Perfect Security

601.442/642 Modern Cryptography

22nd January 2026

Agenda

- Private communication and encryption schemes
- Defining an encryption scheme
  - First crypto definition!
- One-time pads
  - First crypto scheme!

A Few Remarks

- Ask questions!

A Few Remarks

- Ask questions!
    - At any point during the lecture.

A Few Remarks

- Ask questions!
    - At any point during the lecture.
    - As many as you want.

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.
- Build intuition!

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.
- Build intuition!
  - Definitions are tools for modeling security goals, not universal truths. Think about alternatives.

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.
- Build intuition!
  - Definitions are tools for modeling security goals, not universal truths. Think about alternatives.
  - Play around with constructions and proofs. Always ask “why”. Extract the underlying idea.

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.
- Build intuition!
  - Definitions are tools for modeling security goals, not universal truths. Think about alternatives.
  - Play around with constructions and proofs. Always ask “why”. Extract the underlying idea.
  - Sometimes intuition may not align with the proof. But it will, once we make the intuition *robust*.

A Few Remarks

- Ask questions!
  - At any point during the lecture.
  - As many as you want.
  - Review material after class and ask questions on Canvas.
- Build intuition!
  - Definitions are tools for modeling security goals, not universal truths. Think about alternatives.
  - Play around with constructions and proofs. Always ask “why”. Extract the underlying idea.
  - Sometimes intuition may not align with the proof. But it will, once we make the intuition *robust*.

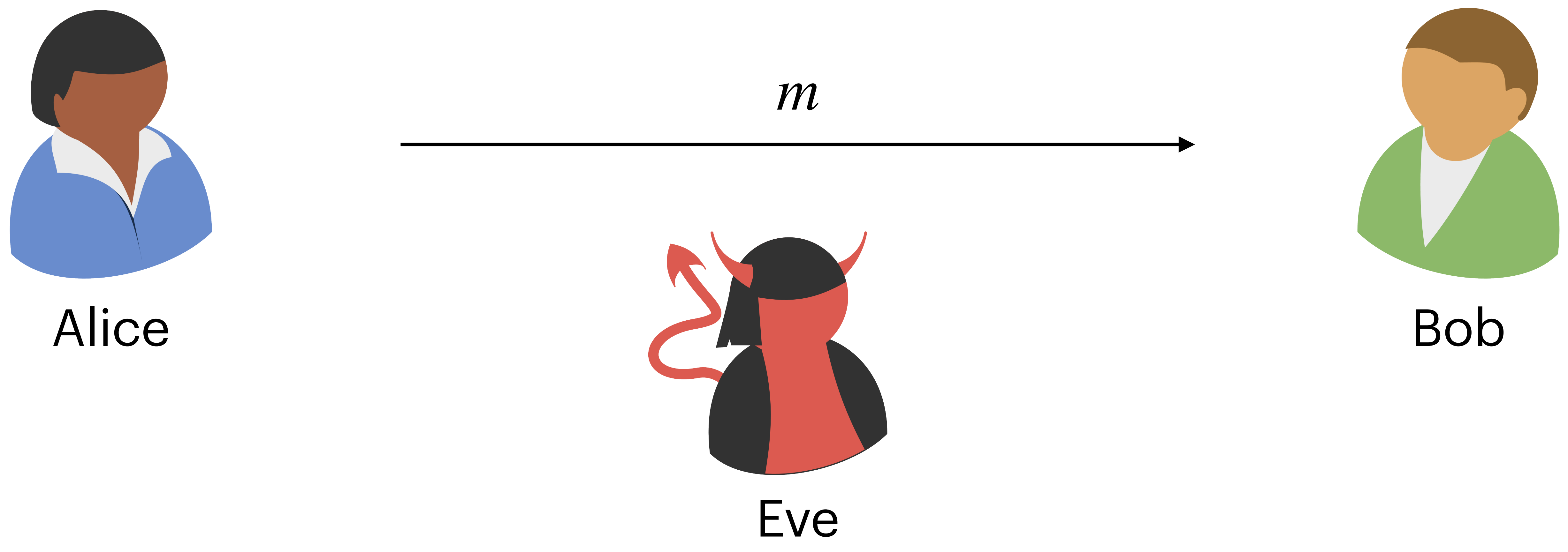
“It is by logic that we prove, but by intuition that we discover.”

- Henri Poincaré

Private Communication

Private Communication

The Private Communication Problem



Alice wants to send a message m to Bob, while keeping the message hidden from an eavesdropper Eve.

