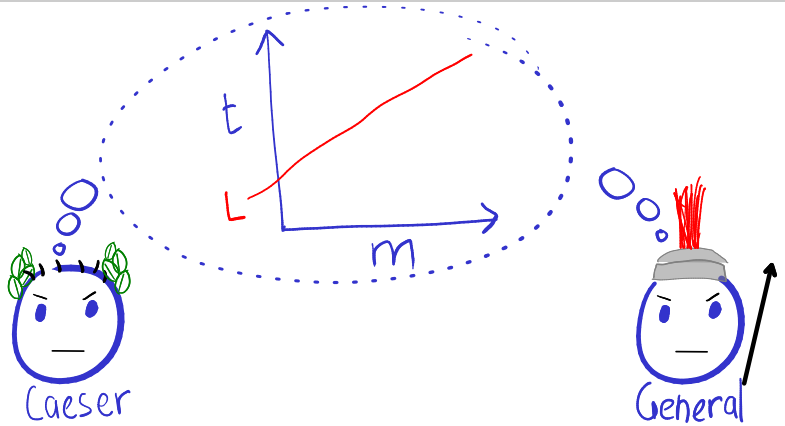
1 Message-Authentication Code (MAC)

2 Constructing MACs

3 Chosen-Ciphertext Attack

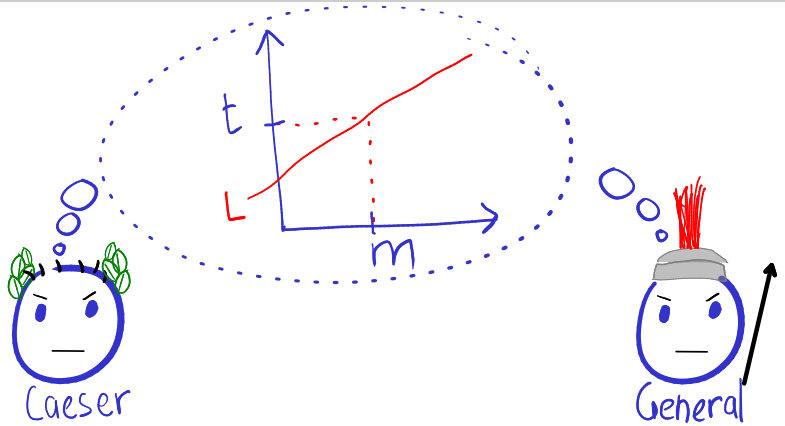
Let's Start with a One-Time MAC ($q = 1$)

Let's Start with a One-Time MAC ($q = 1$)



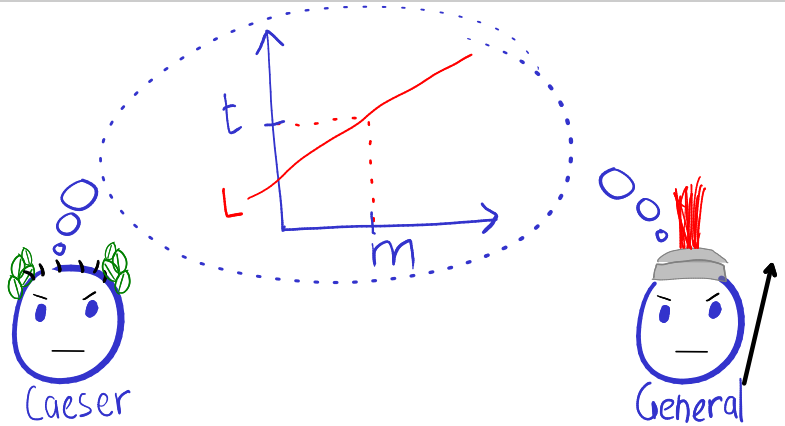
- Caeser & General "share a line" L; MAC of m is its evaluation

Let's Start with a One-Time MAC ($q = 1$)



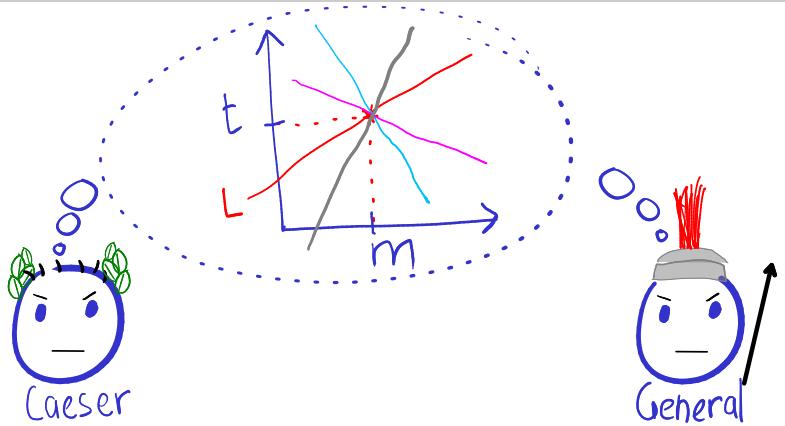
- Caeser & General "share a line" L; MAC of m is its evaluation

Let's Start with a One-Time MAC ($q = 1$)



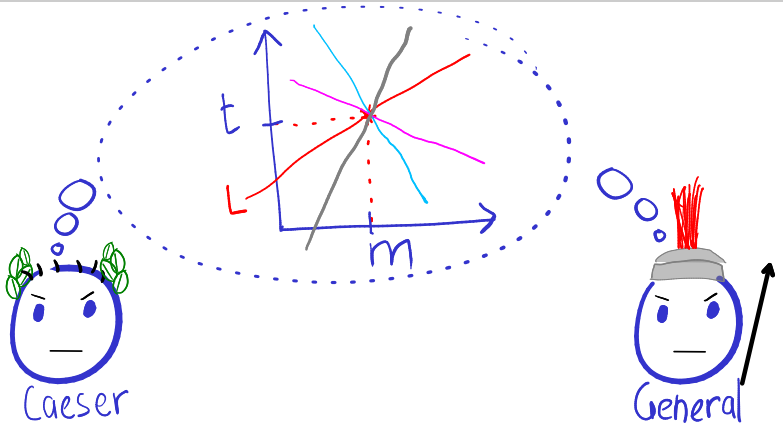
- Caeser & General "share a line" L; MAC of m is its evaluation
- Why is this a one-time MAC? Given (m, t) line is still hidden

Let's Start with a One-Time MAC ($q = 1$)



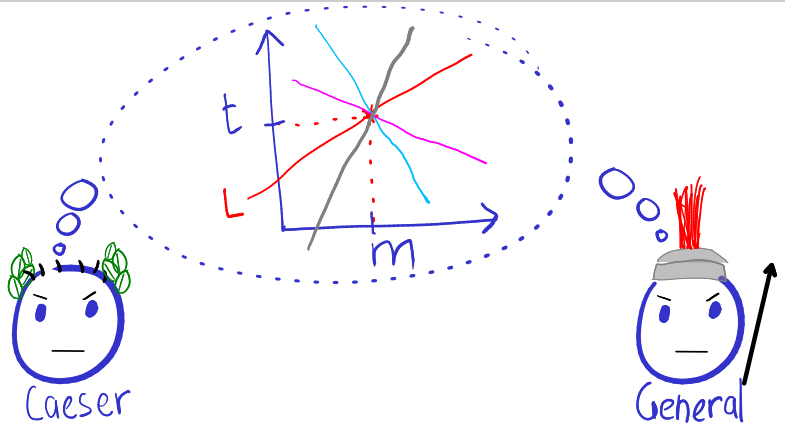
- Caeser & General "share a line" L; MAC of m is its evaluation
- Why is this a one-time MAC? Given (m^*, t) line is still hidden

Let's Start with a One-Time MAC ($q = 1$)



- Caeser & General "share a line" L; MAC of m is its evaluation
- Why is this a one-time MAC? Given (m^*, t) line is still hidden
- Abstraction: *pairwise-independent (PI) hash function*
  - Intuitively: behaves like a random function as long as it is evaluated on *at most two points*

Let's Start with a One-Time MAC ($q = 1$)



- Caeser & General "share a line" L; MAC of m is its evaluation
- Why is this a one-time MAC? Given (m^*, t) line is still hidden
- Abstraction: *pairwise-independent (PI) hash function*
  - Intuitively: behaves like a random function as long as it is evaluated on *at most two points*

Defintion 3

A function $H : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a pairwise-independent (PI) hash function if for all $m \neq m' \in \mathcal{M}$ and all $t, t' \in \mathcal{T}$,

$$\Pr_{k \leftarrow \mathcal{K}}[H_k(m) = t, H_k(m') = t'] = 1/|\mathcal{T}|^2$$

Let's Start with a One-Time MAC ($q = 1$)

Construction 1 (One-Time MAC from PI hash function)

- $\mathsf{Gen}(1^n)$: *sample key* $k \leftarrow \mathcal{K}$
- $\mathsf{Tag}(k, m)$: *output* $t := H(k, m)$
- $\mathsf{Ver}(k, t, m)$: *accept iff* $H(k, m) = t$

Let's Start with a One-Time MAC ($q = 1$)

Construction 1 (One-Time MAC from PI hash function)

- $\mathsf{Gen}(1^n)$: *sample key* $k \leftarrow \mathcal{K}$
- $\mathsf{Tag}(k, m)$: *output* $t := H(k, m)$
- $\mathsf{Ver}(k, t, m)$: *accept iff* $H(k, m) = t$

Theorem 1 (Construction 1 is information-theoretically secure)

If H *is a PI hash function then Construction 1 is* $(1, 1/|\mathcal{T}|)$*-EU-CMA-secure against any* Tam.

Proof.

For m output by Tam

$$\Pr_{k \leftarrow K_n}[H(k, m^*) = t^*]$$

where $t := H(k, m)$ & $(m^*, t^*) := \mathsf{Tam}(t)$

□

Let's Start with a One-Time MAC ($q = 1$)

Construction 1 (One-Time MAC from PI hash function)

- $\mathsf{Gen}(1^n)$: *sample key* $k \leftarrow \mathcal{K}$
- $\mathsf{Tag}(k, m)$: *output* $t := H(k, m)$
- $\mathsf{Ver}(k, t, m)$: *accept iff* $H(k, m) = t$

Theorem 1 (Construction 1 is information-theoretically secure)

If H *is a PI hash function then Construction 1 is* $(1, 1/|\mathcal{T}|)$*-EU-CMA-secure against any* Tam.

Proof.

For m output by Tam

$$\Pr_{k \leftarrow K_n}[H(k,m^*)=t^*] = \sum_t \Pr_{k \leftarrow K_n}[H(k,m^*)=t^* \wedge H(k,m)=t]$$

where $t := H(k,m)$ & $(m^*, t^*) := \mathsf{Tam}(t)$ □

Let's Start with a One-Time MAC ($q = 1$)

Construction 1 (One-Time MAC from PI hash function)

- $\mathsf{Gen}(1^n)$: *sample key* $k \leftarrow \mathcal{K}$
- $\mathsf{Tag}(k, m)$: *output* $t := H(k, m)$
- $\mathsf{Ver}(k, t, m)$: *accept iff* $H(k, m) = t$

Theorem 1 (Construction 1 is information-theoretically secure)

If H *is a PI hash function then Construction 1 is* $(1, 1/|\mathcal{T}|)$*-EU-CMA-secure against any* Tam.

Proof.

For m output by Tam

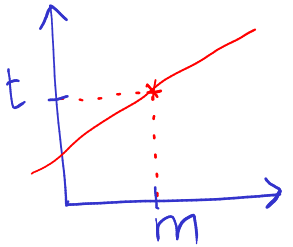
$$\Pr_{k \leftarrow K_n}[H(k, m^*) = t^*] \overset{\text{PI}}{=} \sum_t 1/|\mathcal{T}|^2 = 1/|\mathcal{T}|$$

where $t := H(k, m)$ & $(m^*, t^*) := \text{Tam}(t)$

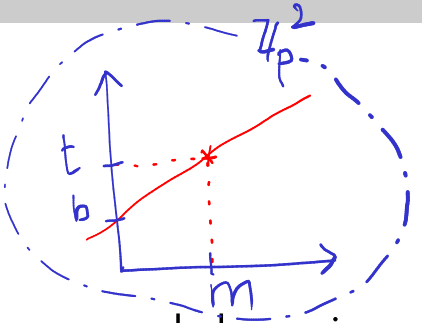
□

How to Construct Pairwise-Independent Hash?

- Recall the informal construction:
  - Key defines a "line" L
  - Hash value on m is its evaluation on L



How to Construct Pairwise-Independent Hash?



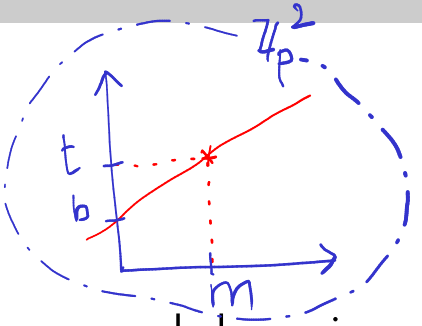
- Recall the informal construction:
  - Key defines a "line" L
  - Hash value on m is its evaluation on L
- Implement this over $(\mathbb{Z}_p, +)$, the additive group modulo prime p
  - Has some "nice properties" of real plane

Construction 2 (PI hash function $H : \mathbb{Z}_p^2 \times \mathbb{Z}_p \to \mathbb{Z}_p$)

- $\mathcal{K} := \mathbb{Z}_p^2$: *a key* $k = (a, b)$ *defines a line* $y = ax + b \bmod p$
- $\mathcal{M} := \mathbb{Z}_p$: *a message* m *is an element of* $\mathbb{Z}_p$
- *The hash* t *of a message* $m \in \mathbb{Z}_p$ *using a key* $k = (a, b) \in \mathbb{Z}_p^2$ *is*

$$t = H_{a,b}(m) := am + b \bmod p$$

How to Construct Pairwise-Independent Hash?



- Recall the informal construction:
  - Key defines a "line" L
  - Hash value on m is its evaluation on L
- Implement this over $(\mathbb{Z}_p, +)$, the additive group modulo prime p
  - Has some "nice properties" of real plane

Construction 2 (PI hash function $H : \mathbb{Z}_p^2 \times \mathbb{Z}_p \to \mathbb{Z}_p$)

- $\mathcal{K} := \mathbb{Z}_p^2$: *a key* $k = (a, b)$ *defines a line* $y = ax + b \bmod p$
- $\mathcal{M} := \mathbb{Z}_p$: *a message* m *is an element of* $\mathbb{Z}_p$
- *The hash* t *of a message* $m \in \mathbb{Z}_p$ *using a key* $k = (a, b) \in \mathbb{Z}_p^2$ *is*

$$t = H_{a,b}(m) := am + b \bmod p$$

Exercise 1

For any prime p*, show that Construction 2 is PI hash function.*

Extending to q-Time MAC, and More?

- Use q-wise independent hash function instead

Defintion 4

For $q \in \mathbb{N}$, a function $H : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a q-wise independent hash function if for all $m_1 \neq \ldots \neq m_q \in \mathcal{M}$ and all $t_1, \ldots, t_q \in \mathcal{T}$,

$$\Pr_{k \leftarrow \mathcal{K}}[H(k, m_1) = t_1, \cdots, H(k, m_q) = t_q = 1/|\mathcal{T}|^q$$

Extending to q-Time MAC, and More?

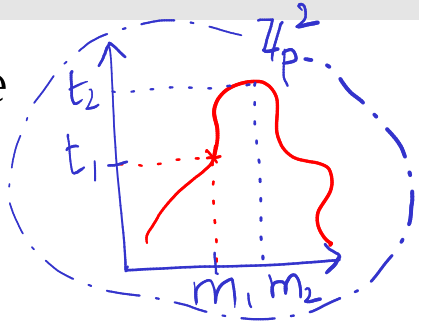
- Use q-wise independent hash function instead

Defintion 4

For $q \in \mathbb{N}$, a function $H : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a q-wise independent hash function if for all $m_1 \neq \ldots \neq m_q \in \mathcal{M}$ and all $t_1, \ldots, t_q \in \mathcal{T}$,

$$\Pr_{k \leftarrow \mathcal{K}}[H(k, m_1) = t_1, \cdots, H(k, m_q) = t_q = 1/|\mathcal{T}|^q$$

- Extend Construction 2: line $\to$ degree-qs curve
  - But key consists of q elements in $\mathbb{Z}_p$.
- This can be shown to be inherent!



Extending to q-Time MAC, and More?

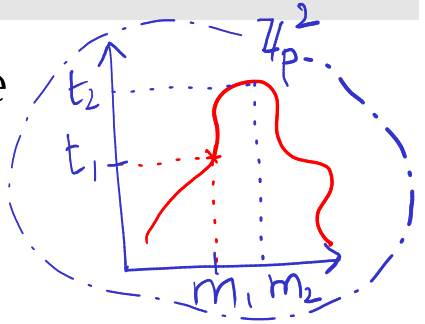
- Use q-wise independent hash function instead

Defintion 4

For $q \in \mathbb{N}$, a function $H : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is a q-wise independent hash function if for all $m_1 \neq \ldots \neq m_q \in \mathcal{M}$ and all $t_1, \ldots, t_q \in \mathcal{T}$,

$$\Pr_{k \leftarrow \mathcal{K}}[H(k, m_1) = t_1, \cdots, H(k, m_q) = t_q = 1/|\mathcal{T}|^q$$

- Extend Construction 2: line $\to$ degree-qs curve
  - But key consists of q elements in $\mathbb{Z}_p$.
- This can be shown to be inherent!