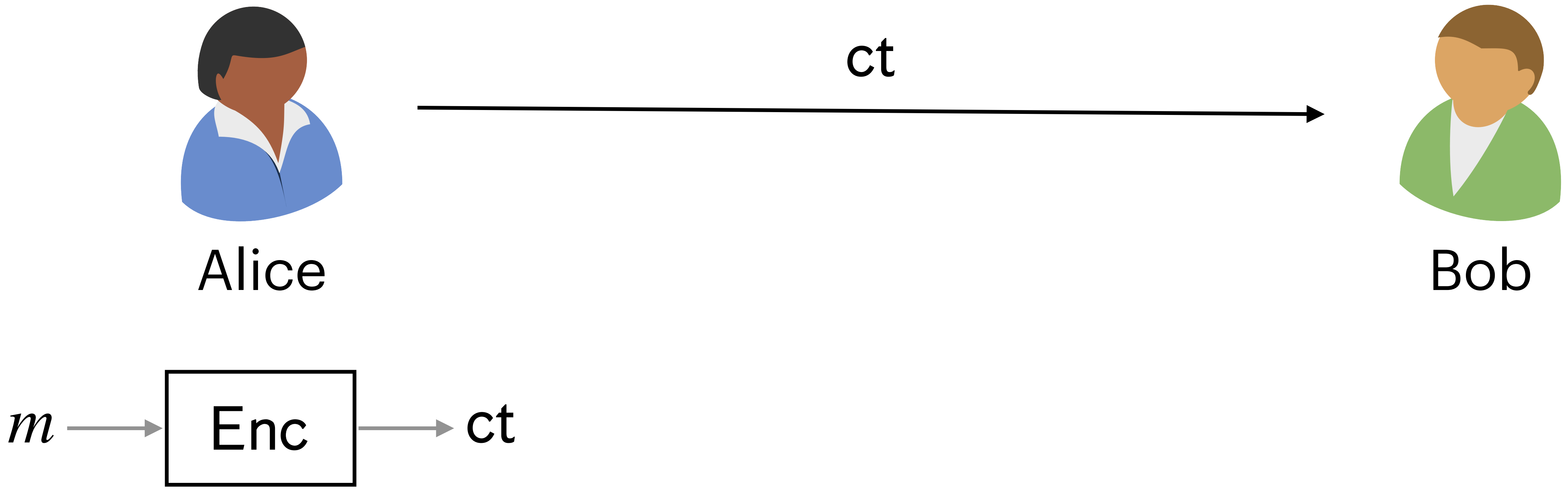
Encryption

Alice

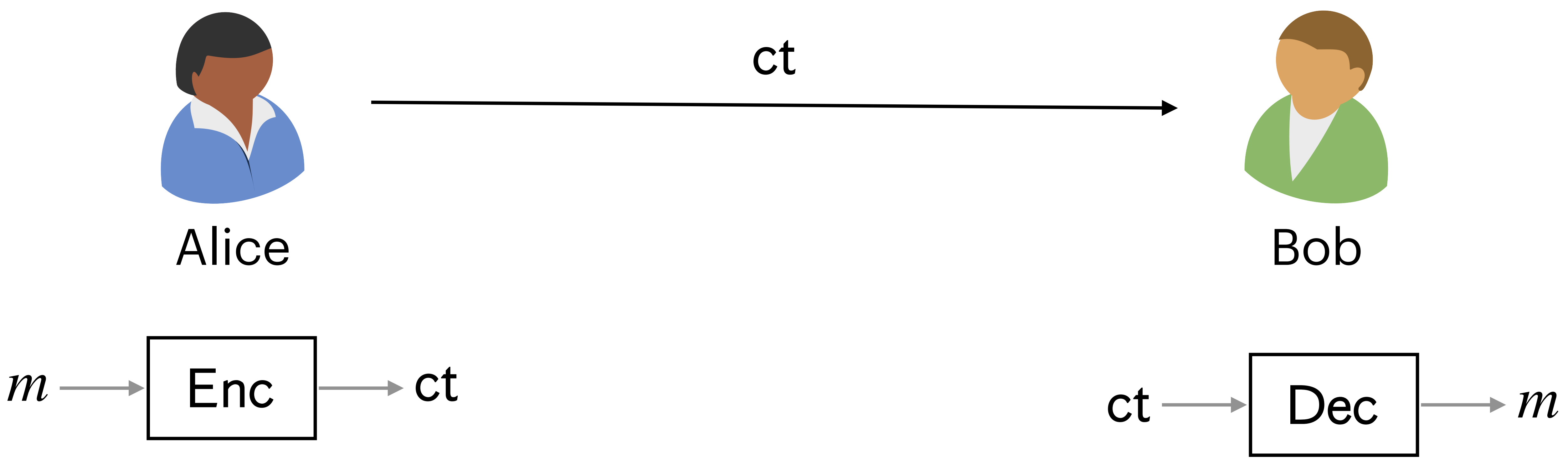
Bob

Encryption

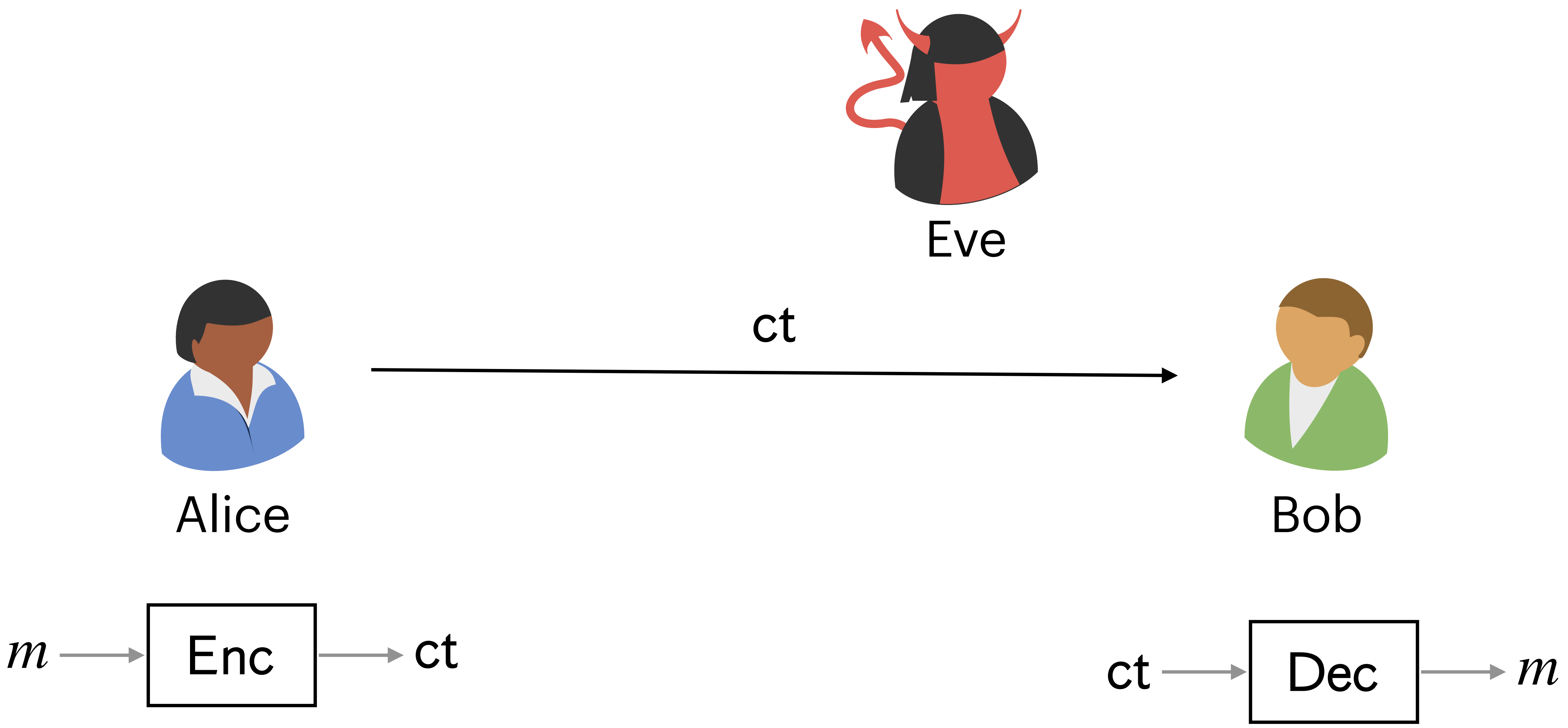
Encryption



Encryption

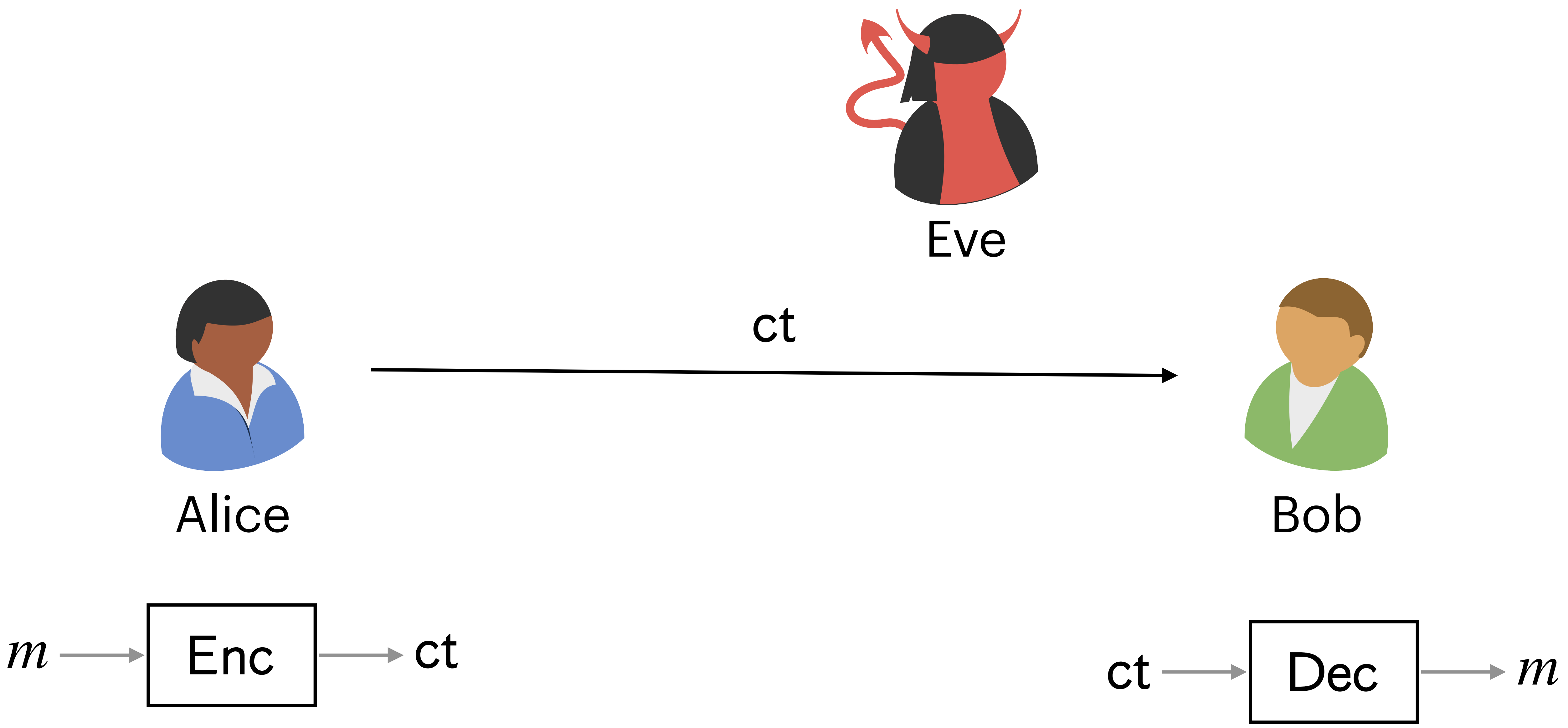


Encryption



Goal: Bob can decrypt ct but not Eve.

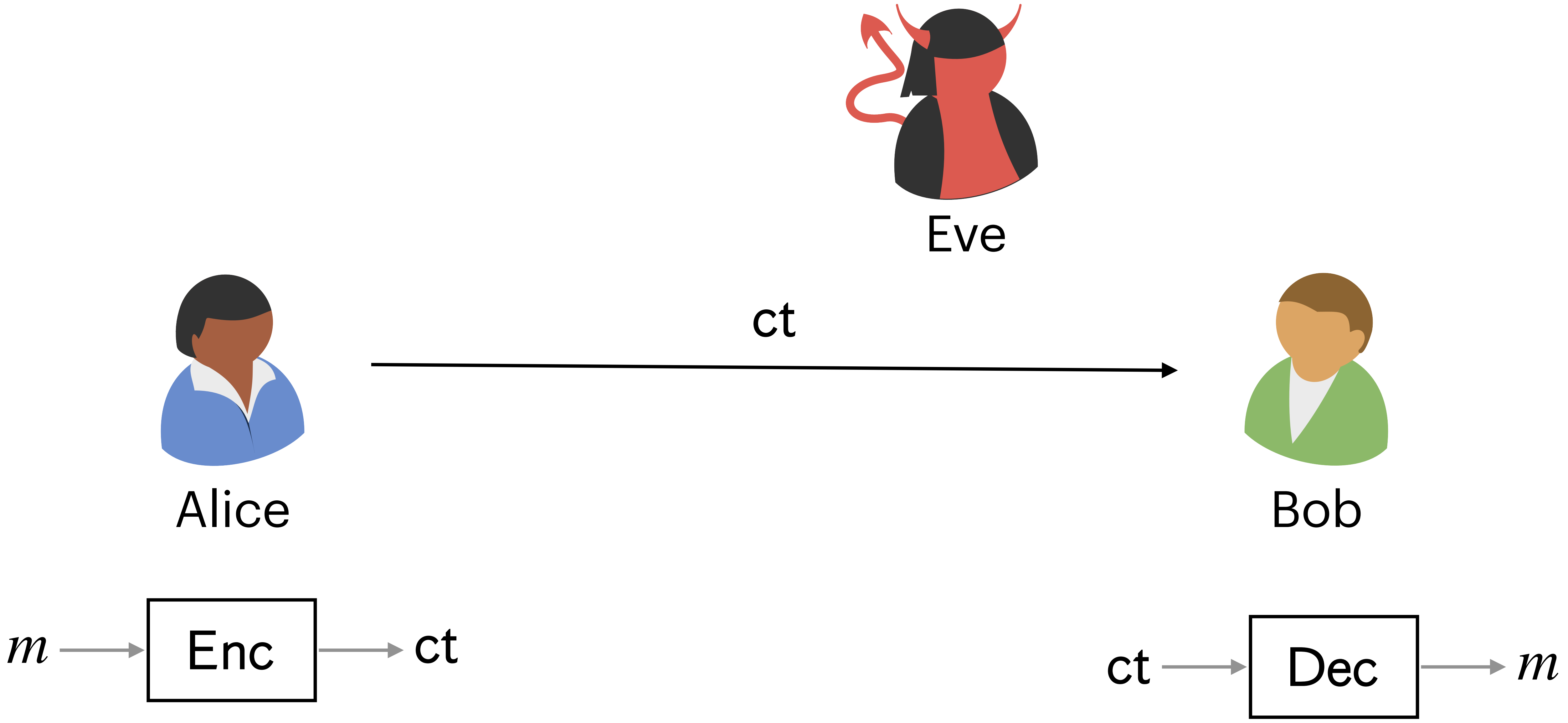
Encryption



Goal: Bob can decrypt ct but not Eve.

- Alice and Bob must have additional "information" compared to Eve.

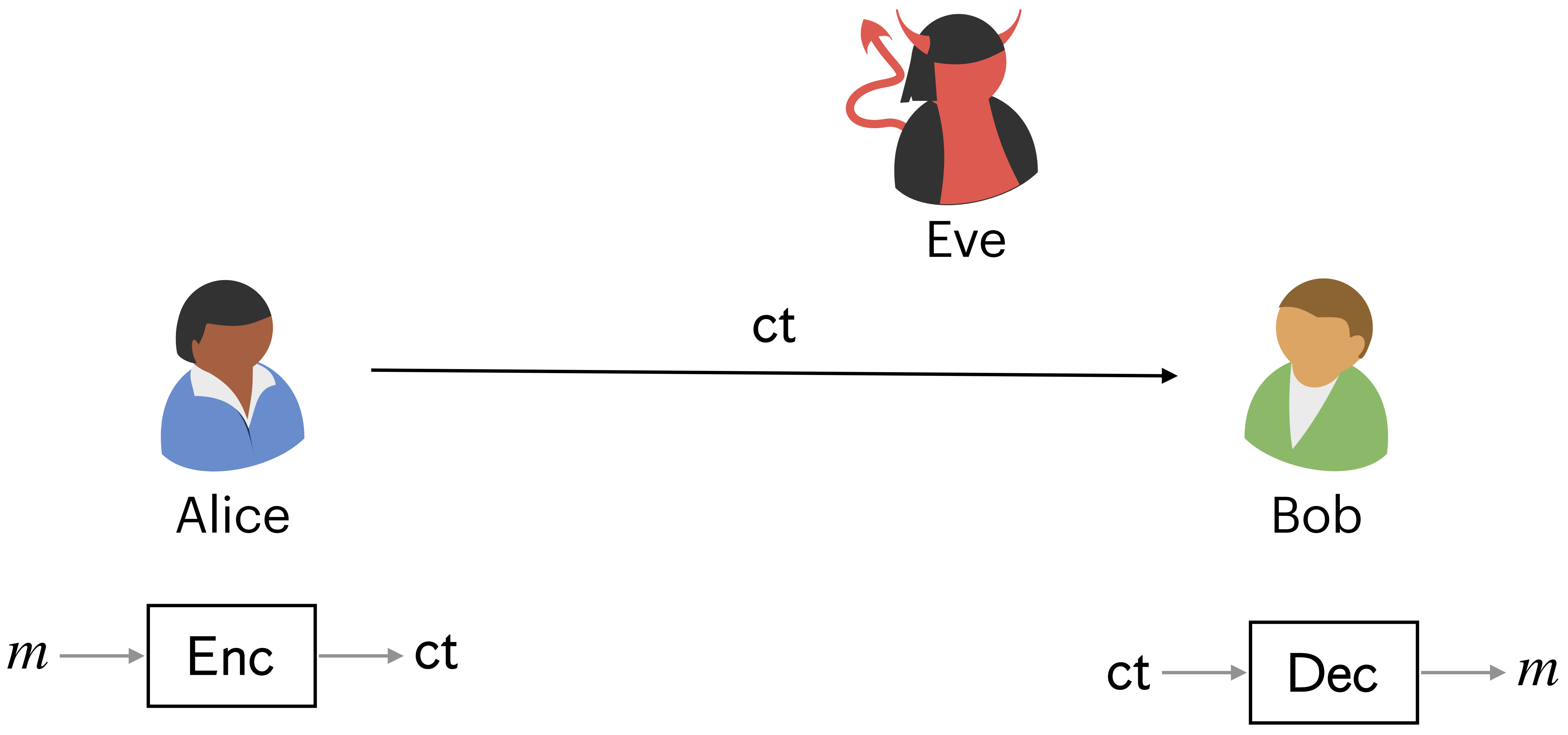
Encryption



Goal: Bob can decrypt ct but not Eve.

- Alice and Bob must have additional "information" compared to Eve.
- Should we rely on keeping the **details** of the Enc and Dec algorithms secret from Eve?

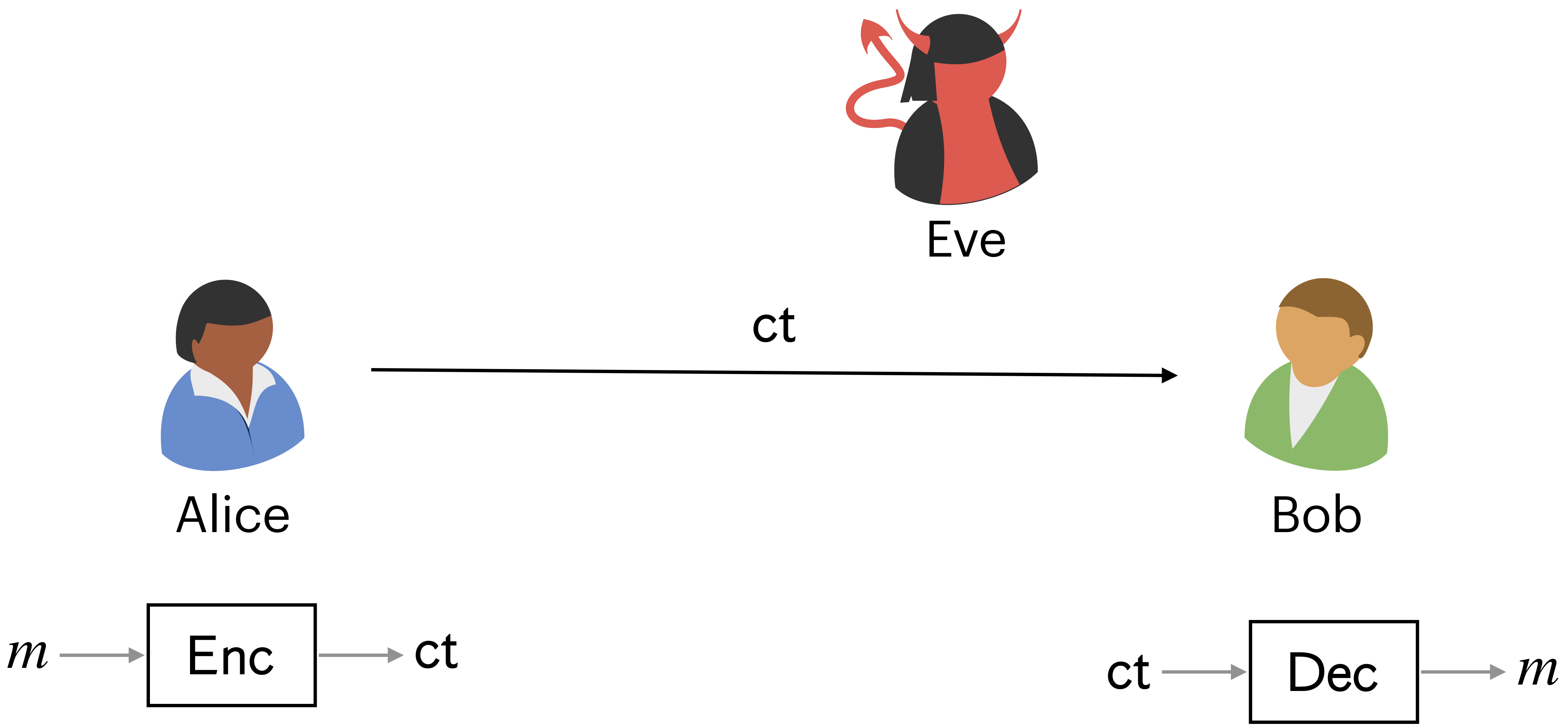
Encryption



Goal: Bob can decrypt ct but not Eve.

- Alice and Bob must have additional "information" compared to Eve.
- Should we rely on keeping the **details** of the Enc and Dec algorithms secret from Eve?
  - No! If Eve eventually learns the details of Enc and Dec, we will have to invent new algorithms.

Encryption



Goal: Bob can decrypt ct but not Eve.

- Alice and Bob must have additional "information" compared to Eve.
- Should we rely on keeping the **details** of the Enc and Dec algorithms secret from Eve?
  - No! If Eve eventually learns the details of Enc and Dec, we will have to invent new algorithms.
  - Security through obscurity is fragile and unsustainable.

Kerckhoff’s Principle

Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

Kerckhoff’s Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.

Kerckhoff's Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.
- **Secret key:** A value generated by a probabilistic (public) algorithm and kept secret from the adversary.

Kerckhoff’s Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.
- **Secret key:** A value generated by a probabilistic (public) algorithm and kept secret from the adversary.
- Advantages

Kerckhoff’s Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.
- **Secret key:** A value generated by a probabilistic (public) algorithm and kept secret from the adversary.
- Advantages
    - It is easier to change a compromised secret key than invent new algorithms.

Kerckhoff’s Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.
- **Secret key:** A value generated by a probabilistic (public) algorithm and kept secret from the adversary.
- Advantages
  - It is easier to change a compromised secret key than invent new algorithms.
  - It is easier to ensure the secrecy of a key than that of an algorithm.

Kerckhoff’s Principle

> Design your system to be secure even if the adversary has complete knowledge of all its algorithms.

- Security of encryption still requires Alice and Bob to have some secret information.
- **Secret key:** A value generated by a probabilistic (public) algorithm and kept secret from the adversary.
- Advantages
  - It is easier to change a compromised secret key than invent new algorithms.
  - It is easier to ensure the secrecy of a key than that of an algorithm.
  - Algorithms can be made public, analyzed and standardized. Crucial for large-scale deployments.

Encryption: Syntax

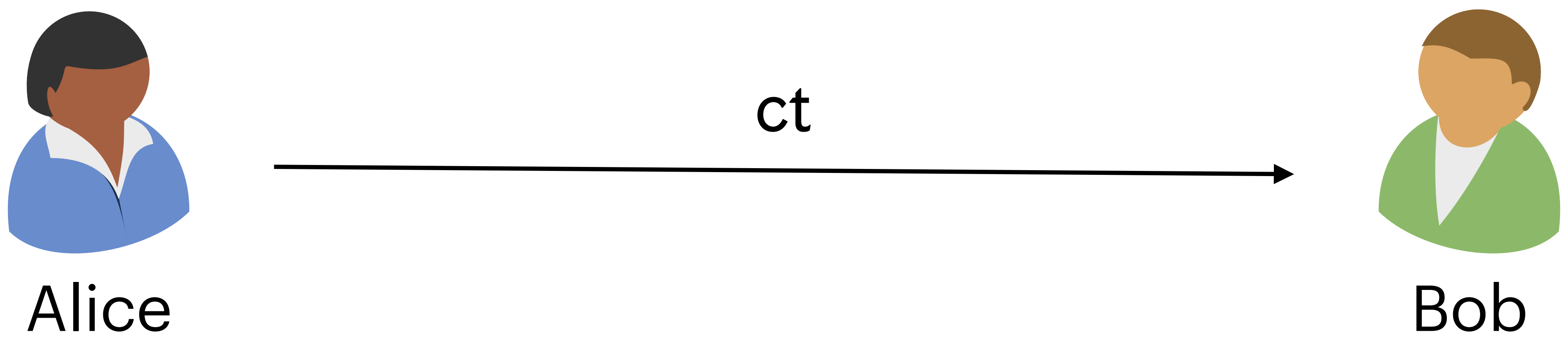
Alice

k

Bob

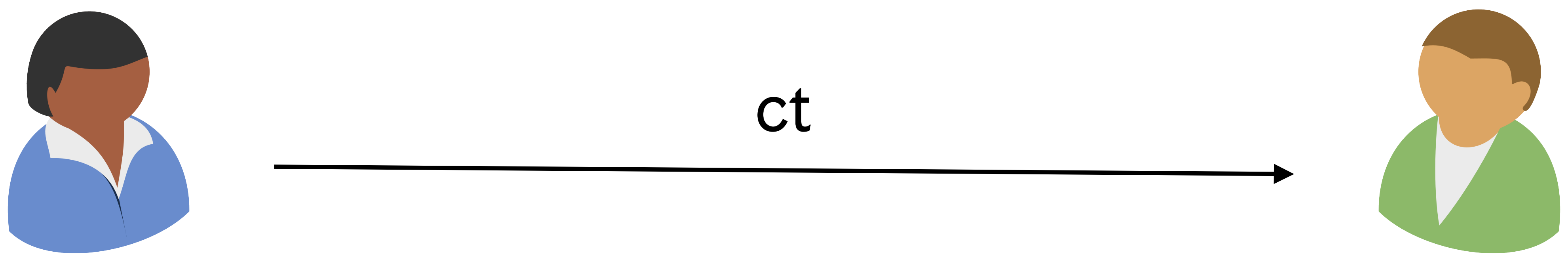
k

Encryption: Syntax



Alice	Bob
k	k
$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$	$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

Encryption: Syntax



Alice

k

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$

Bob

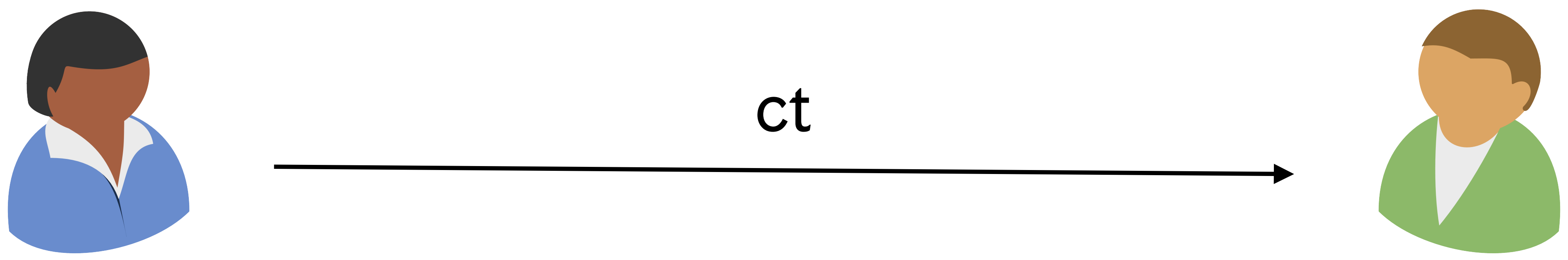
k

$m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

Encryption Scheme Syntax

An encryption scheme consists of three (possibly probabilistic) algorithms:

Encryption: Syntax



Alice | Bob

k | k

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$ | $m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

Encryption Scheme Syntax

An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \rightarrow k$ outputs a key $k \in \mathscr{K}$.

Encryption: Syntax

Alice

k

$\text{ct} \leftarrow \text{Enc}(k, m)$

ct →

Bob

k

$m \leftarrow \text{Dec}(k, \text{ct})$

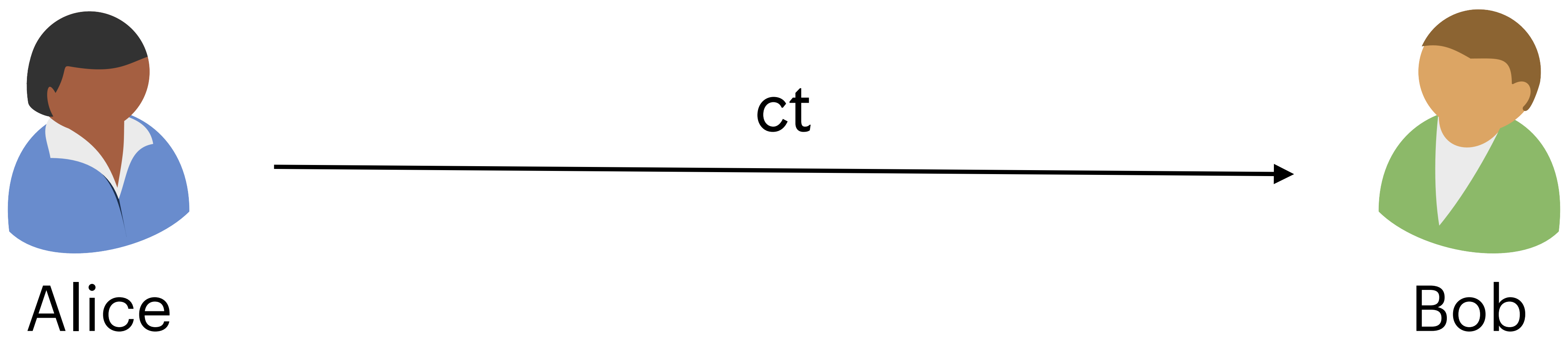
Encryption Scheme Syntax

An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\text{KeyGen}() \rightarrow k$ outputs a key $k \in \mathscr{K}$.

Key space: Set of all possible keys

Encryption: Syntax



k $\qquad$ k

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$ $\qquad$ $m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

<u>Encryption Scheme Syntax</u>

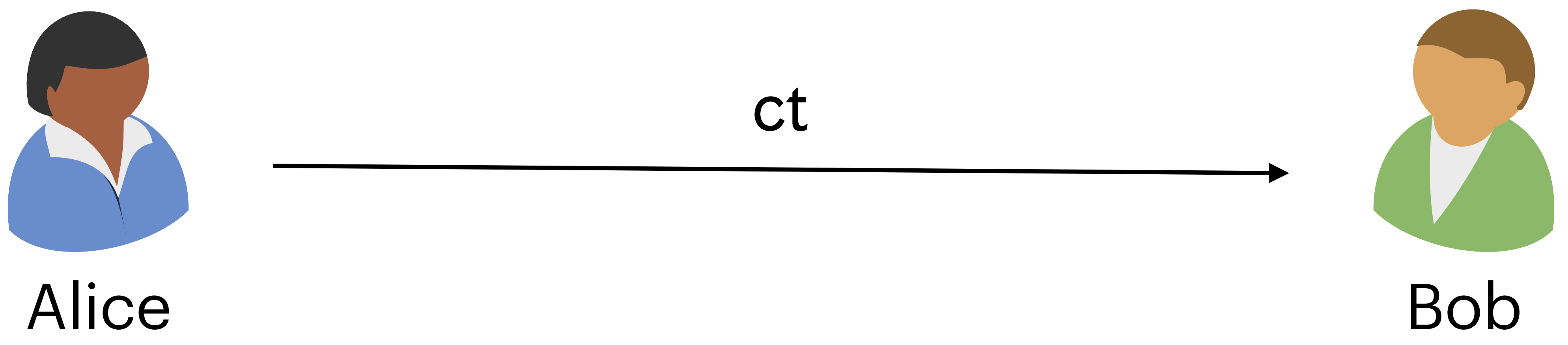
An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \rightarrow k$ outputs a key $k \in \mathscr{K}$.