Exercise 2 (Key-size lower bound for information-theoretic MAC)

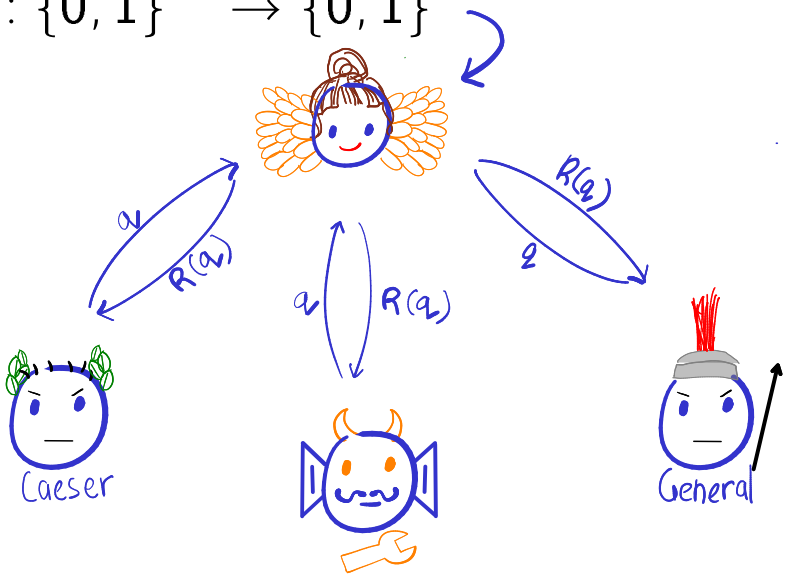
Show that any M that is (ϵ, q)-EU-CMA-secure against all Tams must have $\mathcal{K} \geq 1/\epsilon^{q+1}$.

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

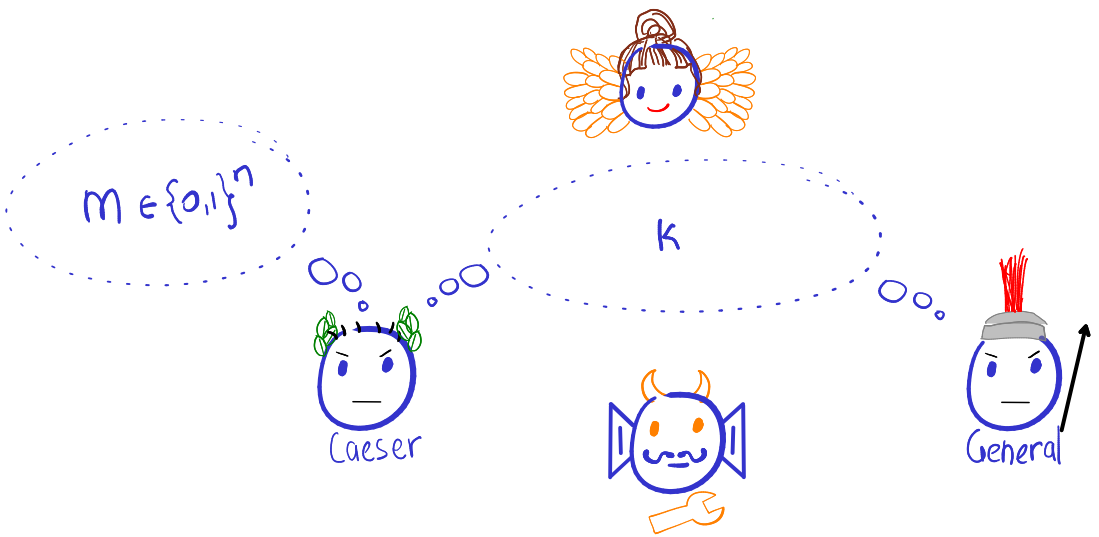
But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



But What if You're Given an Oracle in the Sky?

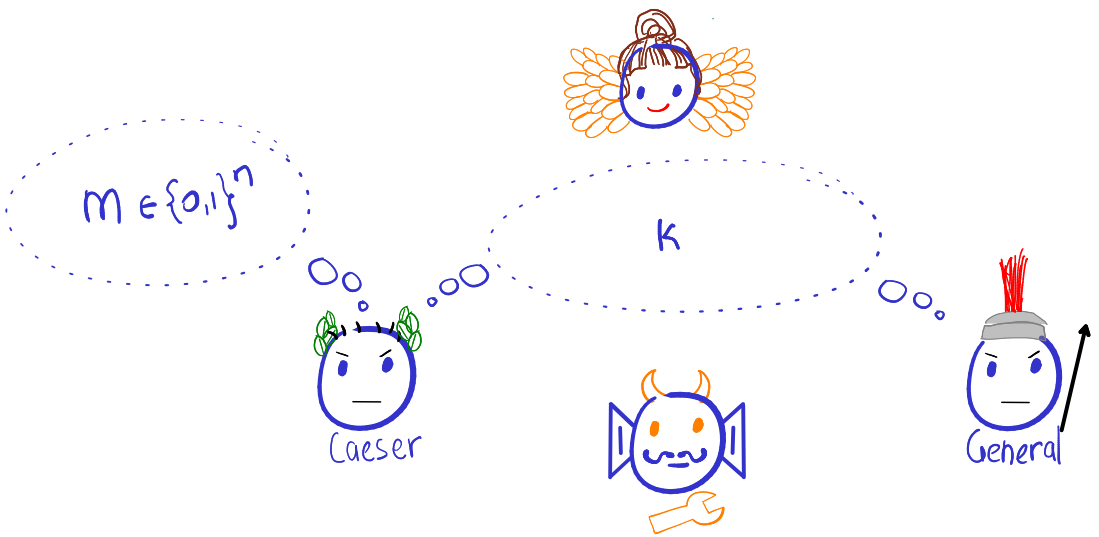
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

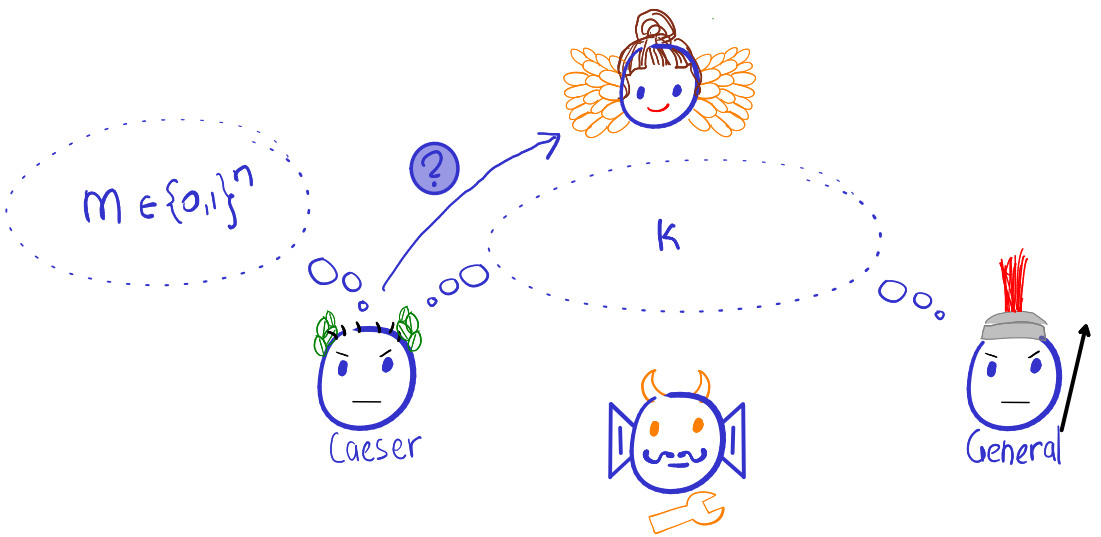


How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

Hint: R is 2^{2n}-wise independent!

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

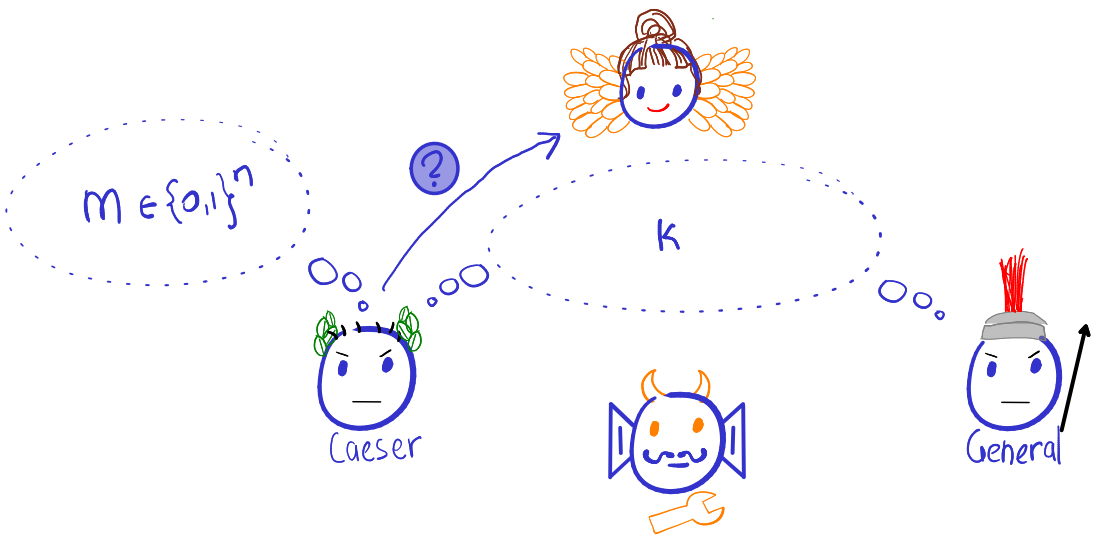


? How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

Hint: R is 2^{2n}-wise independent!

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$

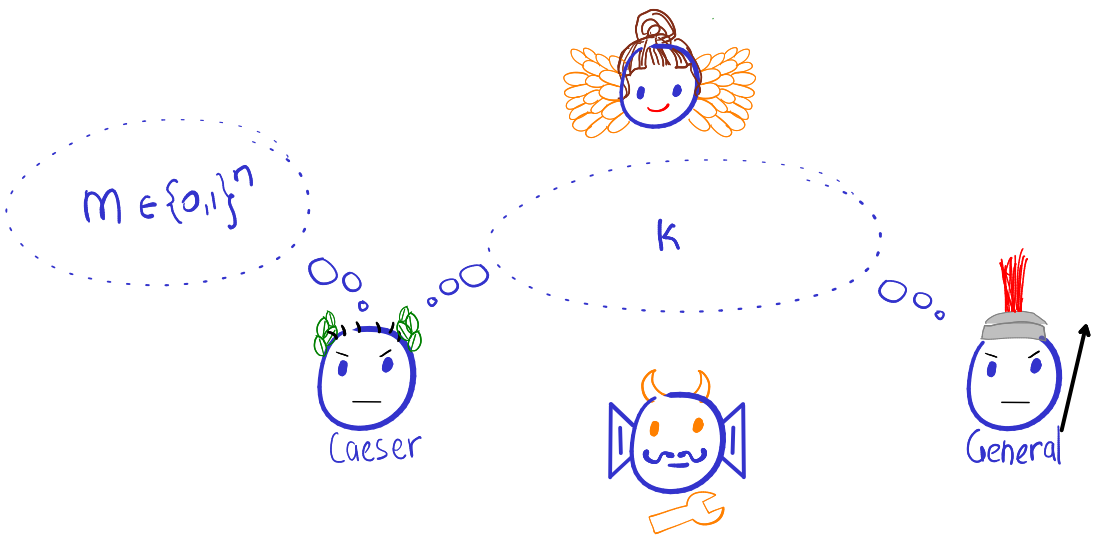


? How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

Hint: R is 2^{2n}-wise independent!

But What if You're Given an Oracle in the Sky?

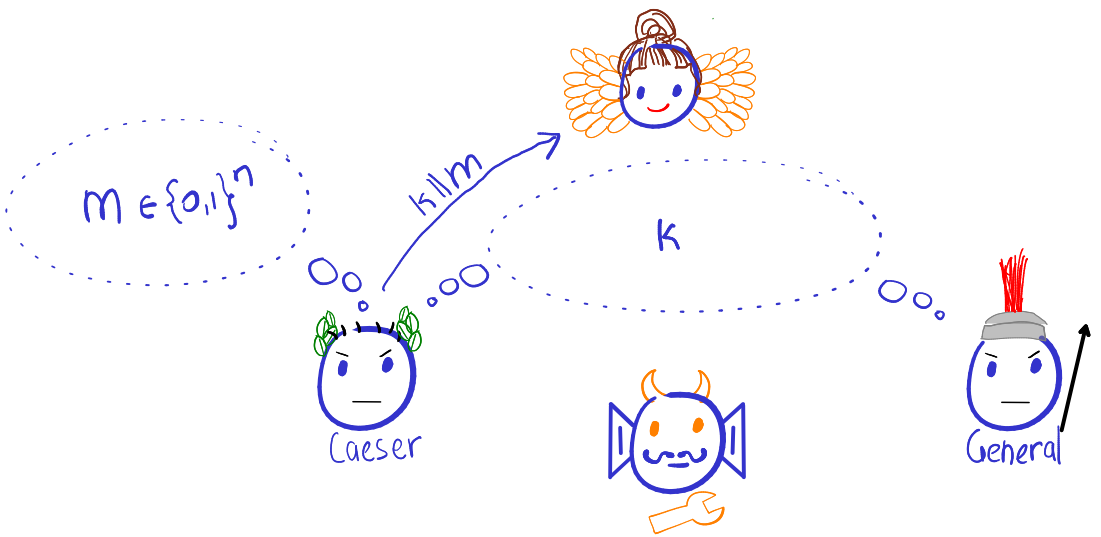
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



- How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?
  - Hint: R is 2^{2n}-wise independent!
  - Define tag for m as $R(k, m)$

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



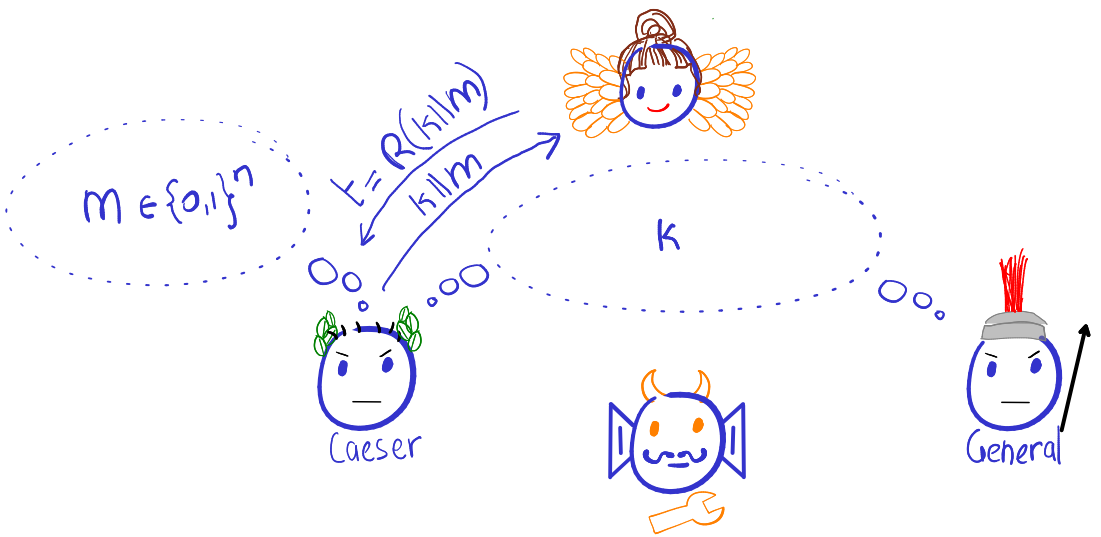
How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

Hint: R is 2^{2n}-wise independent!