Key space: Set of all possible keys

Has to be probabilistic

Encryption: Syntax



Alice: k — $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$

Bob: k — $m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

<u>Encryption Scheme Syntax</u>

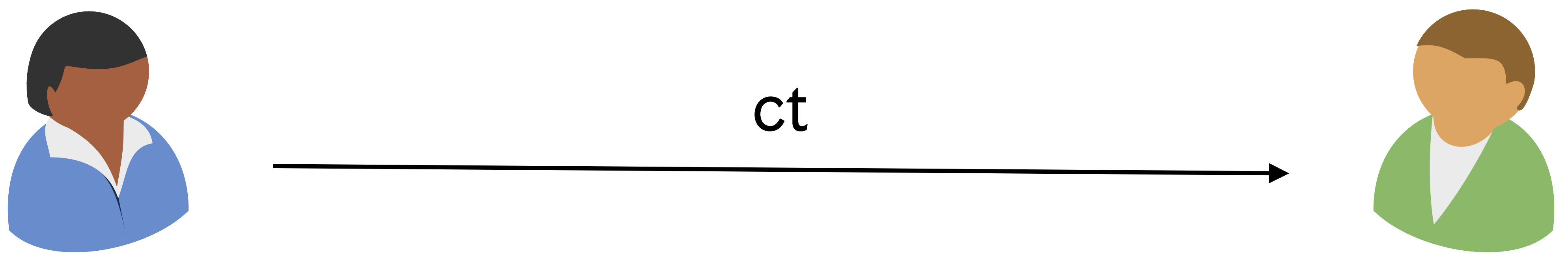
An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \to k$ outputs a key $k \in \mathscr{K}$.
- $\mathsf{Enc}(k, m) \to \mathsf{ct}$ takes key k and message $m \in \mathscr{M}$ and outputs ciphertext $\mathsf{ct} \in \mathscr{C}$.

Message space

Ciphertext space

Encryption: Syntax



Alice | Bob

k | k

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$ | $m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

<u>Encryption Scheme Syntax</u>

An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \rightarrow k$ outputs a key $k \in \mathscr{K}$.
- $\mathsf{Enc}(k, m) \rightarrow \mathsf{ct}$ takes key k and message $m \in \mathscr{M}$ and outputs ciphertext $\mathsf{ct} \in \mathscr{C}$.
- $\mathsf{Dec}(k, \mathsf{ct}) \rightarrow m$ takes key k and ciphertext ct and outputs message m.

Encryption: Syntax

Alice

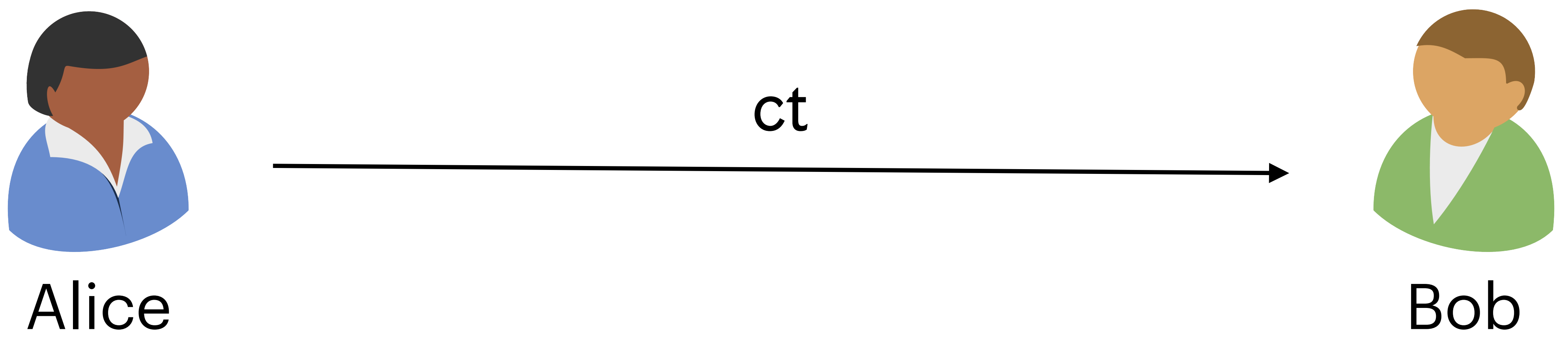
k

Bob

k

Adversarial Model: Eve is “passive”. She reads ciphertexts but does not interfere.

Encryption: Syntax

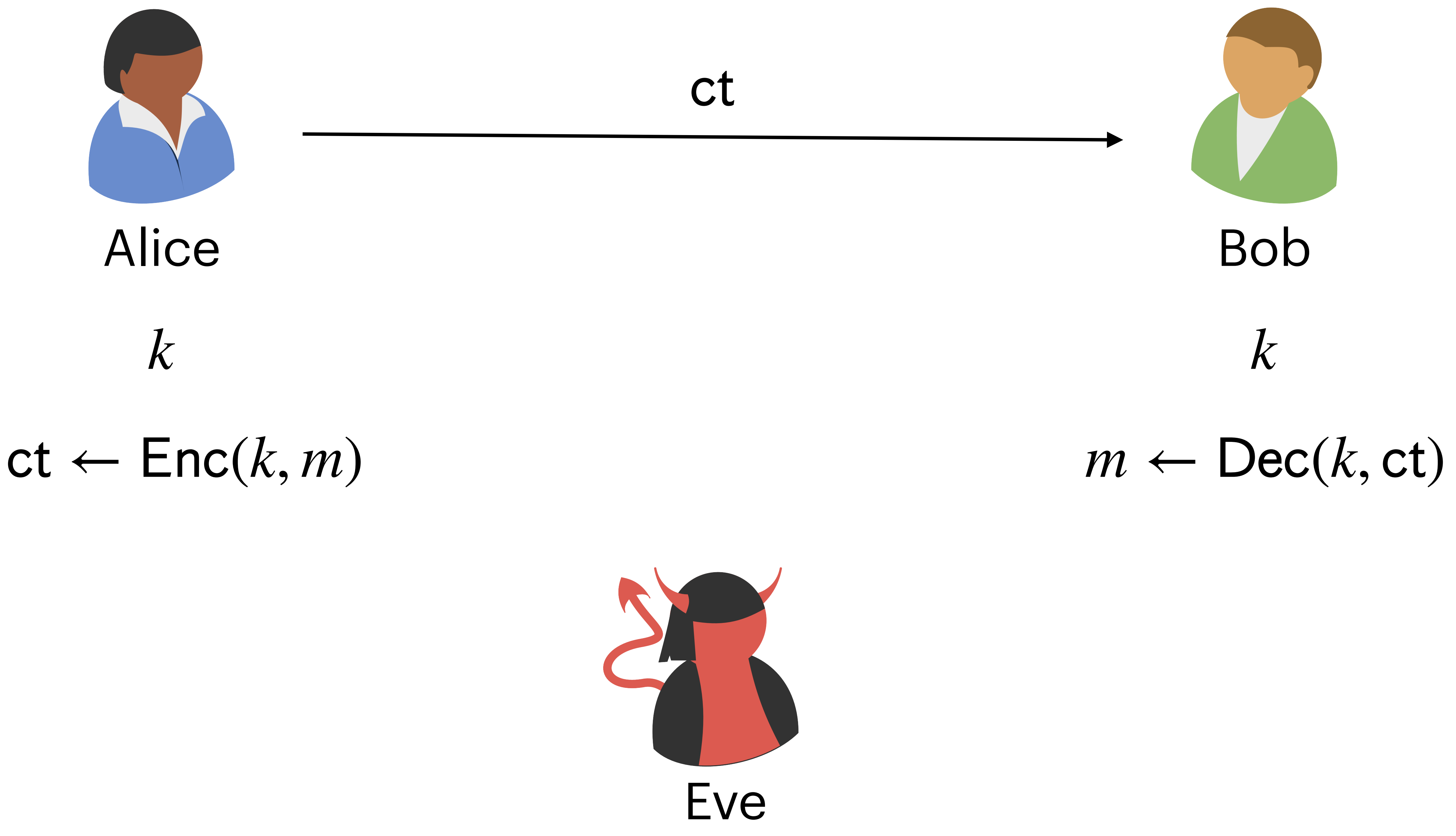


k k

$\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$ $m \leftarrow \mathsf{Dec}(k, \mathsf{ct})$

Adversarial Model: Eve is “passive”. She reads ciphertexts but does not interfere.

Encryption: Syntax



Adversarial Model: Eve is “passive”. She reads ciphertexts but does not interfere.

What are we (not) trying to do?

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).
- We are not guaranteeing that Bob will necessarily receive the ciphertext.

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).
- We are not guaranteeing that Bob will necessarily receive the ciphertext.
- We are assuming Eve is passive and cannot tamper with the ciphertext. We will consider an active Eve later in the course.

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).
- We are not guaranteeing that Bob will necessarily receive the ciphertext.
- We are assuming Eve is passive and cannot tamper with the ciphertext. We will consider an active Eve later in the course.
- For now, we will ignore the issue of how Alice and Bob obtain a common secret key in the first place. Later in the course, we will look at key exchange.

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).
- We are not guaranteeing that Bob will necessarily receive the ciphertext.
- We are assuming Eve is passive and cannot tamper with the ciphertext. We will consider an active Eve later in the course.
- For now, we will ignore the issue of how Alice and Bob obtain a common secret key in the first place. Later in the course, we will look at key exchange.
- We are assuming that keys can be kept private in a reliable manner. We are not discussing how to do key management.

What are we (not) trying to do?

- We are not trying to hide the existence of private communication (aka steganography).
- We are not guaranteeing that Bob will necessarily receive the ciphertext.
- We are assuming Eve is passive and cannot tamper with the ciphertext. We will consider an active Eve later in the course.
- For now, we will ignore the issue of how Alice and Bob obtain a common secret key in the first place. Later in the course, we will look at key exchange.
- We are assuming that keys can be kept private in a reliable manner. We are not discussing how to do key management.
- **Simplification:** We will focus on the case of encrypting a single message. We will consider multi-message security later in the course.

One-Time Pad

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

One-Time Pad

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

Set of all λ-bit strings

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

One-Time Pad

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

Set of all λ-bit strings

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Sampling uniformly at random from the set

One-Time Pad: Correctness

One-Time Pad

Let λ be a positive integer and let $\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Correctness (Intuitive): Does decrypting the ciphertext yield the intended plaintext?

One-Time Pad: Correctness

One-Time Pad

Let λ be a positive integer and let $\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Correctness (Intuitive): Does decrypting the ciphertext yield the intended plaintext?

One-Time Pad Correctness

One-Time Pad: Correctness

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus \text{ct}$.

Correctness (Intuitive): Does decrypting the ciphertext yield the intended plaintext?

One-Time Pad Correctness

Claim: $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$ we have Dec$(k, \text{Enc}(k, m)) = m$.

One-Time Pad: Correctness

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Correctness (Intuitive): Does decrypting the ciphertext yield the intended plaintext?

One-Time Pad Correctness

Claim: $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$ we have $\text{Dec}(k, \text{Enc}(k, m)) = m$.

Proof: Fix arbitrary $k \in \mathcal{K}$ and $m \in \mathcal{M}$. We have

$$\begin{aligned}\text{Dec}(k, \text{Enc}(k, m)) &= \text{Dec}(k, k \oplus m) \\ &= k \oplus k \oplus m \\ &= m\,.\end{aligned}$$

One-Time Pad: Security

One-Time Pad

Let λ be a positive integer and let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Security (Intuitive): The ciphertext does not reveal any information about the plaintext to Eve, no matter what she does with the ciphertext.

One-Time Pad: Security

One-Time Pad

Let λ be a positive integer and let $\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Security (Intuitive): The ciphertext does not reveal any information about the plaintext to Eve, no matter what she does with the ciphertext.

We do not assume anything about the adversary's strategy.

One-Time Pad: Security

One-Time Pad

Let λ be a positive integer and let $\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^\lambda$.

- KeyGen(): $k \xleftarrow{\$} \{0,1\}^\lambda$.
- Enc(k, m): ct $:= k \oplus m$.
- Dec(k, ct): $m := k \oplus$ ct.

Security (Intuitive): The ciphertext does not reveal any information about the plaintext to Eve, no matter what she does with the ciphertext.

We do not assume anything about the adversary's strategy.

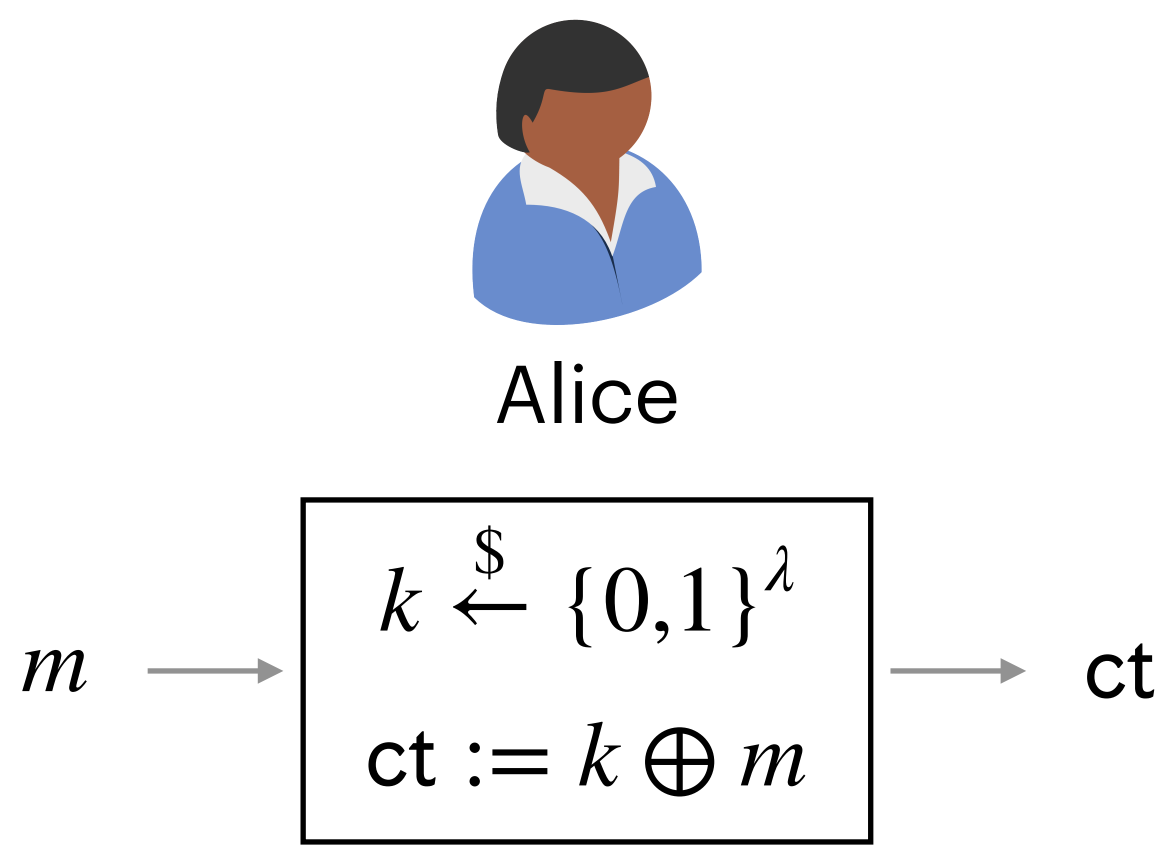
Why is one-time pad secure?

One-Time Pad: Security

- Let us analyze Eve’s view to understand why the scheme is secure.

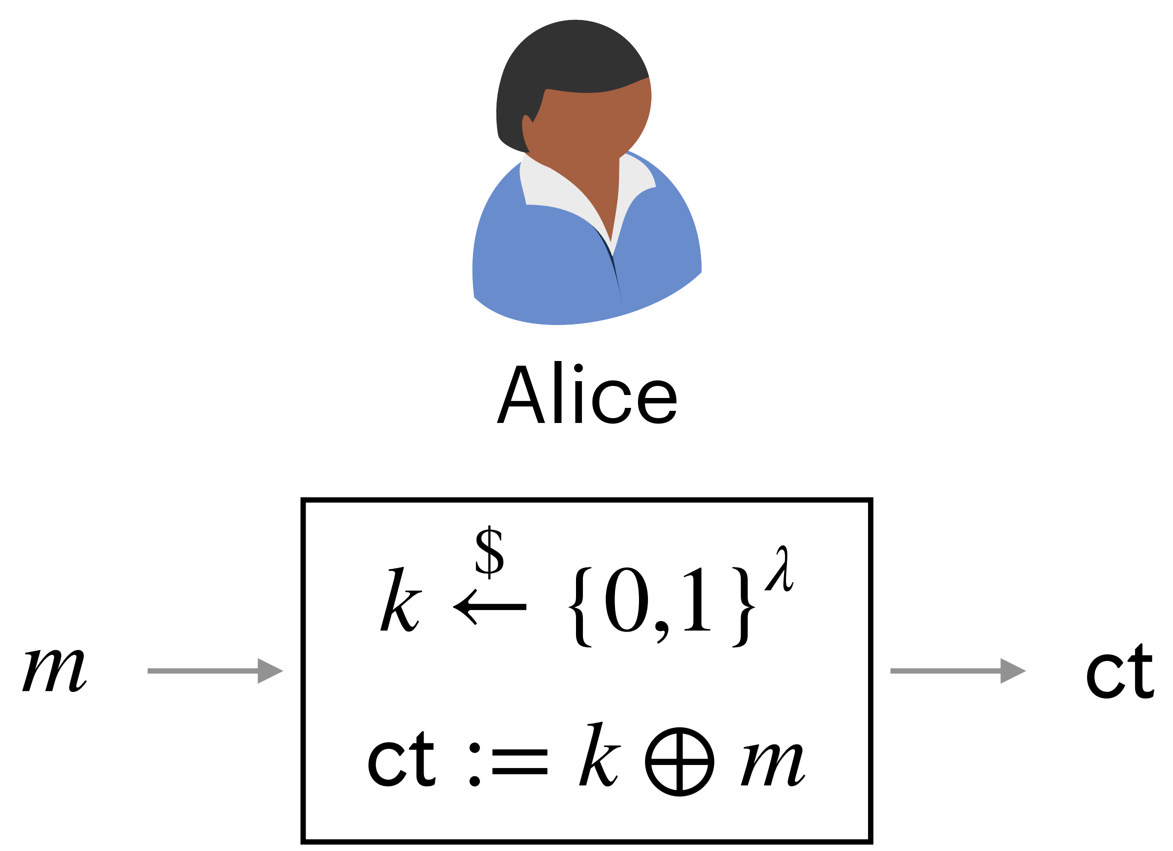
One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.



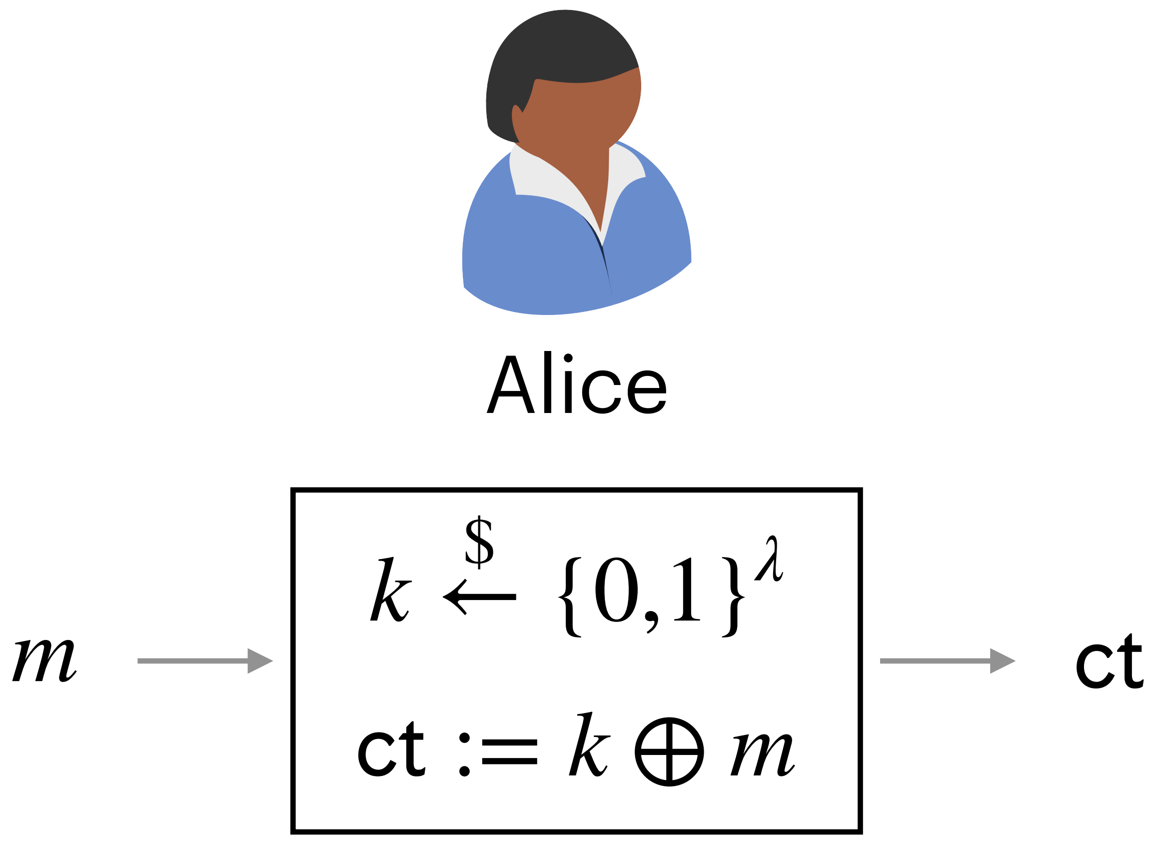
One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^{\lambda}$.
  - Let us analyze the resulting ciphertext distribution.



One-Time Pad: Security

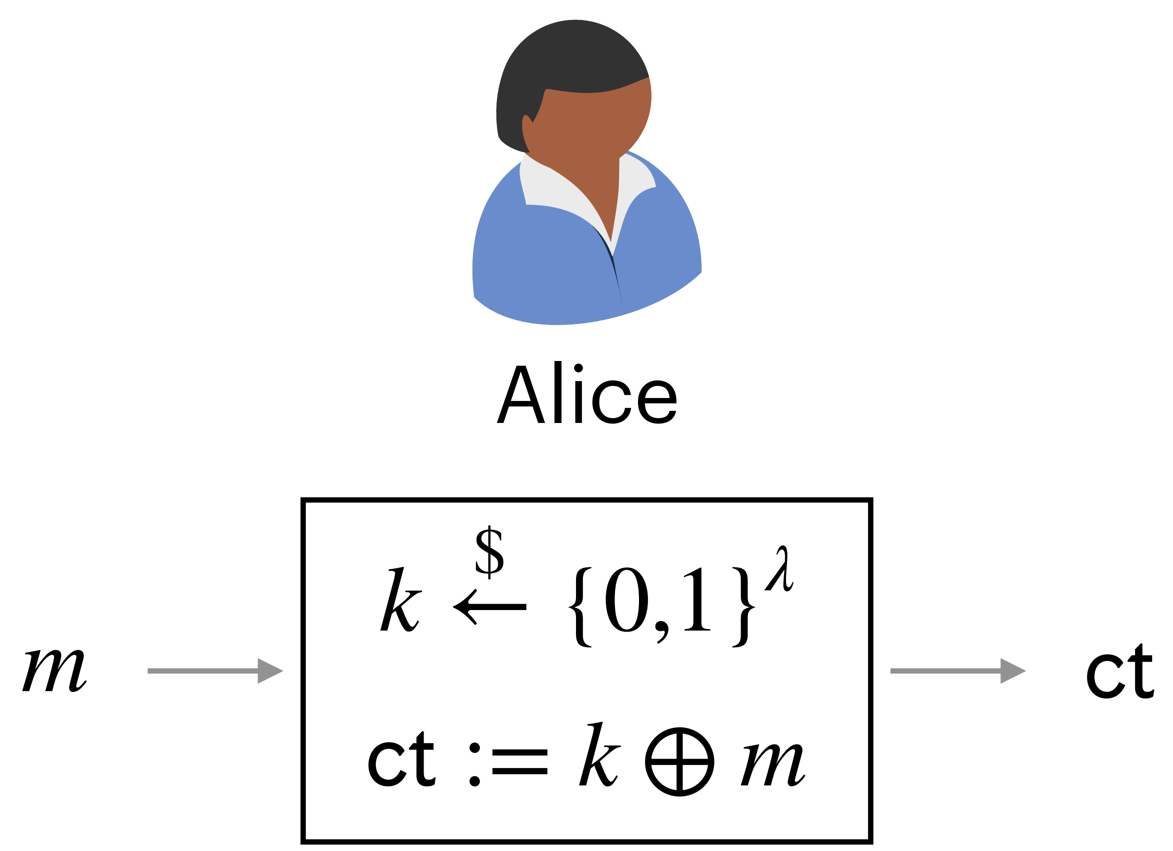
- Let us analyze Eve's view to understand why the scheme is secure.
    - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \stackrel{\$}{\leftarrow} \{0,1\}^\lambda$.
    - Let us analyze the resulting ciphertext distribution.



- **Concrete example:** $\lambda = 3$ and $m = 010$

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.