- Define tag for m as $R(k, m)$

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



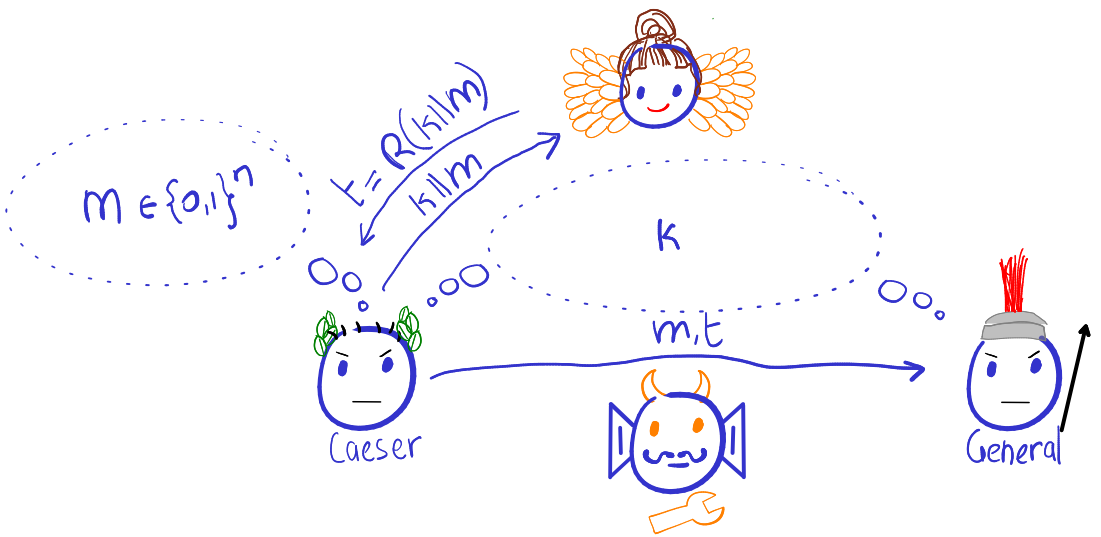
How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?

Hint: R is 2^{2n}-wise independent!

- Define tag for m as $R(k, m)$

But What if You're Given an Oracle in the Sky?

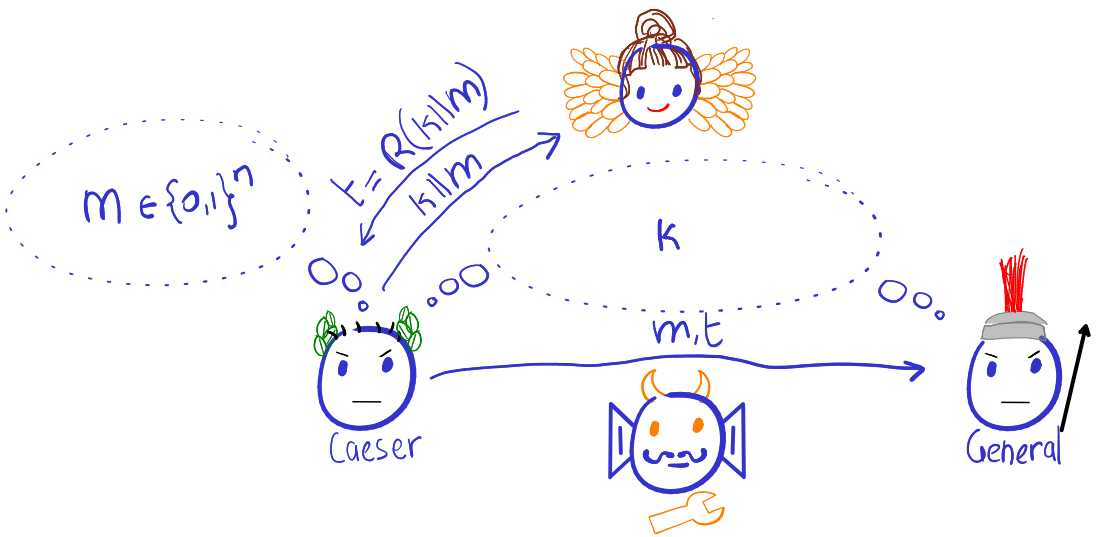
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



- How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?
  - Hint: R is 2^{2n}-wise independent!
  - Define tag for m as $R(k, m)$

But What if You're Given an Oracle in the Sky?

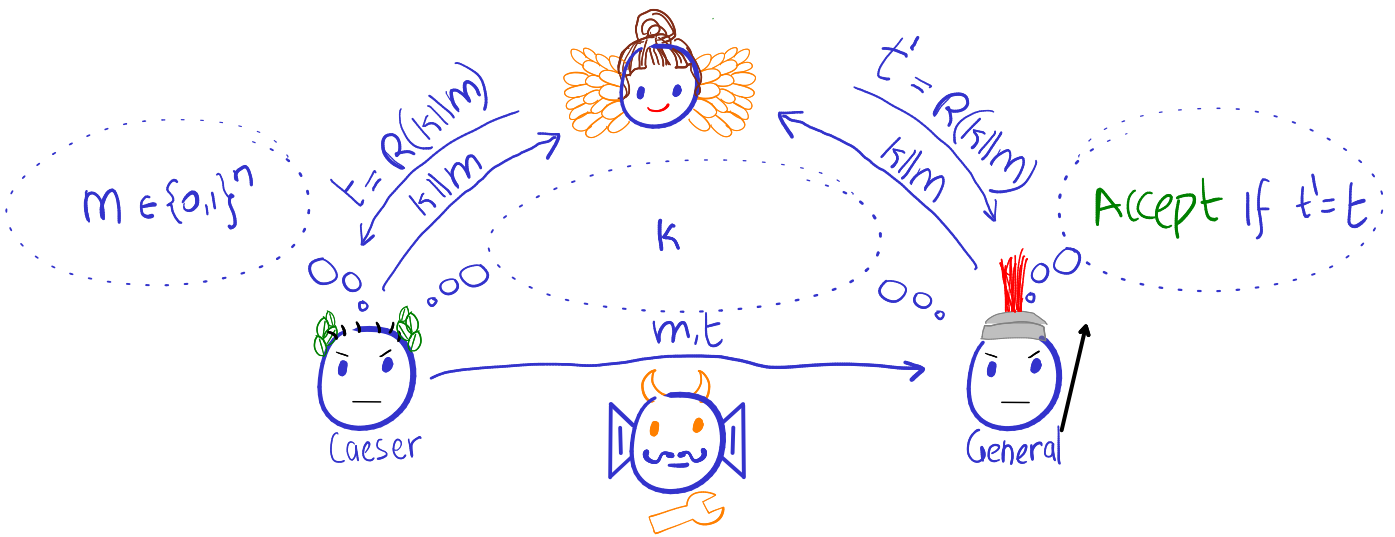
- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



- How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?
  - Hint: R is 2^{2n}-wise independent!
  - Define tag for m as $R(k, m)$

But What if You're Given an Oracle in the Sky?

- Setting:
  - Caeser and his general have shared a key $k \in \{0,1\}^n$
  - Everyone (including Tam) has access to a *random function oracle* $R : \{0,1\}^{2n} \to \{0,1\}^n$



- How will you construct a MAC for $\mathcal{M}_n = \{0,1\}^n$ using R?
  - Hint: R is 2^{2n}-wise independent!
  - Define tag for m as $R(k, m)$

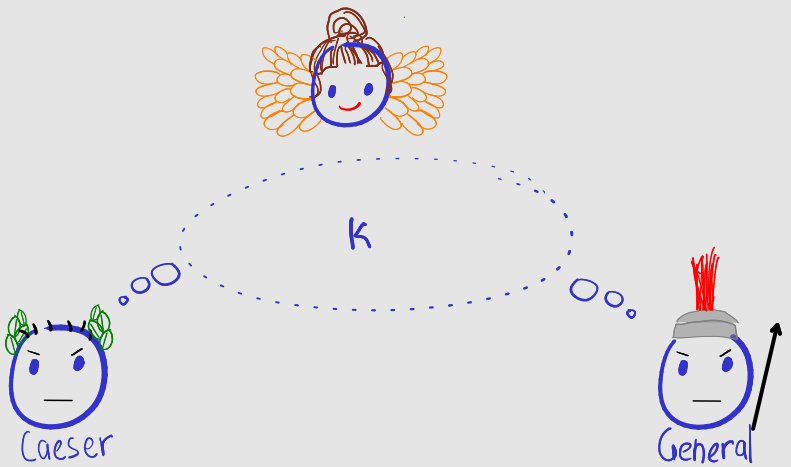
Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

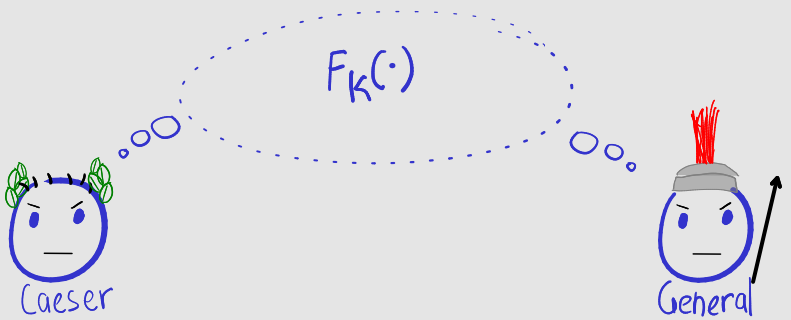
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

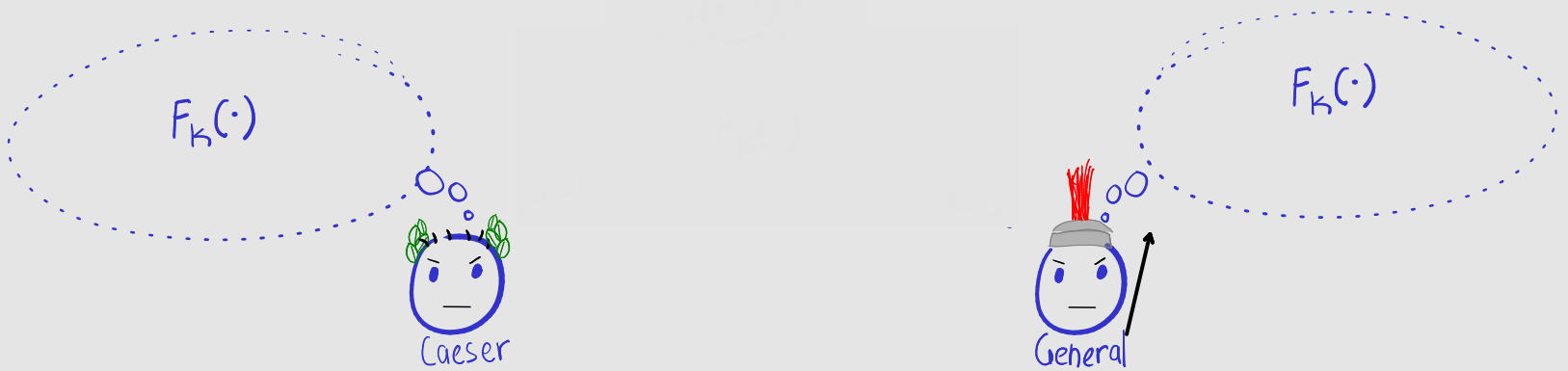
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

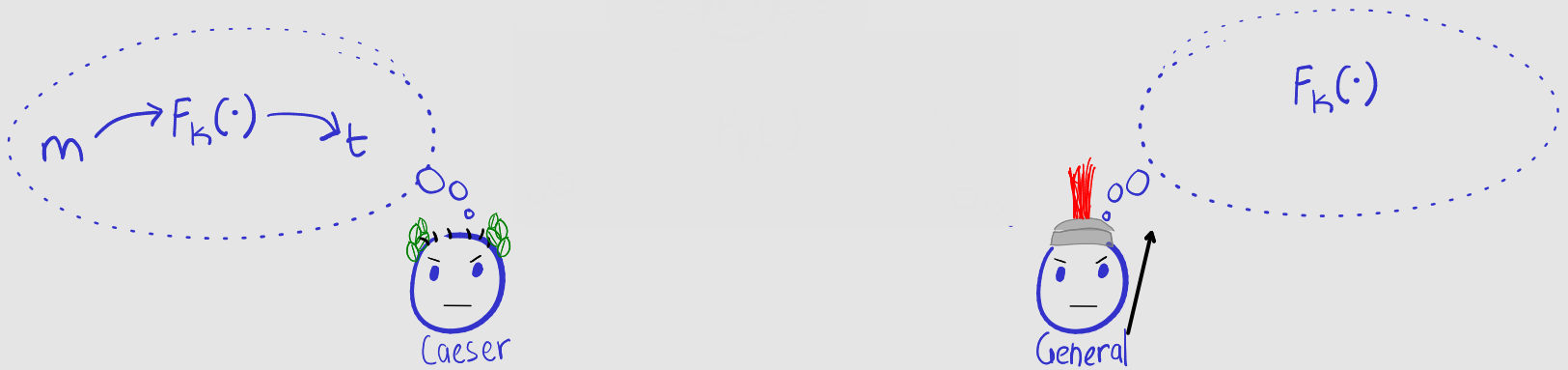
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

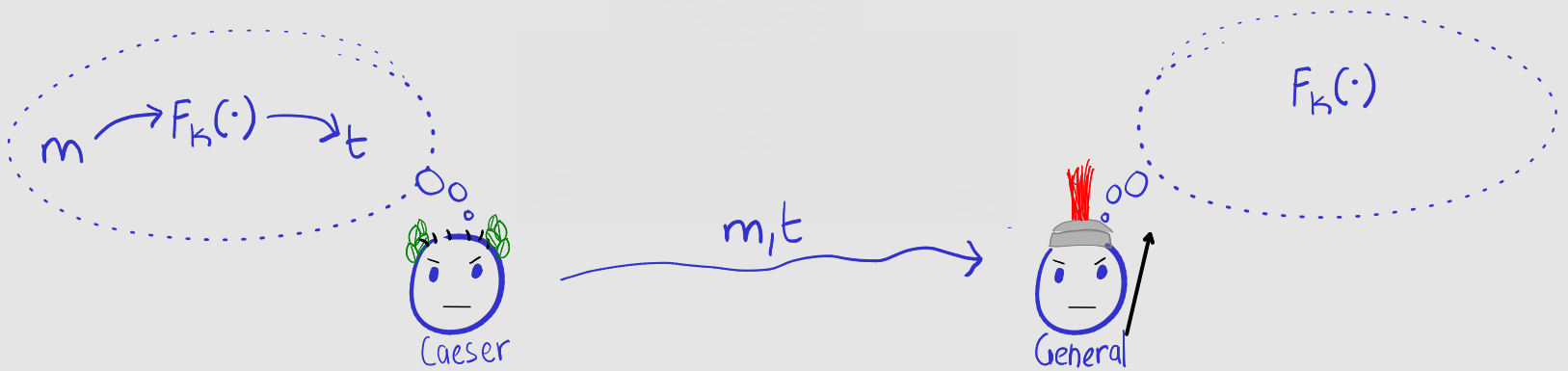
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

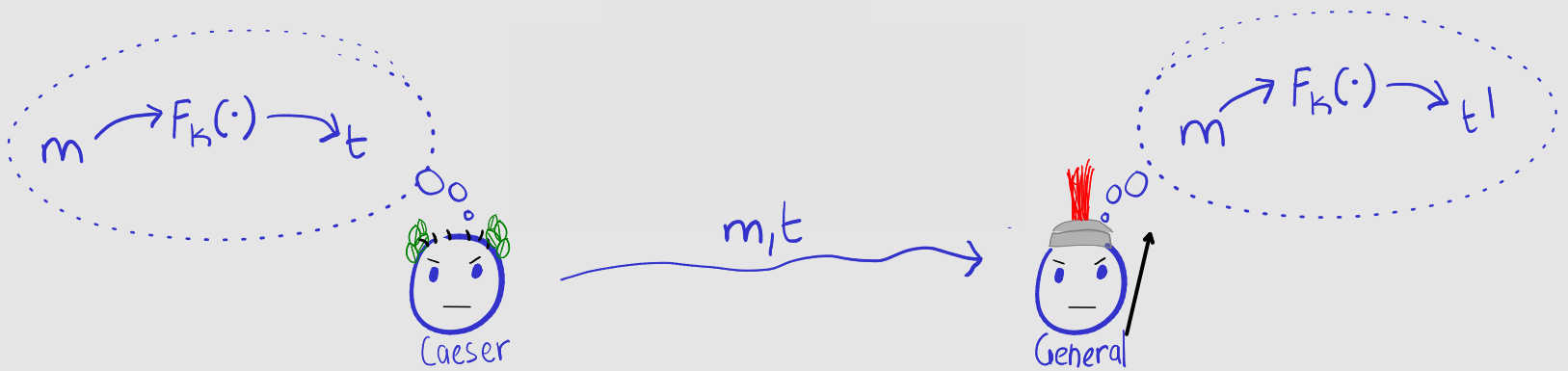
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