- **Concrete example:** $\lambda = 3$ and $m = 010$

Pr	k	ct $= k \oplus 010$
1/8	000	010
1/8	001	011
1/8	010	000
1/8	011	001
1/8	100	110
1/8	101	111
1/8	110	100
1/8	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext ct is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

Alice

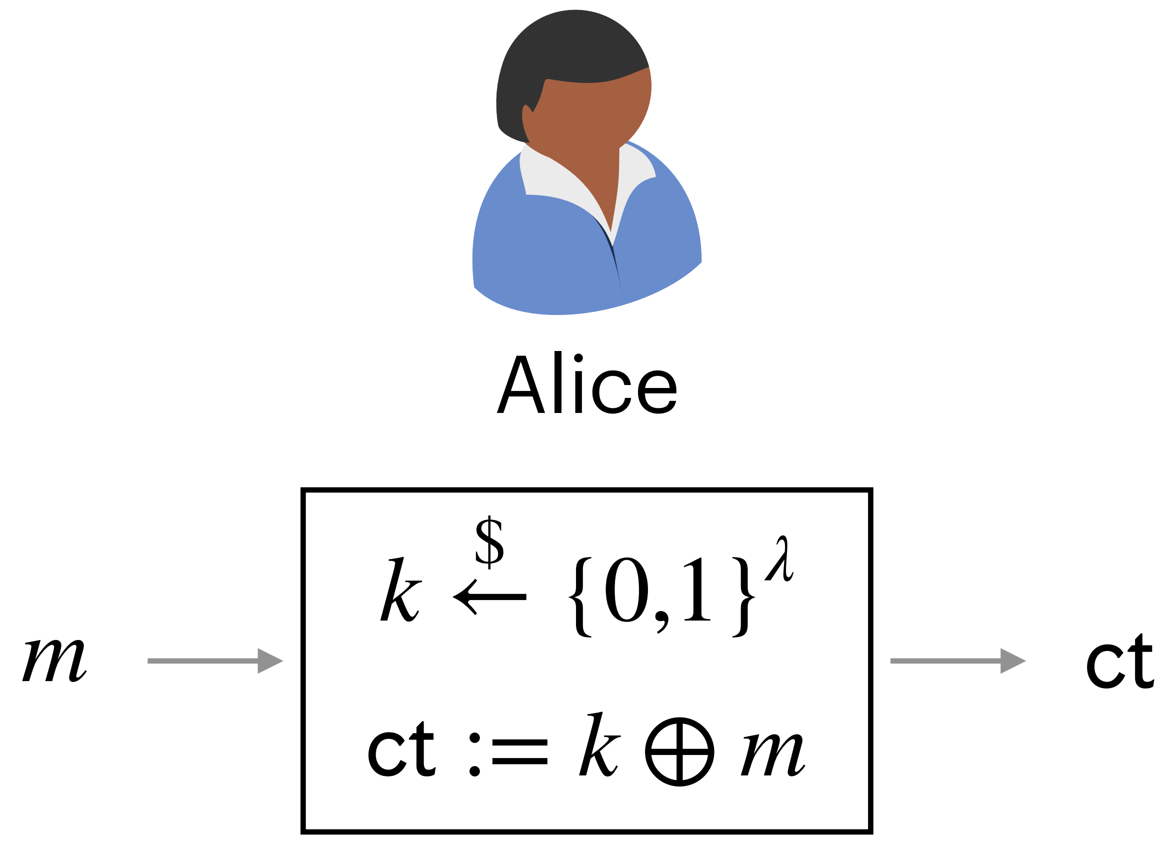
$m \longrightarrow$ [$k \xleftarrow{\$} \{0,1\}^\lambda$; ct $:= k \oplus m$] $\longrightarrow$ ct

- **Concrete example:** $\lambda = 3$ and $m = 010$
  - Every string in $\{0,1\}^3$ occurs exactly once as a ciphertext.

Pr	k	ct $= k \oplus 010$
1/8	000	010
1/8	001	011
1/8	010	000
1/8	011	001
1/8	100	110
1/8	101	111
1/8	110	100
1/8	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

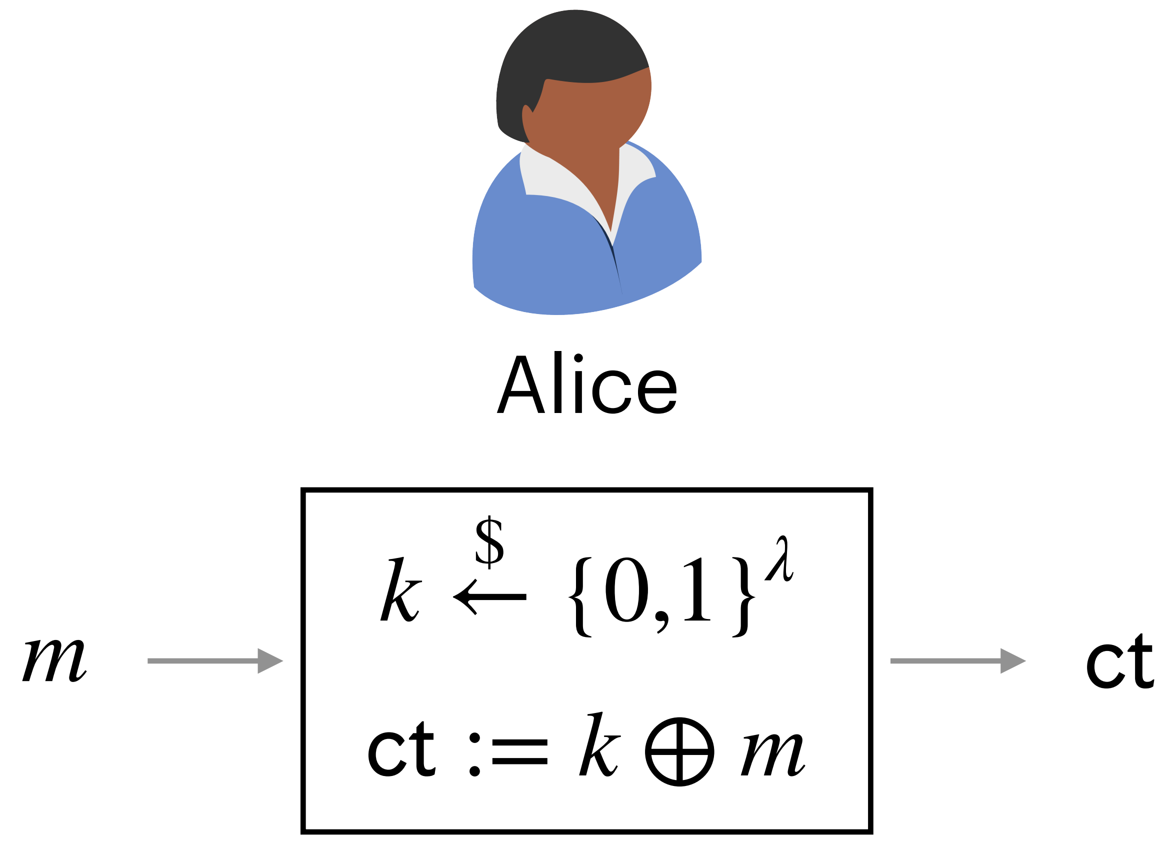


- **Concrete example:** $\lambda = 3$ and $m = 010$
  - Every string in $\{0,1\}^3$ occurs exactly once as a ciphertext.
  - Since the key is sampled uniformly at random, for any $s \in \{0,1\}^3$, the probability that **ct** $= s$ is $1/8$ i.e., **the ciphertext is uniformly random** over $\{0,1\}^3$.

Pr	k	ct $= k \oplus 010$
$1/8$	000	010
$1/8$	001	011
$1/8$	010	000
$1/8$	011	001
$1/8$	100	110
$1/8$	101	111
$1/8$	110	100
$1/8$	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

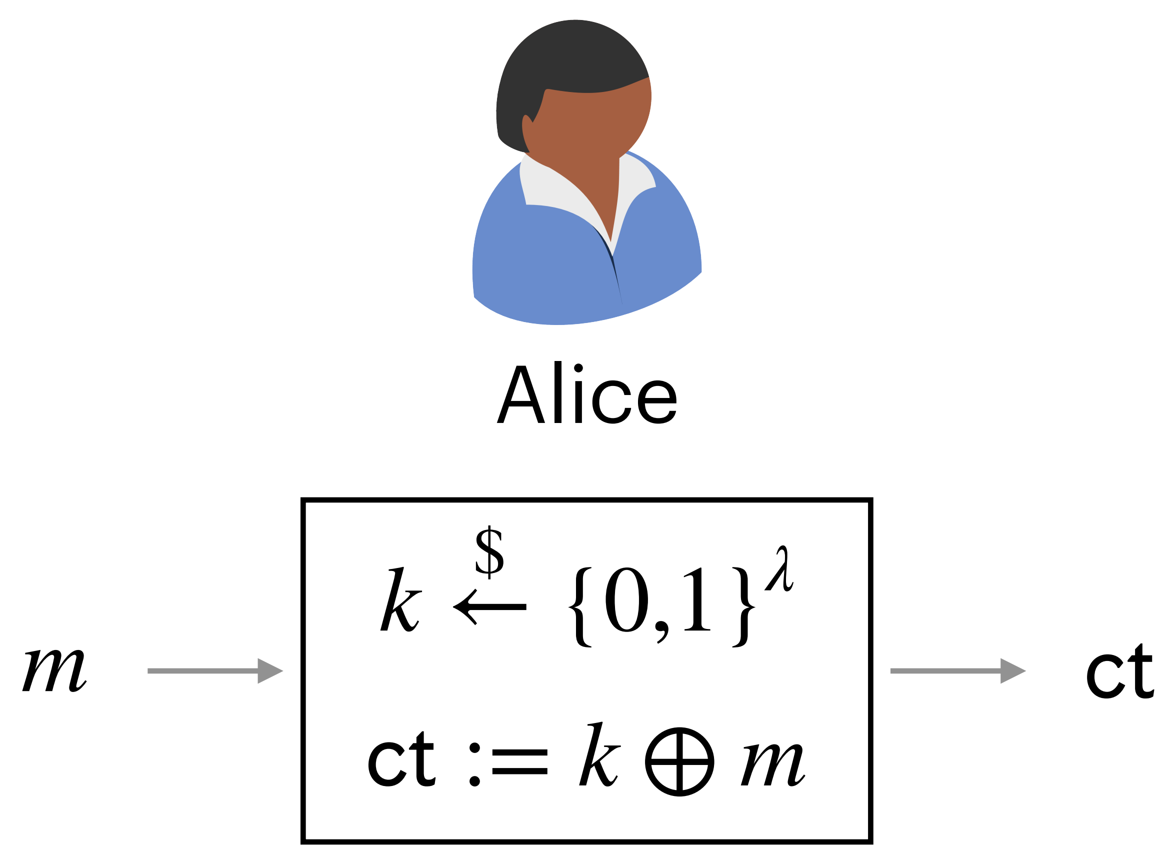


- **Concrete example:** $\lambda = 3$ and $m = 010$
  - Every string in $\{0,1\}^3$ occurs exactly once as a ciphertext.
  - Since the key is sampled uniformly at random, for any $s \in \{0,1\}^3$, the probability that **ct** $= s$ is $1/8$ i.e., **the ciphertext is uniformly random** over $\{0,1\}^3$.
  - True for any $m \in \{0,1\}^3$

Pr	k	ct $= k \oplus 010$
1/8	000	010
1/8	001	011
1/8	010	000
1/8	011	001
1/8	100	110
1/8	101	111
1/8	110	100
1/8	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^{\lambda}$.
  - Let us analyze the resulting ciphertext distribution.



- **Concrete example:** $\lambda = 3$ and $m = 010$
  - Every string in $\{0,1\}^3$ occurs exactly once as a ciphertext.
  - Since the key is sampled uniformly at random, for any $s \in \{0,1\}^3$, the probability that ct $= s$ is $1/8$ i.e., **the ciphertext is uniformly random** over $\{0,1\}^3$.
  - True for any $m \in \{0,1\}^3$

Pr	k	ct $= k \oplus 010$
1/8	000	010
1/8	001	011
1/8	010	000
1/8	011	001
1/8	100	110
1/8	101	111
1/8	110	100
1/8	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message

- **Concrete example:** $\lambda = 3$ and $m = 010$
  - Every string in $\{0,1\}^3$ occurs exactly once as a ciphertext.
  - Since the key is sampled uniformly at random, for any $s \in \{0,1\}^3$, the probability that **ct** $= s$ is $1/8$ i.e., **the ciphertext is uniformly random** over $\{0,1\}^3$.
  - True for any $m \in \{0,1\}^3$

Alice

$m \rightarrow$ [$k \xleftarrow{\$} \{0,1\}^\lambda$; ct $:= k \oplus m$] $\rightarrow$ ct

Pr	k	ct $= k \oplus 010$
1/8	000	010
1/8	001	011
1/8	010	000
1/8	011	001
1/8	100	110
1/8	101	111
1/8	110	100
1/8	111	101

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message

Alice

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^{\lambda}$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message