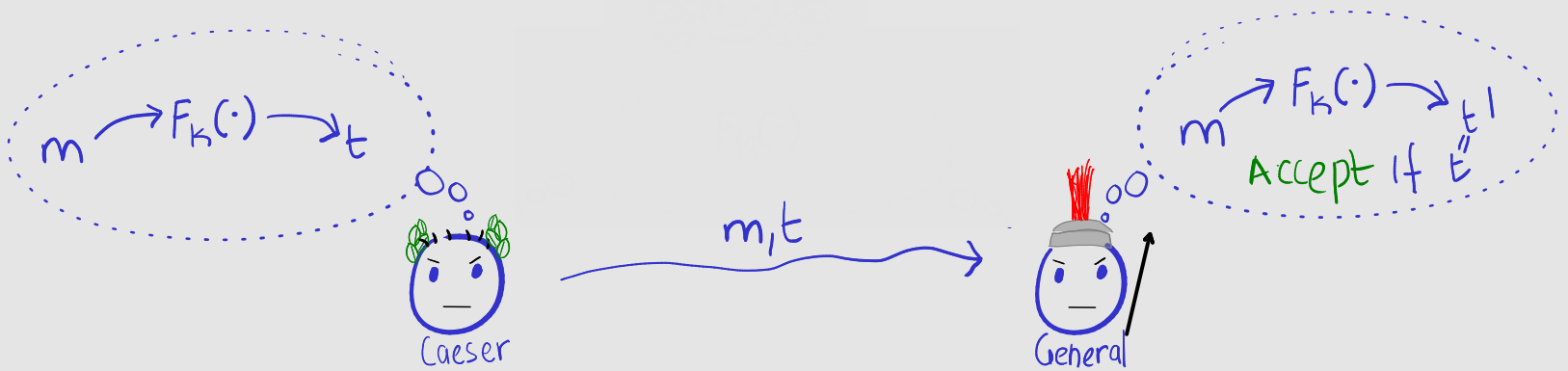
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

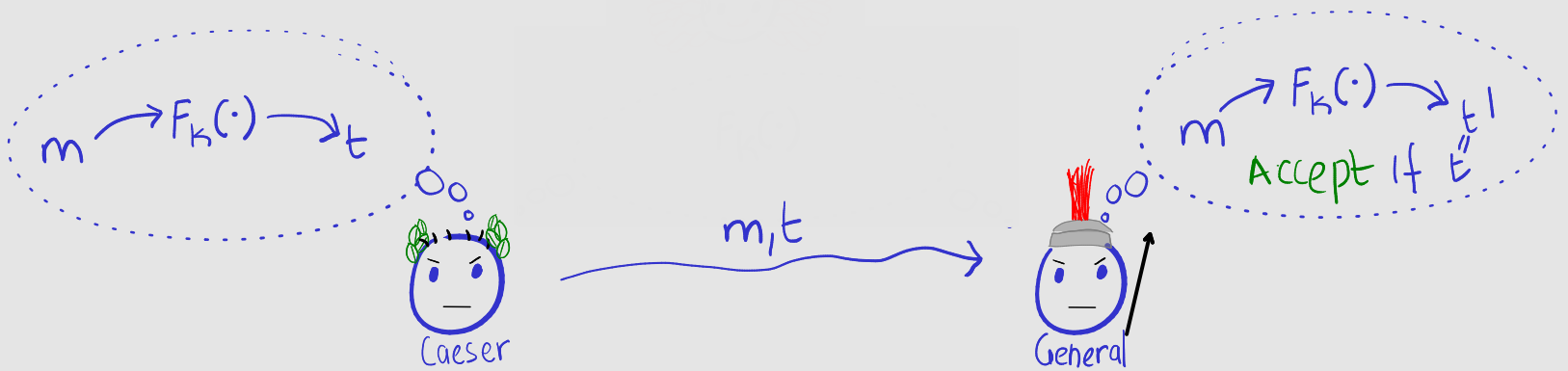
Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Let's Replace the Oracle in the Sky with a PRF!...

- Recall that PRF is a computational analogue of random oracles
- Should "appear" 2^n-wise independent to PPT Tams

Construction 3 (for $\mathcal{M}_n = \{0,1\}^n$ using $\{F_k : \{0,1\}^n \to \{0,1\}^n\}$)



Theorem 2

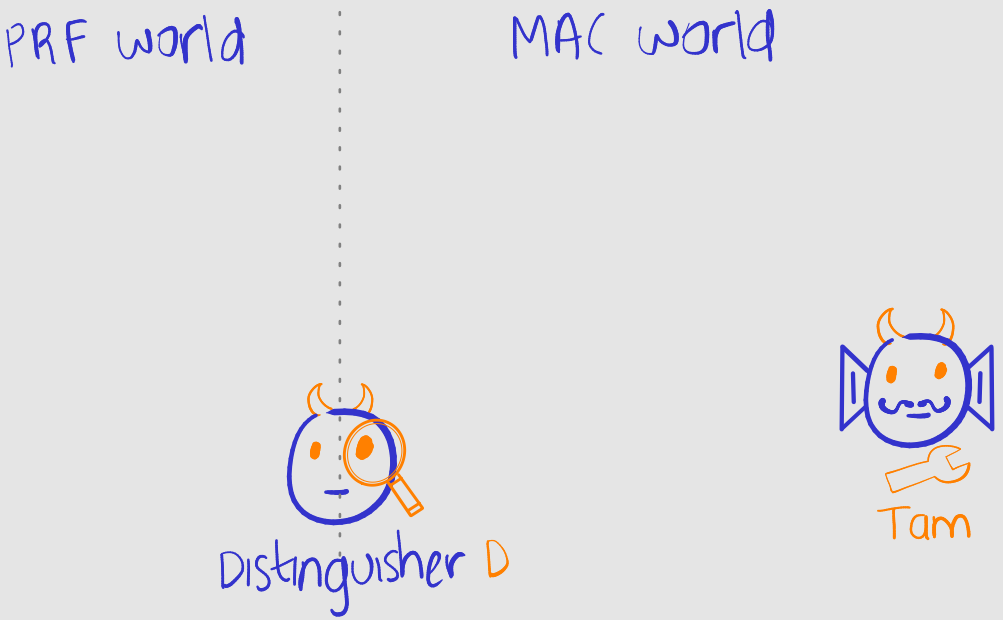
If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ is a PRF then Construction 3 is EU-CMA-secure against any PPT Tam.

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\} \Leftarrow \exists$ Tam against Construction 3.

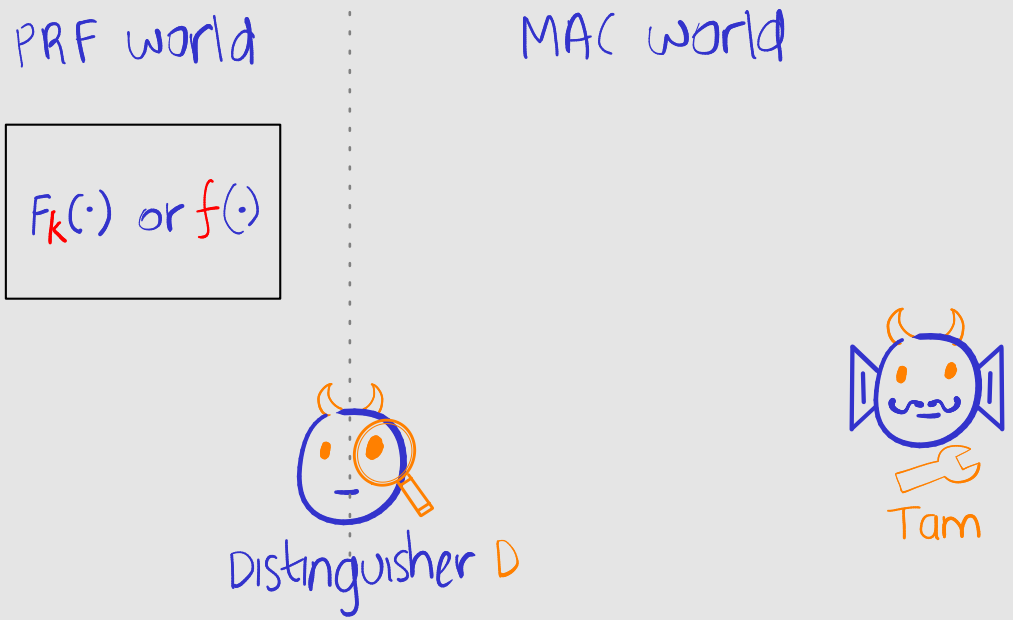


Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\} \Leftarrow \exists$ Tam against Construction 3.

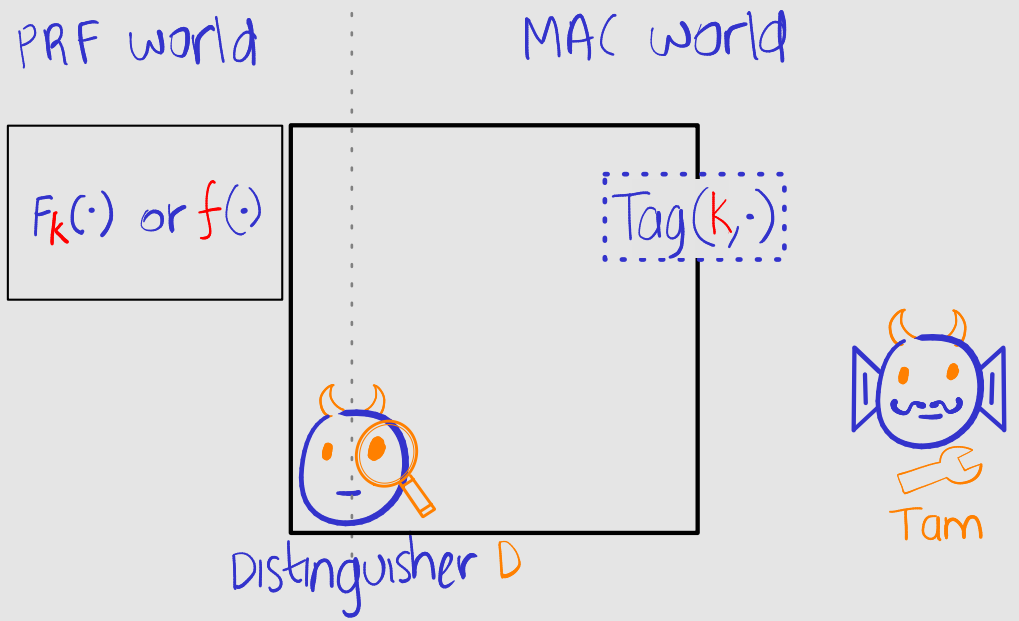


Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.

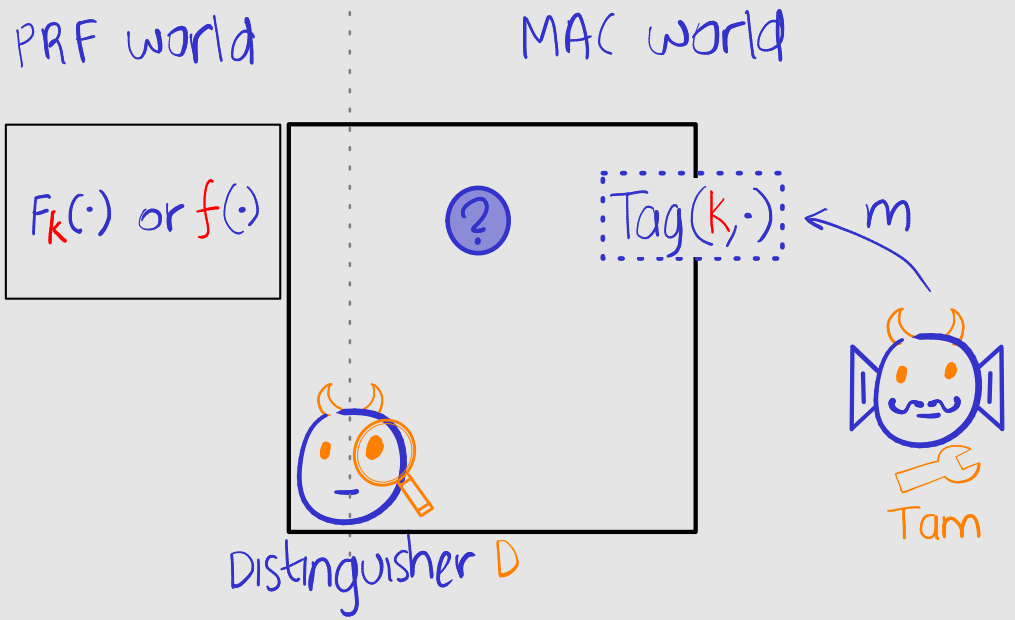


Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.

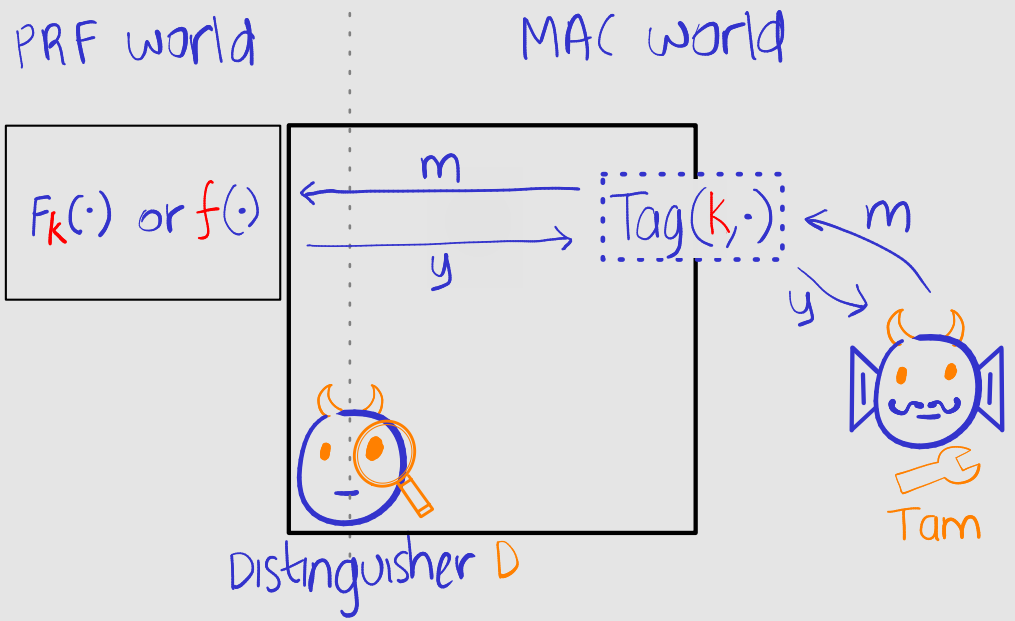


Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.



Reduction strategy

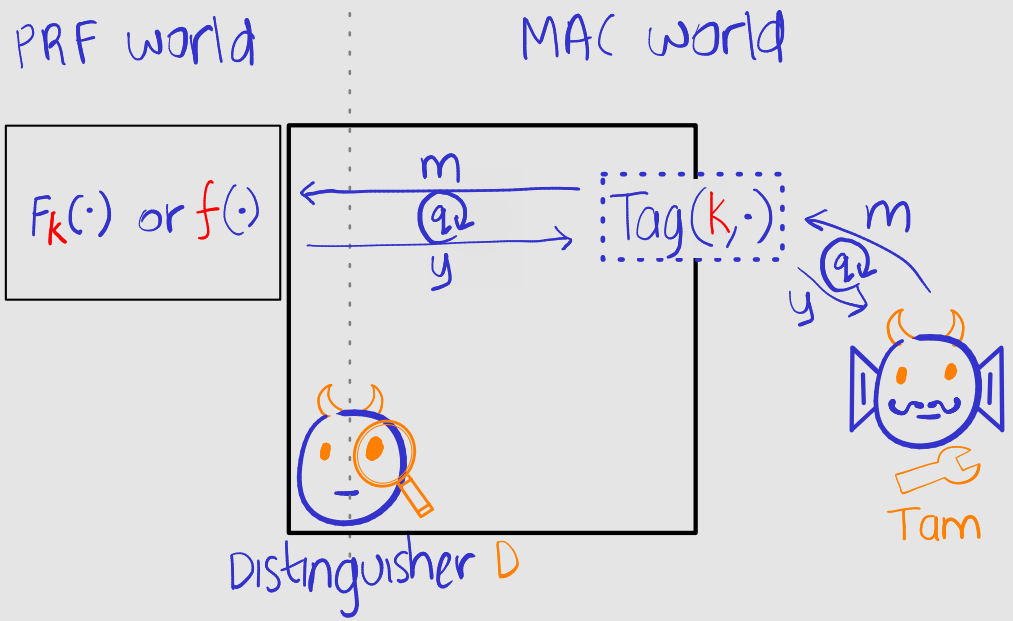
- D forwards Tam's tag oracle queries to its own oracle

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.



Reduction strategy

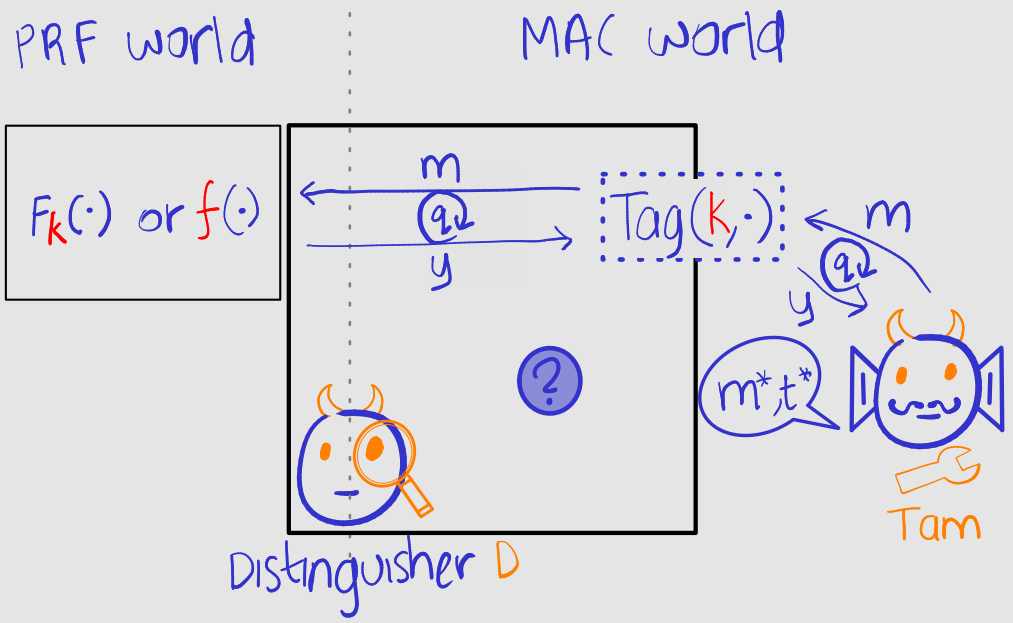
- D forwards Tam's tag oracle queries to its own oracle each time

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.



Reduction strategy

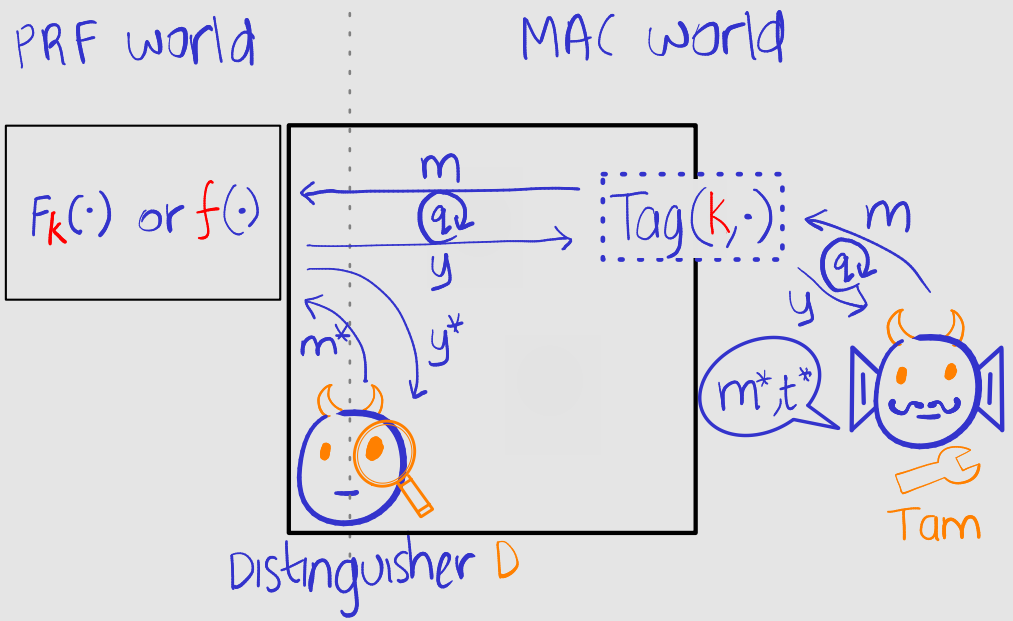
- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.



Reduction strategy

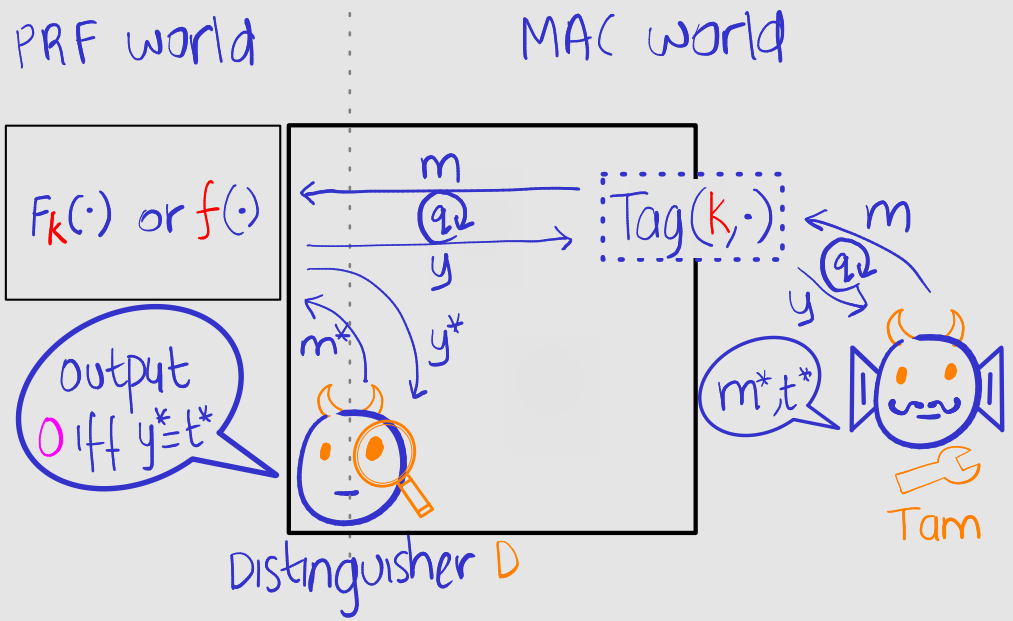
- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)
- D queries its oracle on m^* to obtain y^*

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.



Reduction strategy

- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*,t^*)
- D queries its oracle on m^* to obtain y^* & outputs 0 (pseudorandom) iff $y^*=t^*$

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\} \Leftarrow \exists$ Tam against Construction 3.

Analysis

D outputs 0 iff $y^* = t^* \Rightarrow$

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n)=0\right] - \Pr_{f \leftarrow \mathcal{F}_n}\left[D^{f(\cdot)}(1^n)=0\right] \right|$$

Reduction strategy

- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)
- D queries its oracle on m^* to obtain y^* & outputs 0 (pseudorandom) iff $y^* = t^*$

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.

Analysis

D outputs 0 iff $y^* = t^*$ ⇒

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n)=0\right] - \Pr_{f \leftarrow \mathcal{F}_n}\left[D^{f(\cdot)}(1^n)=0\right] \right|$$

$$= \left| \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{F_k(\cdot)}(1^n)}\left[F_k(m^*)=t^*\right] - \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{f(\cdot)}(1^n)}\left[f(m^*)=t^*\right] \right|$$

Reduction strategy

- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)
- D queries its oracle on m^* to obtain y^* & outputs 0 (pseudorandom) iff $y^* = t^*$

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.

Analysis

D outputs 0 iff $y^* = t^*$ ⇒

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n)=0\right] - \Pr_{f \leftarrow \mathcal{F}_n}\left[D^{f(\cdot)}(1^n)=0\right] \right|$$

$$= \left| \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{F_k(\cdot)}(1^n)}\left[F_k(m^*)=t^*\right] - \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{f(\cdot)}(1^n)}\left[f(m^*)=t^*\right] \right|$$

$$= \left| \text{non-negl} - 1/2^n \right| \quad (?)$$

= non-negl.

Reduction strategy

- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)
- D queries its oracle on m^* to obtain y^* & outputs 0 (pseudorandom) iff $y^* = t^*$

Let's Replace the Oracle in the Sky with a PRF!...

Theorem 2

If $\{F_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^n}$ *is a PRF then Construction 3 is EU-CMA-secure against any PPT* Tam.

Proof. ∃Distinguisher D for $\{F_k\}$ ⇐ ∃ Tam against Construction 3.

Analysis

D outputs 0 iff $y^* = t^*$ ⇒

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n)=0\right] - \Pr_{f \leftarrow \mathcal{F}_n}\left[D^{f(\cdot)}(1^n)=0\right] \right|$$

$$= \left| \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{F_k(\cdot)}(1^n)}\left[F_k(m^*)=t^*\right] - \Pr_{(m^*,t^*) \leftarrow \mathrm{Tam}^{f(\cdot)}(1^n)}\left[f(m^*)=t^*\right] \right|$$

$$= \left| \text{non-negl} - 1/2^n \right| \quad \text{(f random!)}$$

= non-negl.

Reduction strategy

- D forwards Tam's tag oracle queries to its own oracle each time
- Tam outputs a forgery (m^*, t^*)
- D queries its oracle on m^* to obtain y^* & outputs 0 (pseudorandom) iff $y^* = t^*$

Plan for this Lecture

1 Message-Authentication Code (MAC)

2 Constructing MACs

3 Chosen-Ciphertext Attack

Why are Malleable Ciphertexts Problematic?

- Recall: all SKE constructions so far are malleable

$c = k \oplus m$ OTP

$c = G(k) \oplus m$ computational OTP

$c = (F_K(r) \oplus m, r)$ PRF-SKE

Why are Malleable Ciphertexts Problematic?

- Recall: all SKE constructions so far are malleable

$c = k \oplus m$ $c = G(k) \oplus m$ $c = (F_k(r) \oplus m, r)$

OTP (computational OTP) PRF-SKE

- There were historical cases where this was exploited
  - Padding oracle attack
  - Bleichenbacher's attack on PKCS#1 v1.5

Why are Malleable Ciphertexts Problematic?

- Recall: all SKE constructions so far are malleable

- There were historical cases where this was exploited
  - Padding oracle attack
  - Bleichenbacher's attack on PKCS#1 v1.5

- These attacks roughly follow following high-level template:
  - Maul ciphertext
  - Use a 'decryption oracle' to learn info. about mauled plaintext
  - Infer information about the original plaintext
  - (Repeat if necessary)

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages

Chosen-Ciphertext Attack (CCA)

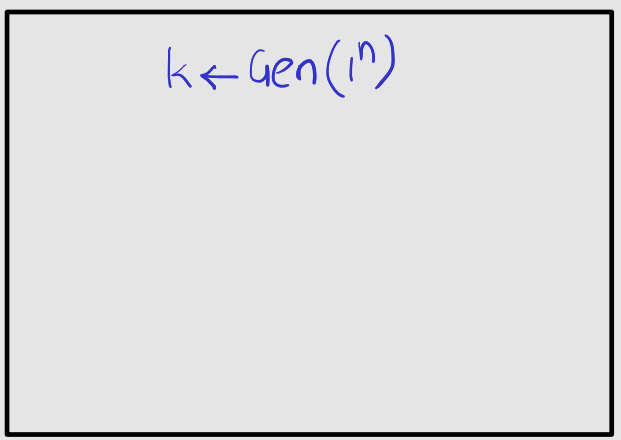
- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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

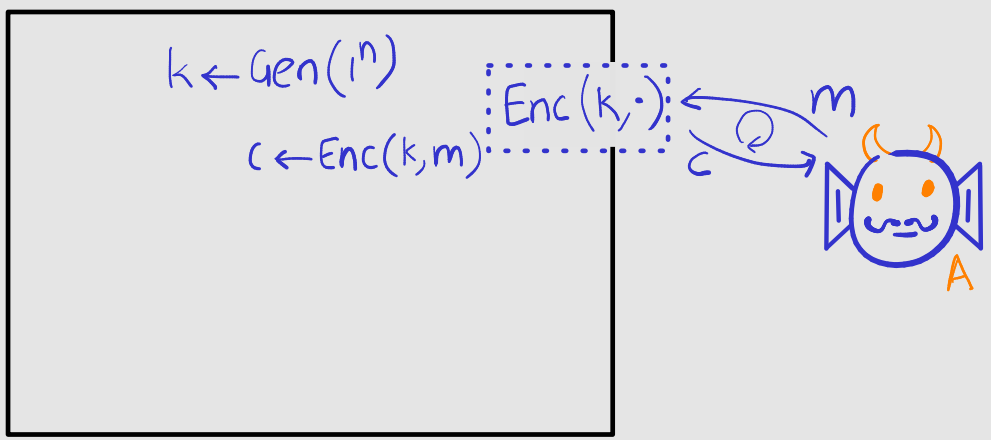


Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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



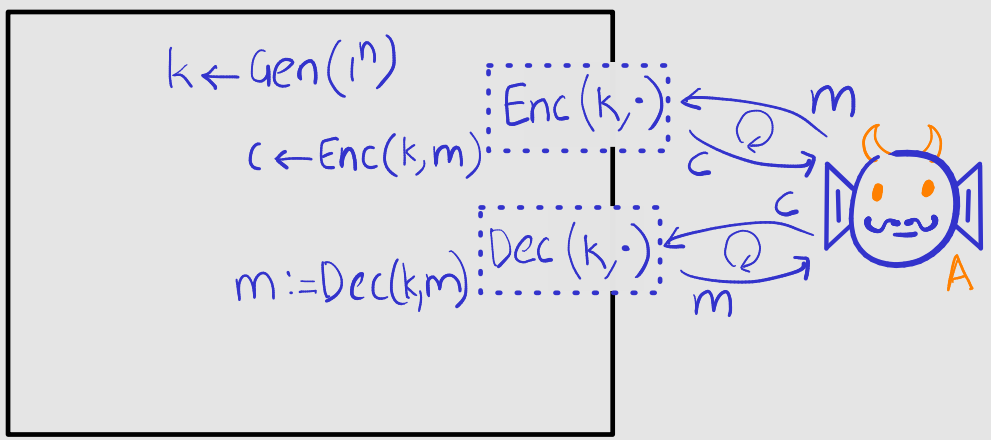
- A can query Enc oracle on messages of its choice

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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



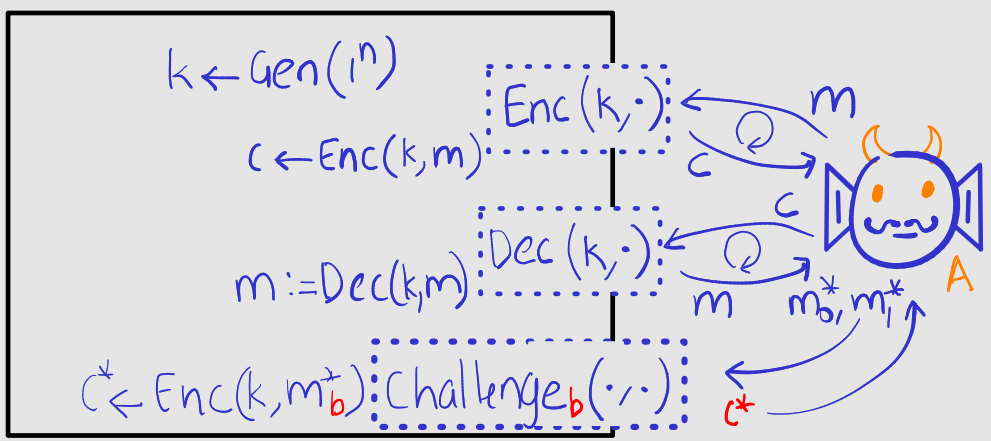
- A can query Enc oracle on messages of its choice
- A can query Dec oracle on ciphertexts of its choice

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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



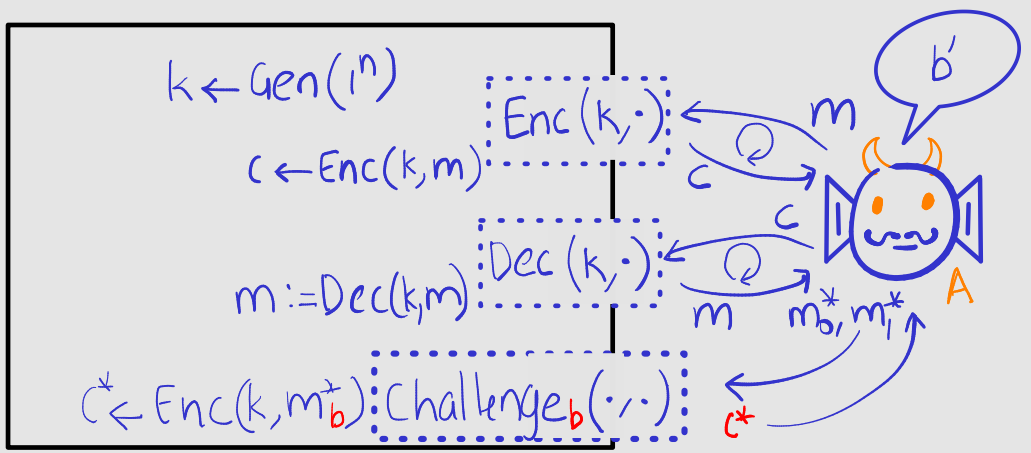
- A can query Enc oracle on messages of its choice
- A can query Dec oracle on ciphertexts of its choice
- A can query the challenge oracle once
- A can make further Enc/Dec queries, but not Dec(k,c*)

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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



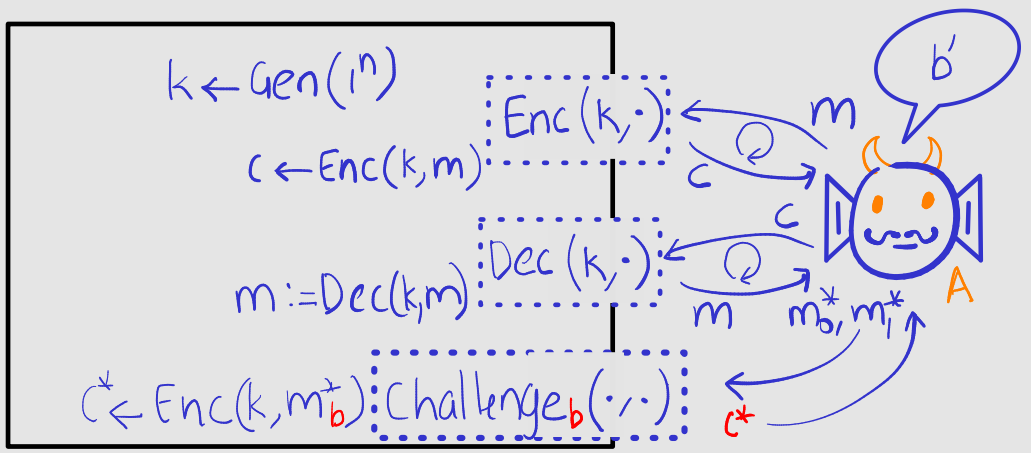
- A can query Enc oracle on messages of its choice
- A can query Dec oracle on ciphertexts of its choice
- A can query the challenge oracle once
- A can make further Enc/Dec queries, but not Dec(k,c*)

Chosen-Ciphertext Attack (CCA)

- Recall CPA: adversaries who can influence Caeser's messages
- CCA = CPA + access to decryption oracle

Defintion 5 (Secrecy against chosen-ciphertext attack (CCA))

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



- A can query Enc oracle on messages of its choice
- A can query Dec oracle on ciphertexts of its choice
- A can query the challenge oracle once
- A can make further Enc/Dec queries, but not Dec(k,c*)

Exercise 3 (CCA model)

Show that Construction 1 from Lecture 5 is not CCA-secure.