Alice

m

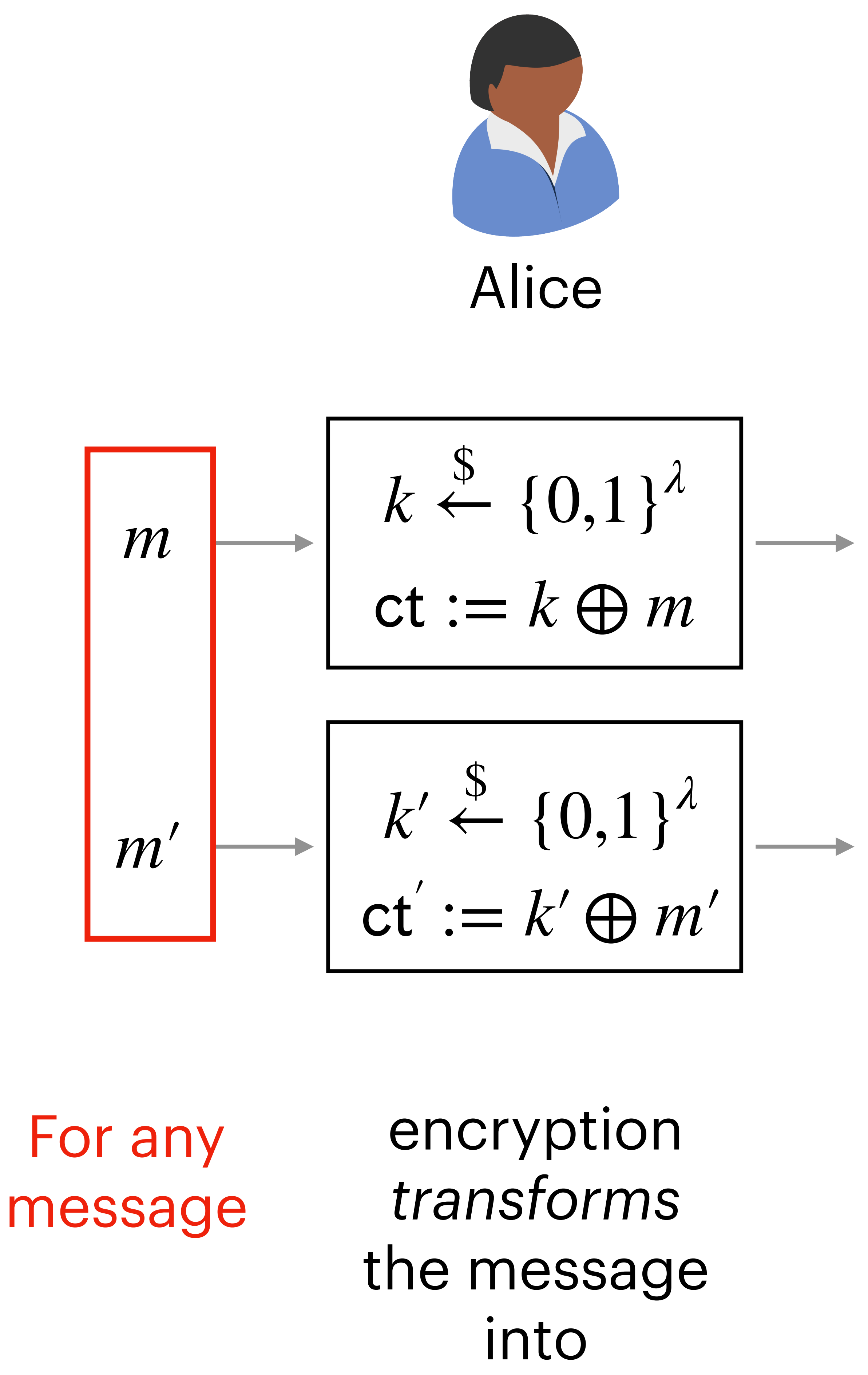
m'

For any message

One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext ct is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

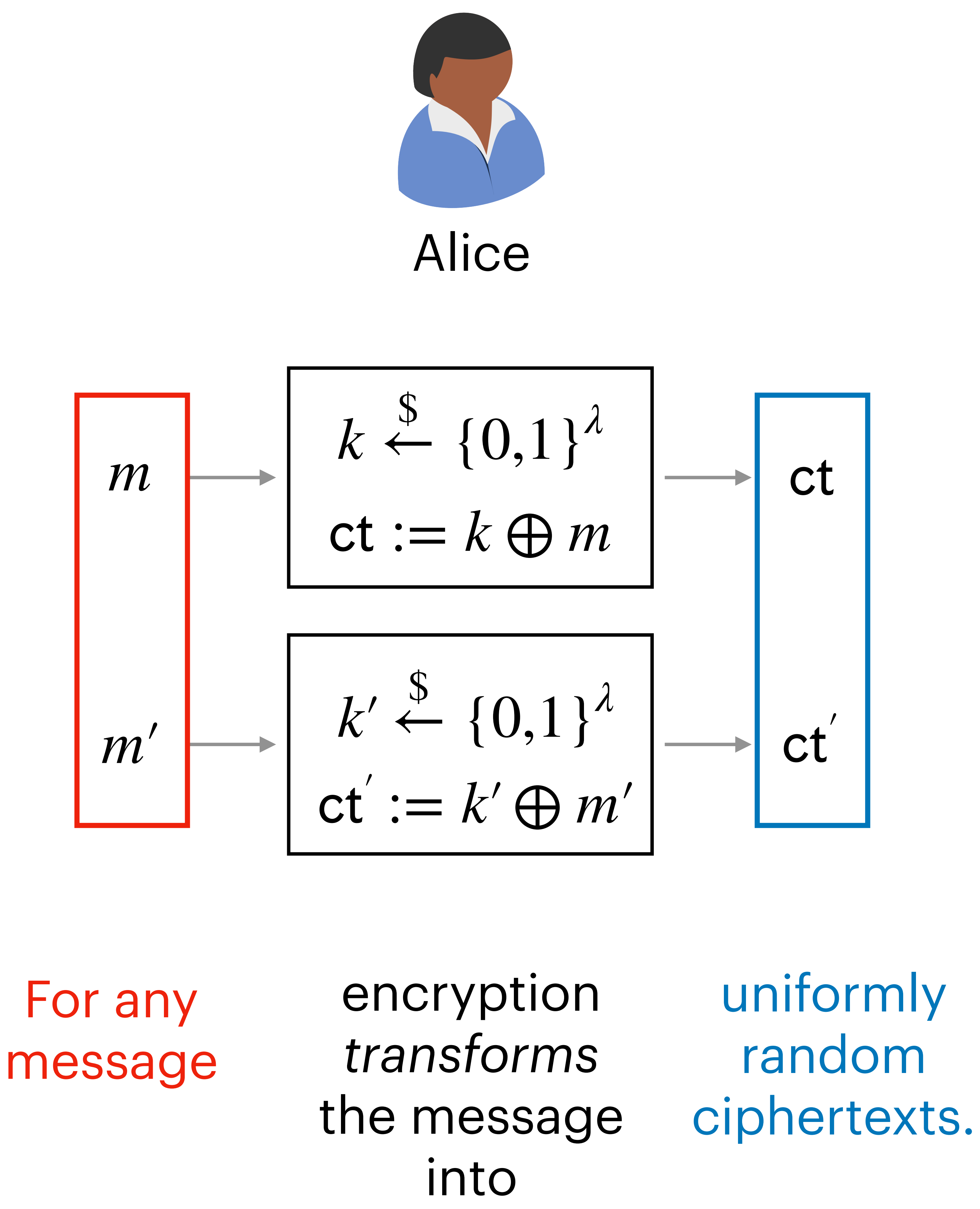
The ciphertext is uniformly distributed,
irrespective of the message



One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message

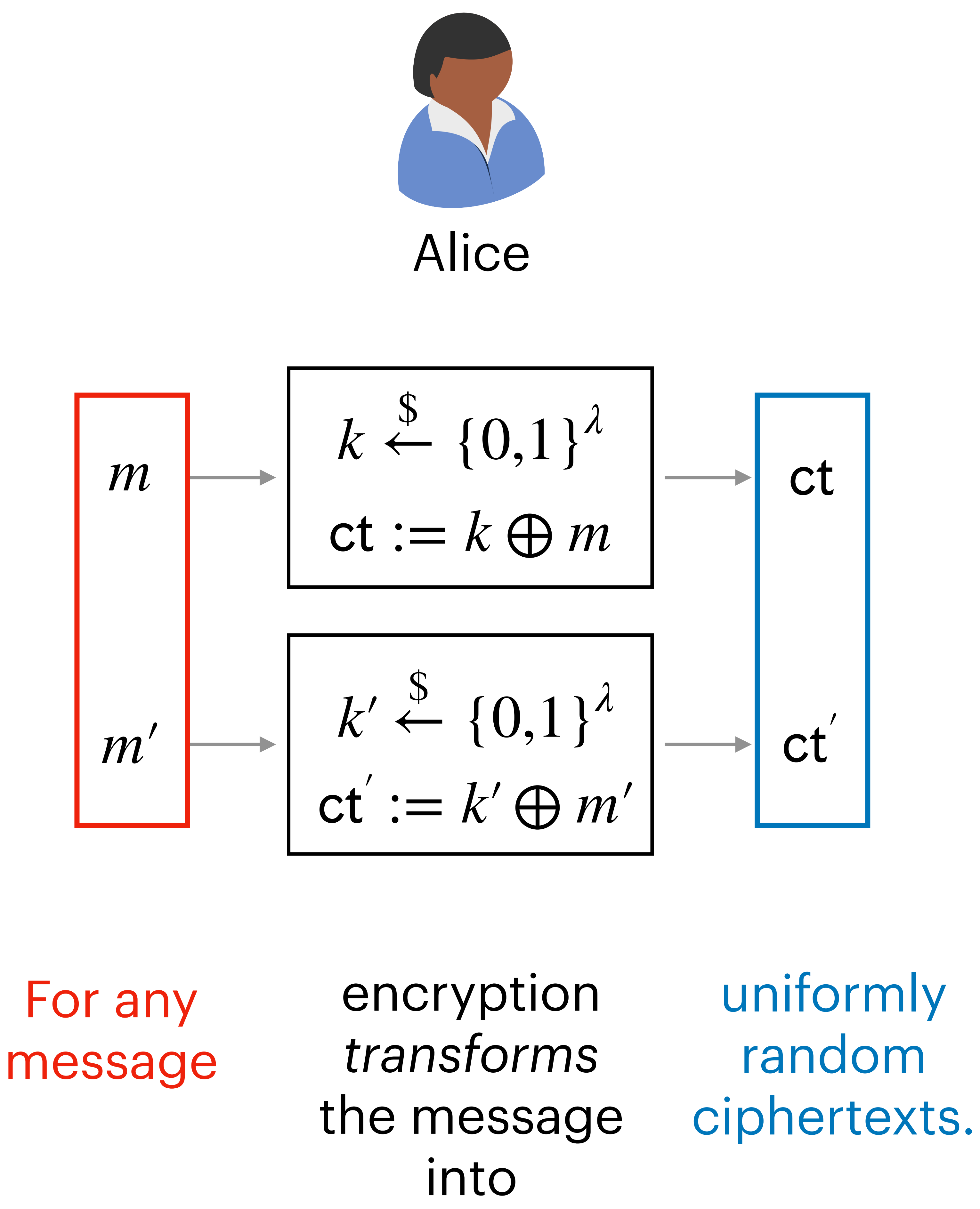


One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message

- If the ciphertext is always uniformly random then it cannot carry any information about the message!
  - Obtaining the ciphertext is useless to Eve. She can sample from this distribution herself, without knowing the message.

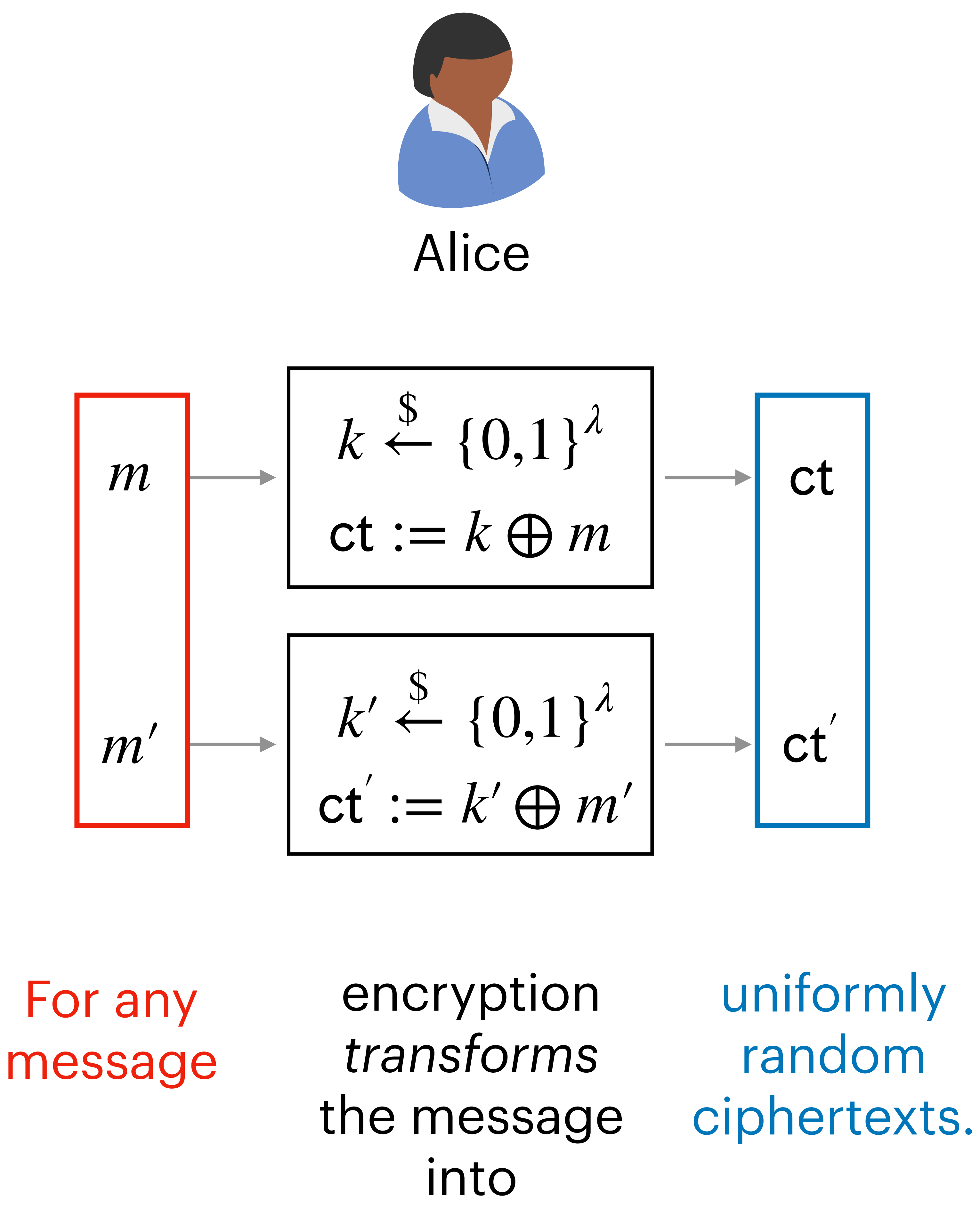


One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext **ct** is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

> The ciphertext is uniformly distributed,
> irrespective of the message

- If the ciphertext is always uniformly random then it cannot carry any information about the message!
  - Obtaining the ciphertext is useless to Eve. She can sample from this distribution herself, without knowing the message.
- Paradox? How can the ciphertext decrypt to the correct message if it does not carry any information?