CCA-Secure SKE via Encrypt-then-MAC

Construction 4 (Based on CPA-secret SKE $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and EU-CMA secure MAC $M := (\mathsf{Gen}^*, \mathsf{Tag}, \mathsf{Ver})$)

- $\mathsf{Gen}'(1^n)$: *output keys* $k \leftarrow \mathsf{Gen}(1^n)$ *and* $k^* \leftarrow \mathsf{Gen}^*(1^n)$
- $\mathsf{Enc}((k, k^*), m)$: *output* $c \leftarrow \mathsf{Enc}(k, m)$ *and* $t \leftarrow \mathsf{Tag}(k^*, c)$
- $\mathsf{Dec}((k, k^*), (c, t))$: *output* $m := \mathsf{Dec}(k, c)$ *if* $\mathsf{Ver}(k^*, c, t) = 1$

CCA-Secure SKE via Encrypt-then-MAC

Construction 4 (Based on CPA-secret SKE $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and EU-CMA secure MAC $M := (\mathsf{Gen}^*, \mathsf{Tag}, \mathsf{Ver})$)

- $\mathsf{Gen}'(1^n)$: *output keys* $k \leftarrow \mathsf{Gen}(1^n)$ *and* $k^* \leftarrow \mathsf{Gen}^*(1^n)$
- $\mathsf{Enc}((k, k^*), m)$: *output* $c \leftarrow \mathsf{Enc}(k, m)$ *and* $t \leftarrow \mathsf{Tag}(k^*, c)$
- $\mathsf{Dec}((k, k^*), (c, t))$: *output* $m := \mathsf{Dec}(k, c)$ *if* $\mathsf{Ver}(k^*, c, t) = 1$

Theorem 3

If Π is a CPA-secret SKE and M is a EU-CMA-secure MAC then Construction 3 is CCA-secure.

Proof intuition.

- Since a ciphertext cannot be mauled to another valid one thanks to MAC security, the decryption oracle is useless.
- Now exploit CPA-secrecy

□

CCA-Secure SKE via Encrypt-then-MAC

Construction 4 (Based on CPA-secret SKE $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and EU-CMA secure MAC $M := (\mathsf{Gen}^*, \mathsf{Tag}, \mathsf{Ver})$)

- $\mathsf{Gen}'(1^n)$: *output keys* $k \leftarrow \mathsf{Gen}(1^n)$ *and* $k^* \leftarrow \mathsf{Gen}^*(1^n)$
- $\mathsf{Enc}((k, k^*), m)$: *output* $c \leftarrow \mathsf{Enc}(k, m)$ *and* $t \leftarrow \mathsf{Tag}(k^*, c)$
- $\mathsf{Dec}((k, k^*), (c, t))$: *output* $m := \mathsf{Dec}(k, c)$ *if* $\mathsf{Ver}(k^*, c, t) = 1$

Theorem 3

If Π is a CPA-secret SKE and M is a EU-CMA-secure MAC then Construction 3 is CCA-secure.

strongly

Proof intuition.

- Since a ciphertext cannot be mauled to another valid one thanks to MAC security, the decryption oracle is useless.
- Now exploit CPA-secrecy

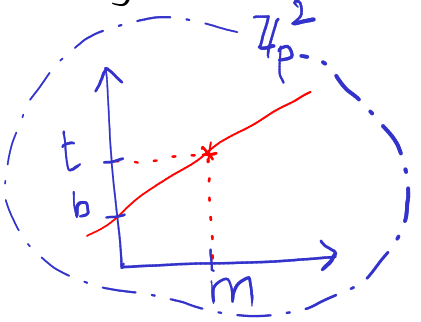
□

To Recap Today's Lecture

- Task 2: secret communication against *active* adversary

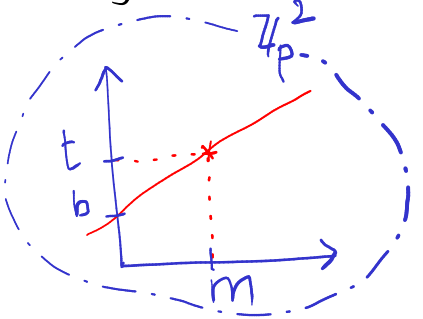
To Recap Today's Lecture

- Task 2: secret communication against *active* adversary
  - Sub-task: How to detect tampering?
    - Message authentication codes (MAC)
    - Pairwise-independent hash → one-time MAC
    - PRF → (many-time) MAC

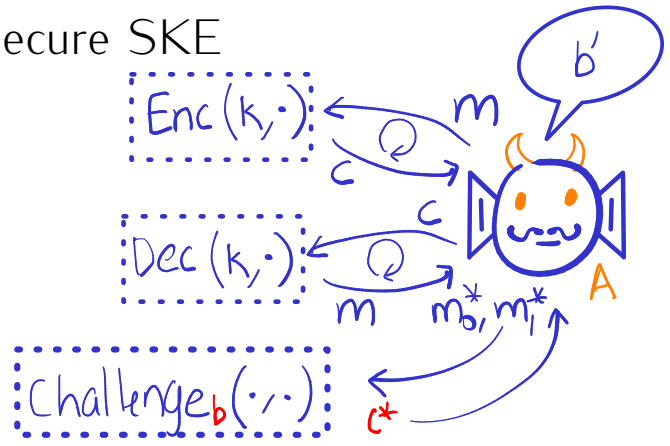


To Recap Today's Lecture

- Task 2: secret communication against *active* adversary
  - Sub-task: How to detect tampering?
    - Message authentication codes (MAC)
    - Pairwise-independent hash → one-time MAC
    - PRF → (many-time) MAC

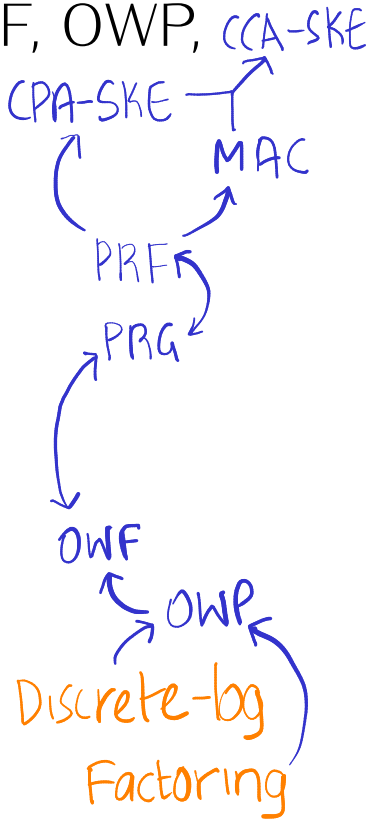


  - How to model secrecy against tampering adversary?
    - Chosen-ciphertext attack (CCA)
    - CPA-secret SKE + MAC → CCA-secure SKE



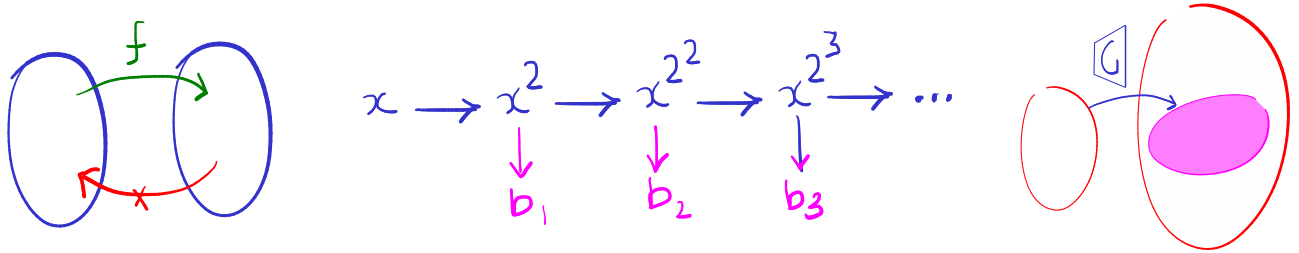
To Recap Module I

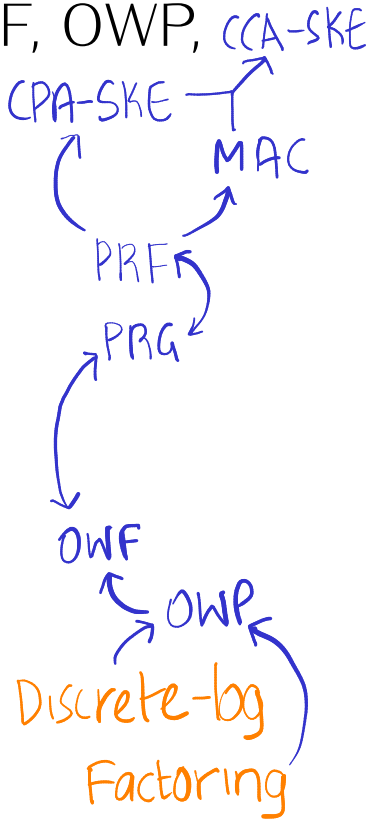
- We learnt: secure communication using SKE + MAC
- Cryptographic primitives encountered: PRG, PRF, OWF, OWP, CCA-SKE, CPA-SKE, MAC, pairwise-independent hash
- Hardness assumptions: factoring, discrete-logarithm



To Recap Module I

- We learnt: secure communication using SKE + MAC
- Cryptographic primitives encountered: PRG, PRF, OWF, OWP, CCA-SKE pairwise-independent hash
- Hardness assumptions: factoring, discrete-logarithm

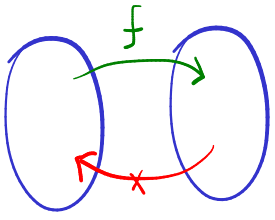


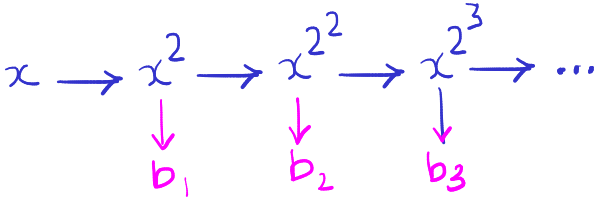


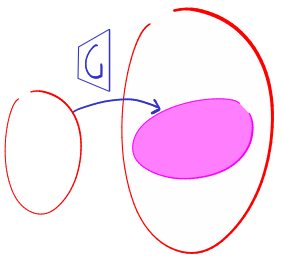
- Key conceptual takeaway: pseudo-randomness $\leftrightarrow$ hardness $\leftrightarrow$ unpredictability

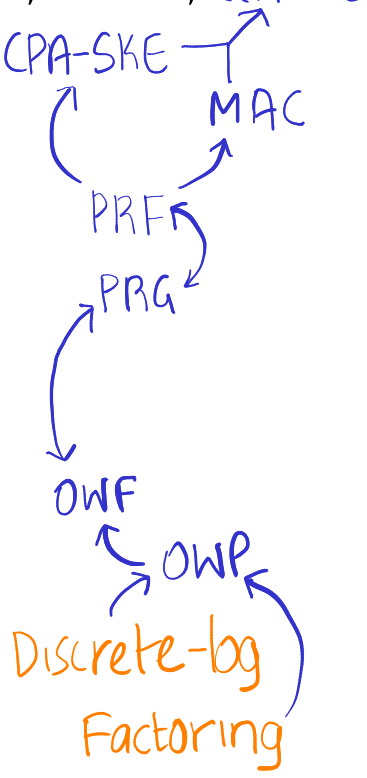
To Recap Module I

- We learnt: secure communication using SKE + MAC
- Cryptographic primitives encountered: PRG, PRF, OWF, OWP, CCA-SKE, CPA-SKE, MAC, pairwise-independent hash
- Hardness assumptions: factoring, discrete-logarithm

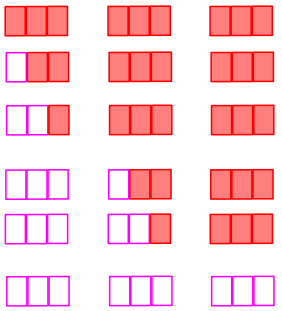








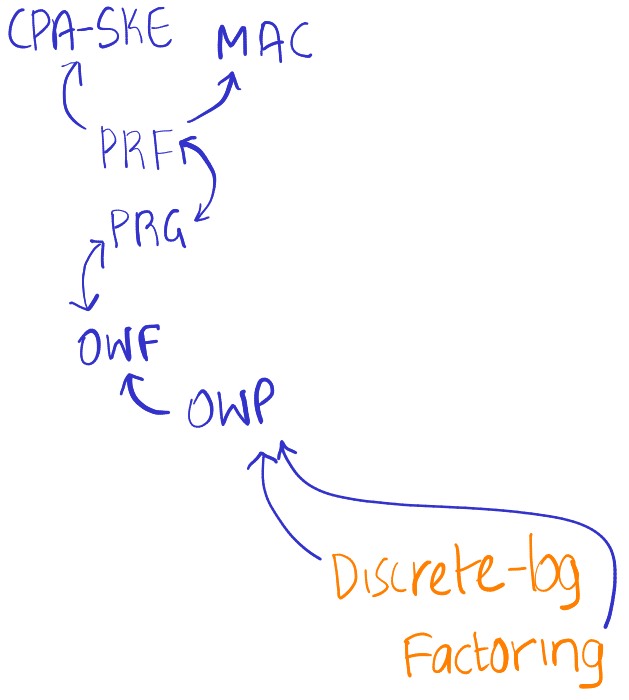
- Key conceptual takeaway: pseudo-randomness $\leftrightarrow$ hardness $\leftrightarrow$ unpredictability



- Key tools: security reduction, hybrid argument

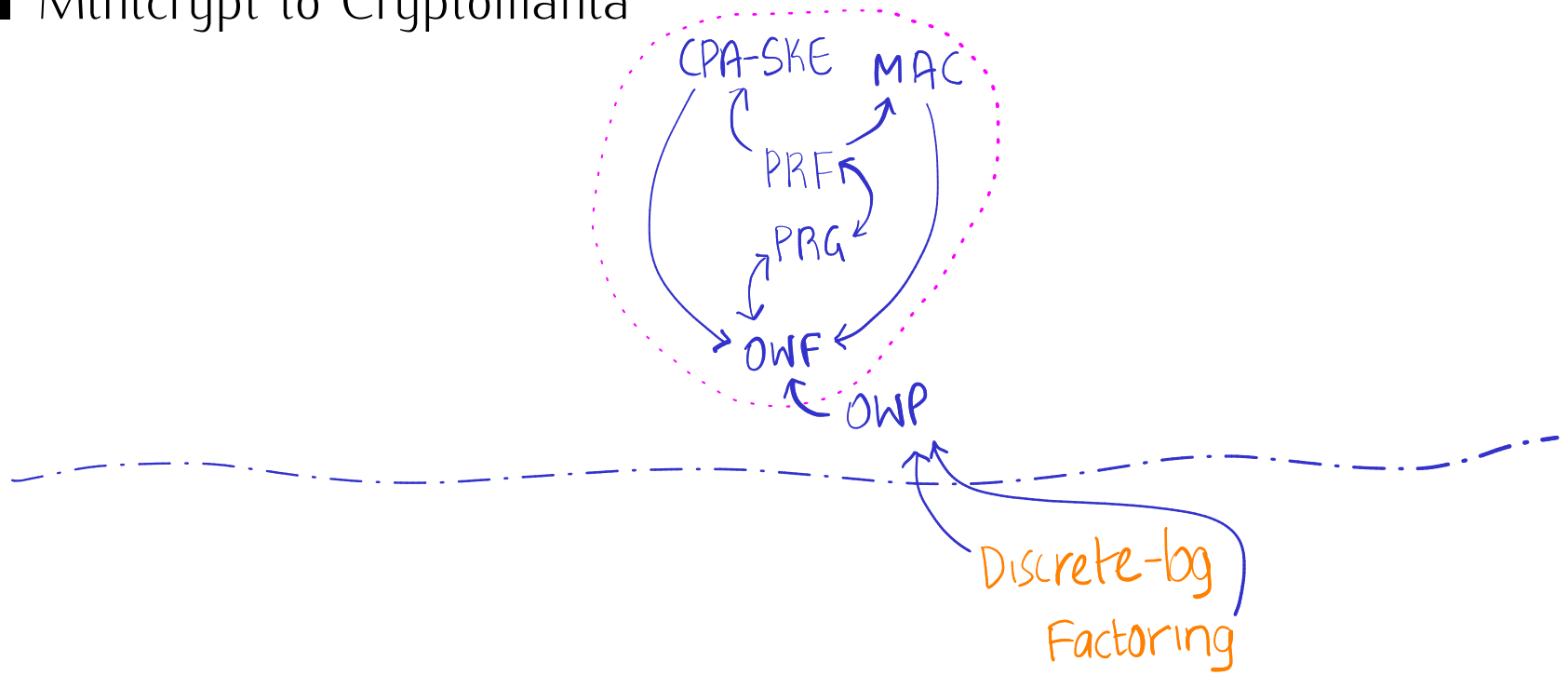
Next Module/Lecture

- Minicrypt to Cryptomania



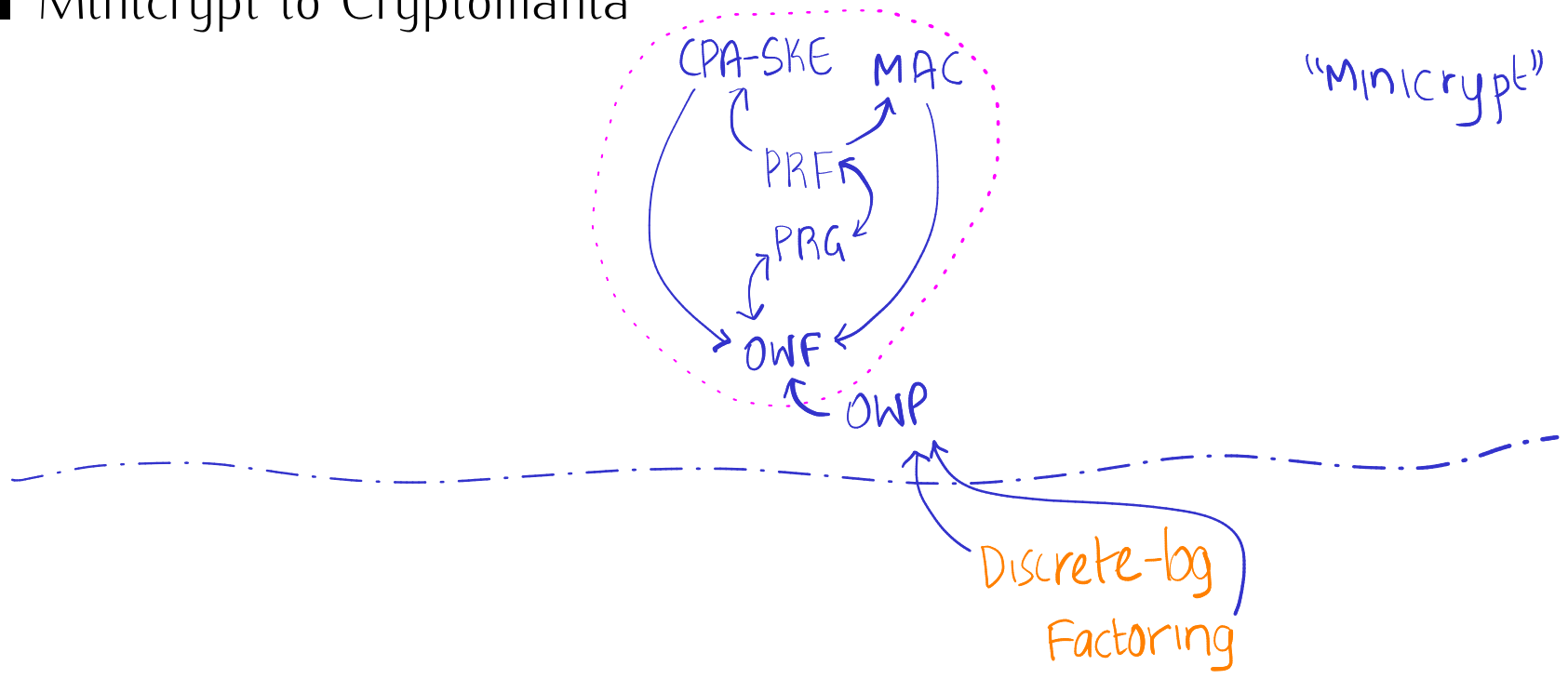
Next Module/Lecture

- Minicrypt to Cryptomania



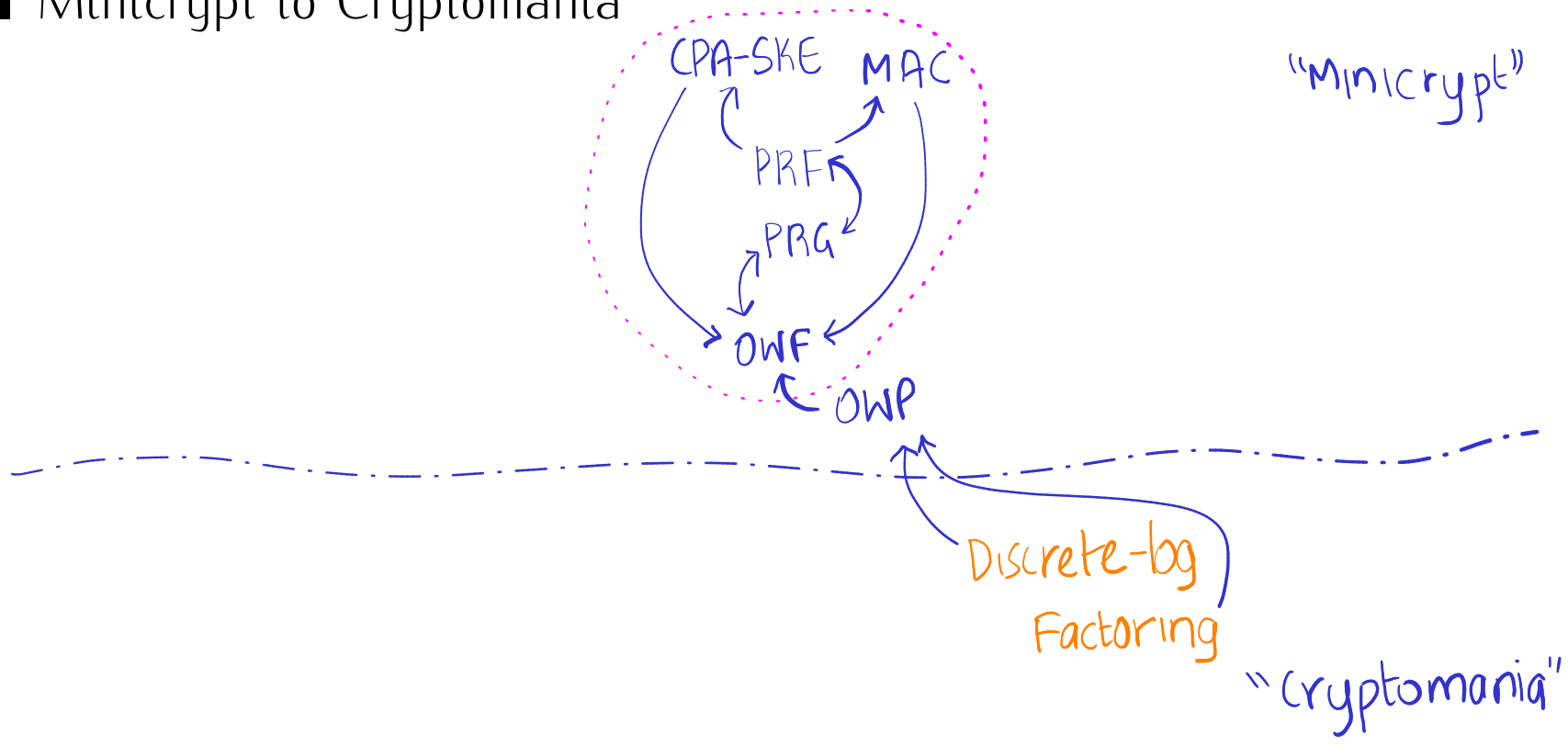
Next Module/Lecture

- Minicrypt to Cryptomania



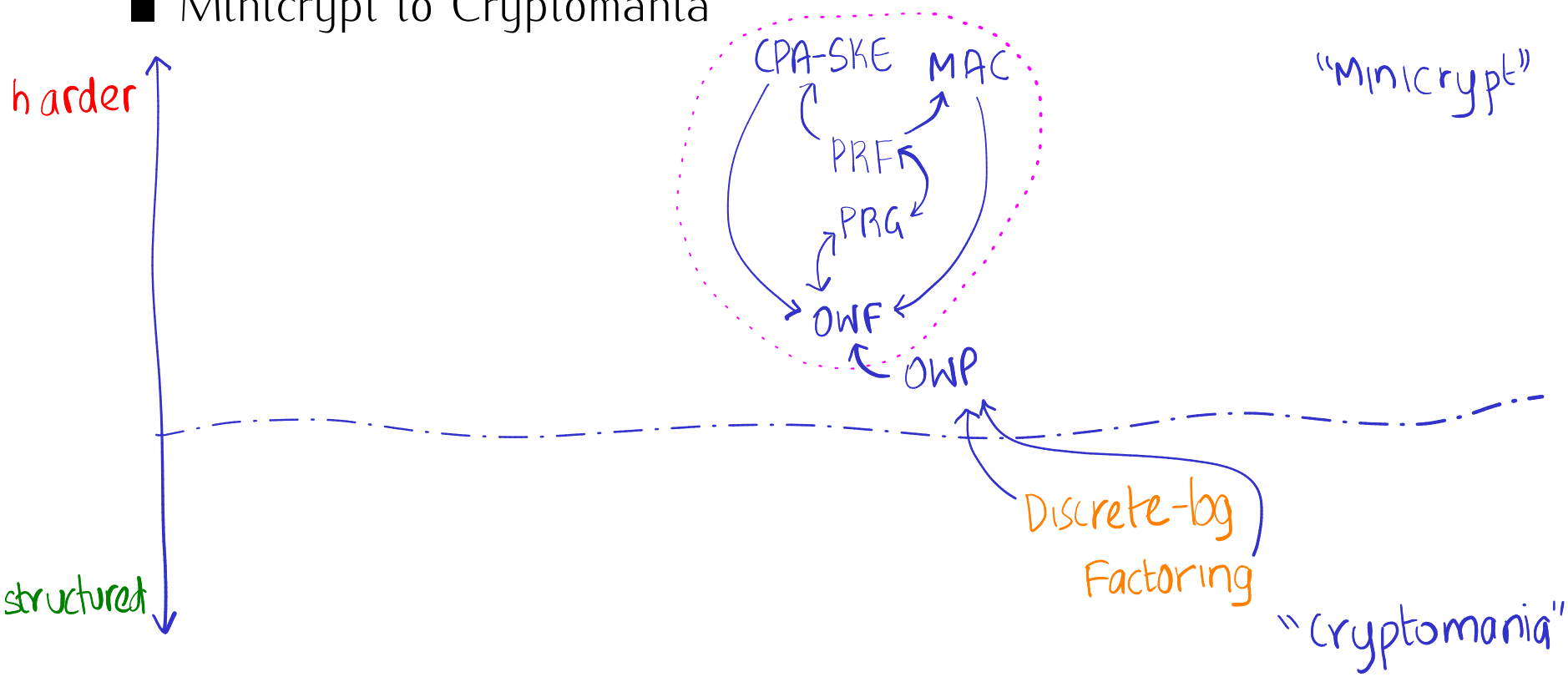
Next Module/Lecture

- Minicrypt to Cryptomania



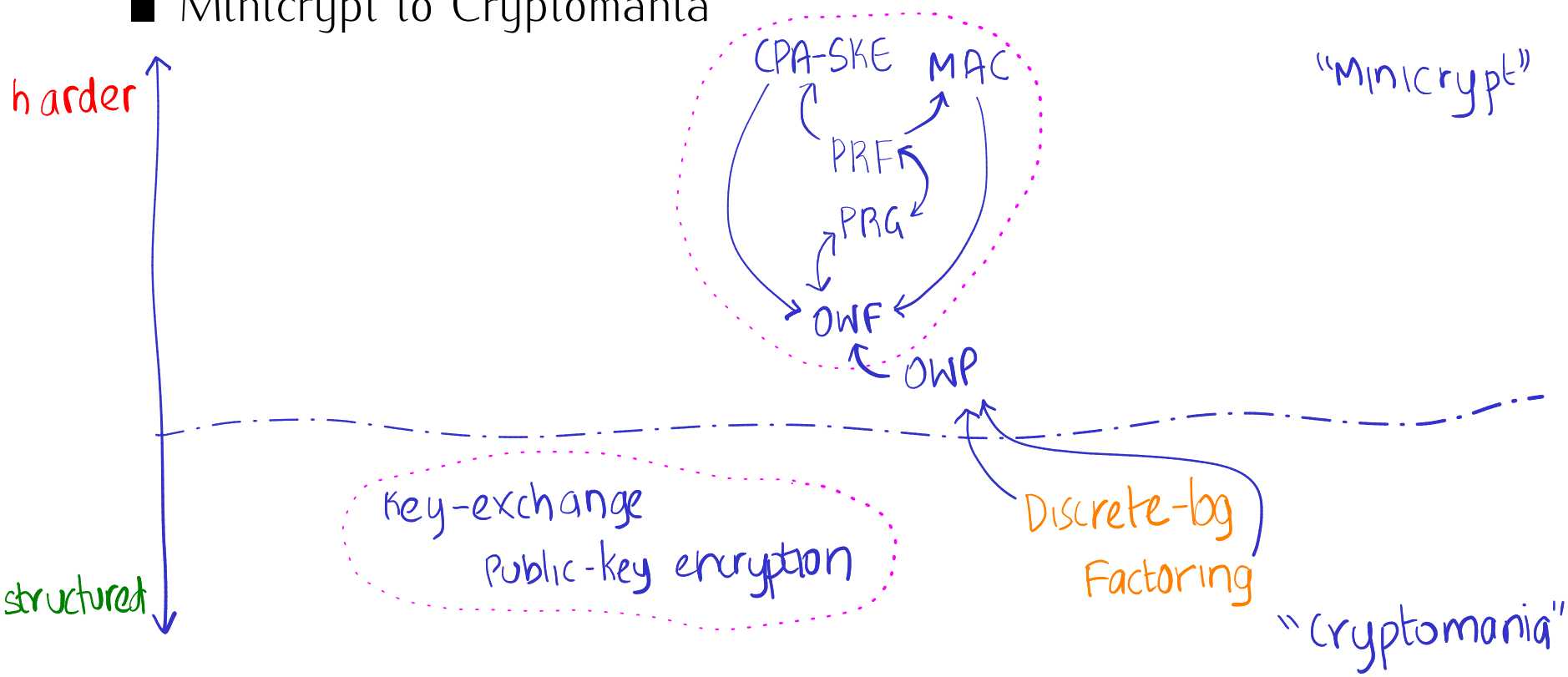
Next Module/Lecture

- Minicrypt to Cryptomania



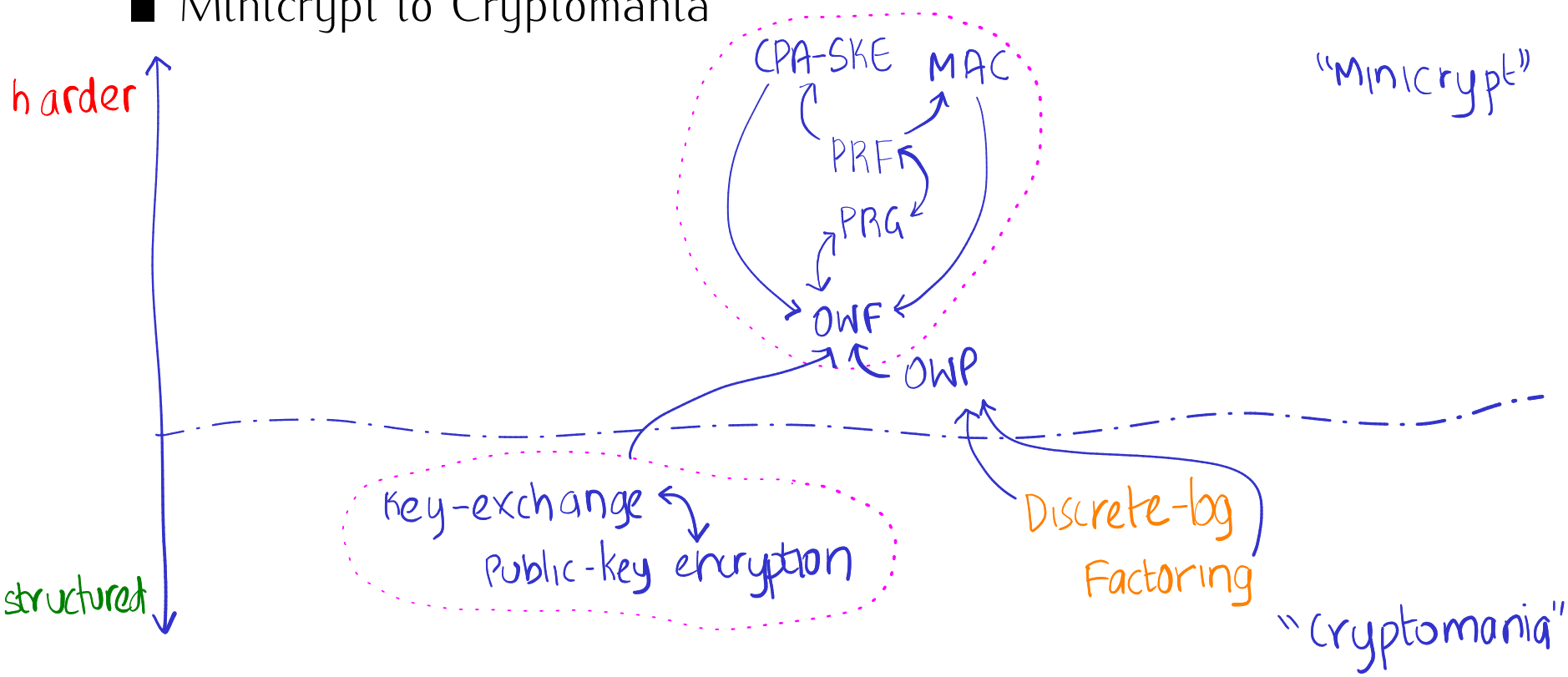
Next Module/Lecture

- Minicrypt to Cryptomania



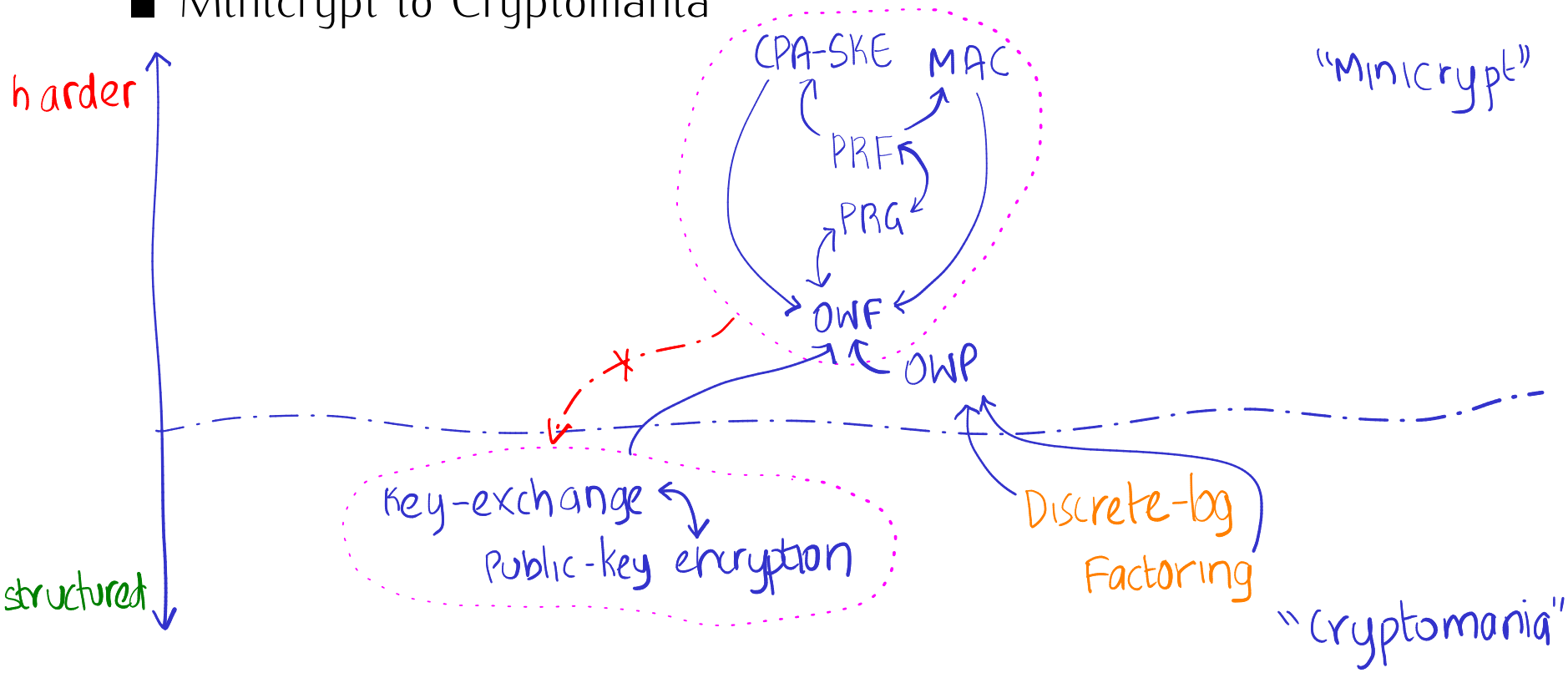
Next Module/Lecture

- Minicrypt to Cryptomania



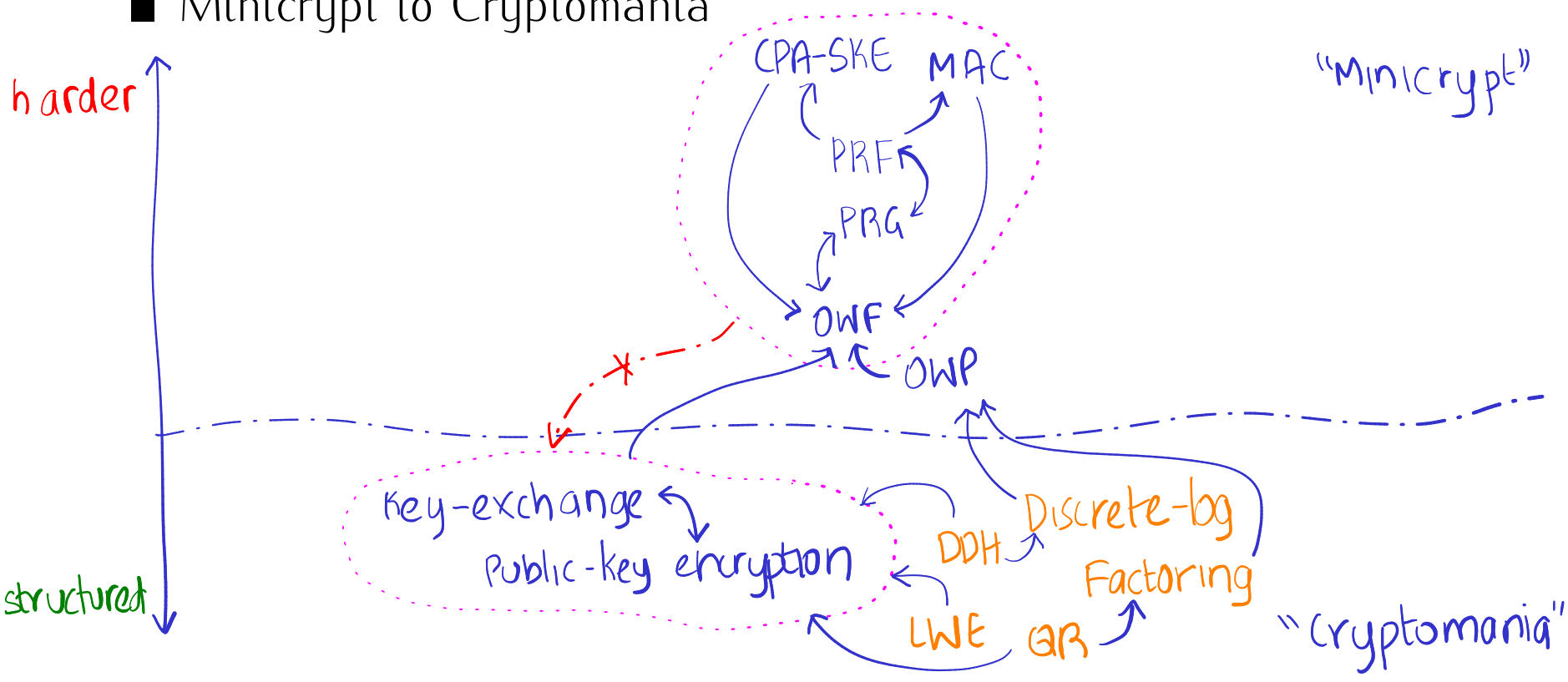
Next Module/Lecture

- Minicrypt to Cryptomania



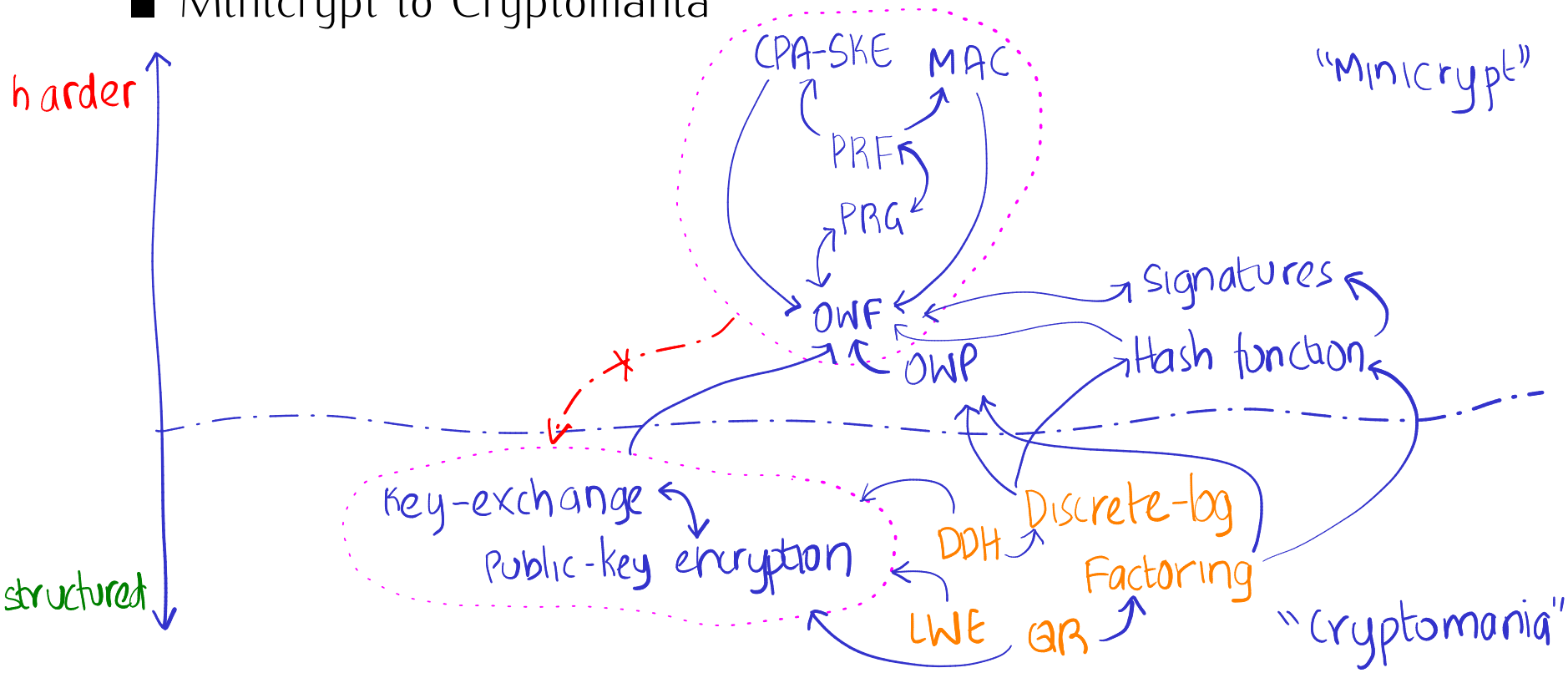
Next Module/Lecture

- Minicrypt to Cryptomania



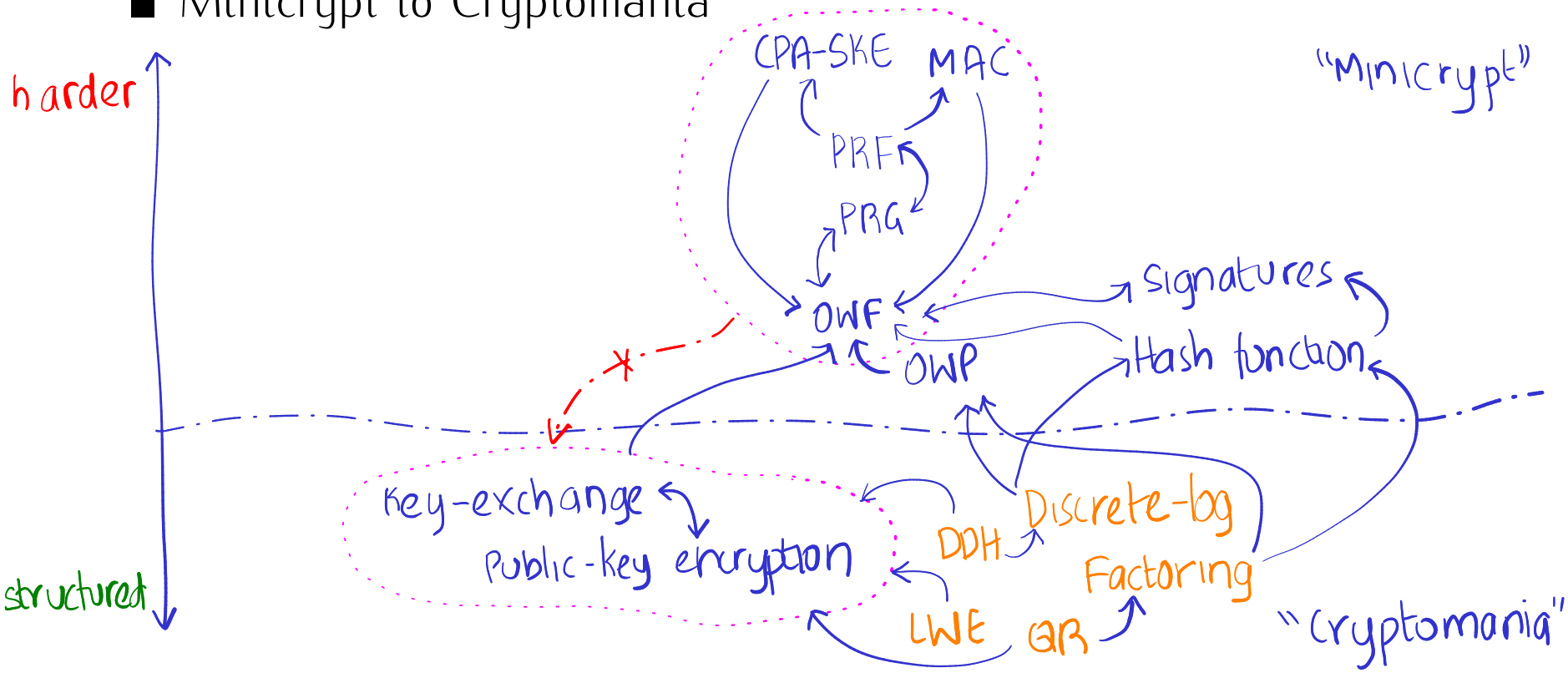
Next Module/Lecture

- Minicrypt to Cryptomania



Next Module/Lecture

- Minicrypt to Cryptomania



- We start with Task 3: how to establishing a shared key without having met before!

More Questions?

References

1. [KL14, Chapters 4 and 5] for more details about the lecture.
2. Pairwise- and k-wise independent hash functions were introduced in [WC81]. The information-theoretic MACs based on k-wise independent hash functions were also proposed there.
3. CCA was studied in [RS92, NY90].
4. You can read more about Bleichenbacher's attack in [Ble97].


Daniel Bleichenbacher.

On the security of the KMOV public key cryptosystem.

In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 235–248. Springer, Heidelberg, August 1997.


Jonathan Katz and Yehuda Lindell.

Introduction to Modern Cryptography (3rd ed.).

Chapman and Hall/CRC, 2014.


Moni Naor and Moti Yung.

Public-key cryptosystems provably secure against chosen ciphertext attacks.

In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.


Charles Rackoff and Daniel R. Simon.

Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack.

In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992.


Mark N. Wegman and J.Lawrence Carter.

New hash functions and their use in authentication and set equality.

Journal of Computer and System Sciences, 22(3):265–279, 1981.