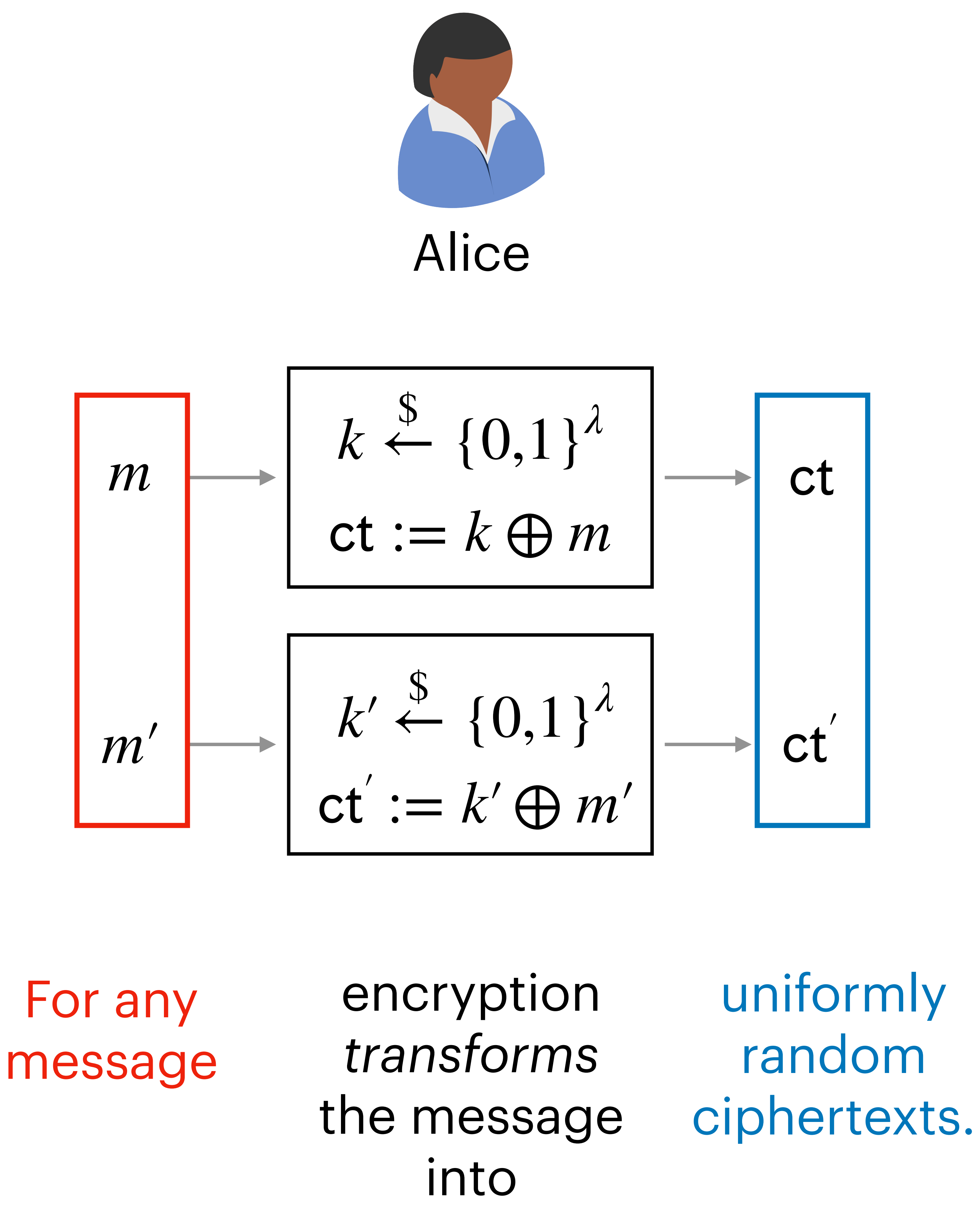


One-Time Pad: Security

- Let us analyze Eve's view to understand why the scheme is secure.
  - From Eve's viewpoint, the ciphertext ct is a *transformation of* the message m by XORing it with a uniformly random key $k \xleftarrow{\$} \{0,1\}^\lambda$.
  - Let us analyze the resulting ciphertext distribution.

The ciphertext is uniformly distributed, irrespective of the message

- If the ciphertext is always uniformly random then it cannot carry any information about the message!
  - Obtaining the ciphertext is useless to Eve. She can sample from this distribution herself, without knowing the message.
- Paradox? How can the ciphertext decrypt to the correct message if it does not carry any information?
  - Eve's view does not include the secret key!



Basics of Provable Security

- We want to generalize and formalize our intuition of the properties of OTP so that they hold for any encryption scheme.

Basics of Provable Security

- We want to generalize and formalize our intuition of the properties of OTP so that they hold for any encryption scheme.
- Two types of properties

Basics of Provable Security

- We want to generalize and formalize our intuition of the properties of OTP so that they hold for any encryption scheme.
- Two types of properties
  - Ones that should hold in the absence of an attacker e.g., correctness

Basics of Provable Security

- We want to generalize and formalize our intuition of the properties of OTP so that they hold for any encryption scheme.
- Two types of properties
  - Ones that should hold in the absence of an attacker e.g., correctness
  - Ones that specify what can happen to a system in the presence of an attacker e.g., security.

Basics of Provable Security

- We want to generalize and formalize our intuition of the properties of OTP so that they hold for any encryption scheme.
- Two types of properties
  - Ones that should hold in the absence of an attacker e.g., correctness
  - Ones that specify what can happen to a system in the presence of an attacker e.g., security.
- **Eventual Goal:** Write formal definitions to capture all required properties from any given system.

Encryption: Correctness

<u>Encryption Scheme Syntax</u>

An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \to k$ outputs a key $k \in \mathscr{K}$.
- $\mathsf{Enc}(k, m) \to \mathsf{ct}$ takes key k and message $m \in \mathscr{M}$ and outputs ciphertext $\mathsf{ct} \in \mathscr{C}$.
- $\mathsf{Dec}(k, \mathsf{ct}) \to m$ takes key k and ciphertext ct and outputs message m.

Encryption: Correctness

Encryption Scheme Syntax

An encryption scheme consists of three (possibly probabilistic) algorithms:

- $\mathsf{KeyGen}() \to k$ outputs a key $k \in \mathcal{K}$.
- $\mathsf{Enc}(k, m) \to \mathsf{ct}$ takes key k and message $m \in \mathcal{M}$ and outputs ciphertext $\mathsf{ct} \in \mathcal{C}$.
- $\mathsf{Dec}(k, \mathsf{ct}) \to m$ takes key k and ciphertext ct and outputs message m.

Encryption Scheme Correctness

An encryption scheme satisfies correctness if $\forall k \in \mathcal{K}$, $\forall m \in \mathcal{M}$, we have

$$\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] = 1,$$

where the probability is over the randomness used in encryption and decryption.

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext. What happens if the key is re-used?

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext. What happens if the key is re-used?
  - The ciphertext looks uniformly random to Eve.

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.
  - The ciphertext looks uniformly random to Eve.

<u>One-Time Uniform Ciphertext Security</u>

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathscr{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \qquad \equiv \qquad D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

Encryption: One-Time Uniform Ciphertext Security

- What the security definition should capture for encryption schemes like OTP
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.
  - The ciphertext looks uniformly random to Eve.

<u>One-Time Uniform Ciphertext Security</u>

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathscr{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

Identical distributions

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathscr{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{matrix} \right\} \quad \equiv \quad D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathcal{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := k \oplus m \end{array} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \{0,1\}^\lambda \right\}$$

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathscr{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \text{ct} \overset{\$}{\leftarrow} \mathscr{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

Fix arbitrary $m \in \mathscr{M}$ and $c \in \mathscr{C}$.

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \overset{\$}{\leftarrow} \{0,1\}^\lambda \\ \text{ct} := k \oplus m \end{array} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \text{ct} : \text{ct} \overset{\$}{\leftarrow} \{0,1\}^\lambda \right\}$$

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{matrix} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathcal{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := k \oplus m \end{matrix} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \{0,1\}^\lambda \right\}$$

Fix arbitrary $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = \mathsf{Enc}(k, m)] = \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = k \oplus m]$$

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{l} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathcal{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

$$D_0 = \left\{ \text{ct} : \begin{array}{l} k \xleftarrow{\$} \{0,1\}^\lambda \\ \text{ct} := k \oplus m \end{array} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \{0,1\}^\lambda \right\}$$

Fix arbitrary $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = \text{Enc}(k, m)] = \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = k \oplus m]$$

$$= \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [k = c \oplus m]$$

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathscr{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{l} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \qquad \equiv \qquad D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

$$D_0 = \left\{ \text{ct} : \begin{array}{l} k \xleftarrow{\$} \{0,1\}^\lambda \\ \text{ct} := k \oplus m \end{array} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \{0,1\}^\lambda \right\}$$

Fix arbitrary $m \in \mathscr{M}$ and $c \in \mathscr{C}$.

$$\Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = \text{Enc}(k, m)] = \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [c = k \oplus m]$$

$$= \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda} [k = c \oplus m] = \frac{1}{2^\lambda}$$

One-Time Pad: Security Proof

One-Time Uniform Ciphertext Security

An encryption scheme is one-time uniform ciphertext secure if $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{l} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathcal{C} \right\}$$

Claim: One-time pad is one-time uniform ciphertext secure.

Proof:

We need to show that $\forall m \in \{0,1\}^\lambda$

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{l} k \xleftarrow{\$} \{0,1\}^\lambda \\ \mathsf{ct} := k \oplus m \end{array} \right\}$$

$$\equiv$$

$$D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \{0,1\}^\lambda \right\}$$

Fix arbitrary $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\Pr_{k \xleftarrow{\$} \{0,1\}^\lambda}[c = \mathsf{Enc}(k, m)] = \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda}[c = k \oplus m]$$

$$= \Pr_{k \xleftarrow{\$} \{0,1\}^\lambda}[k = c \oplus m] = \frac{1}{2^\lambda}$$

$$\Pr_{\mathsf{ct} \xleftarrow{\$} \{0,1\}^\lambda}[\mathsf{ct} = c] = \frac{1}{2^\lambda}$$

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct?

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure?

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)
  - Consider $\mathsf{Enc}(k, m) =: m$

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)
  - Consider $\mathsf{Enc}(k, m) =: m$
    - Is it correct?

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider Enc$(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)
  - Consider Enc$(k, m) =: m$
    - Is it correct? ✓

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)
  - Consider $\mathsf{Enc}(k, m) =: m$
    - Is it correct? ✓
    - Is it one-time uniform ciphertext secure?

Requiring Both Correctness and Security

- Both correctness and security are required for a meaningful notion of encryption
  - Consider $\mathsf{Enc}(k, m) =: 0^\lambda$
    - Is it correct? ✗
    - Is it one-time uniform ciphertext secure? ✓ (Note that $\mathscr{C} = \{0^\lambda\}$)
  - Consider $\mathsf{Enc}(k, m) =: m$
    - Is it correct? ✓
    - Is it one-time uniform ciphertext secure? ✗

Insecure Encryption

Insecure Encryption

- An encryption scheme does NOT satisfy one-time uniform ciphertext security if $\exists m \in \mathscr{M}$ such that

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \qquad \not\equiv \qquad D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

Insecure Encryption

- An encryption scheme does NOT satisfy one-time uniform ciphertext security if $\exists m \in \mathscr{M}$ such that

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \qquad \not\equiv \qquad D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

- Is the following encryption scheme secure?
  - $\mathsf{KeyGen}() : k \xleftarrow{\$} \{0,1\}^\lambda$
  - $\mathsf{Enc}(k, m) : \mathsf{ct} := k \wedge m$

Insecure Encryption

- An encryption scheme does NOT satisfy one-time uniform ciphertext security if $\exists m \in \mathscr{M}$ such that

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \qquad \not\equiv \qquad D_1 = \left\{ \text{ct} : \text{ct} \overset{\$}{\leftarrow} \mathscr{C} \right\}$$

- Is the following encryption scheme secure?
  - $\text{KeyGen}() : k \overset{\$}{\leftarrow} \{0,1\}^\lambda$
  - $\text{Enc}(k, m) : \text{ct} := k \wedge m$

Ans: For $m = 0^\lambda$,

$$\Pr_{\text{ct} \leftarrow D_0}[\text{ct} = 0^\lambda] = 1,$$

$$\Pr_{\text{ct} \leftarrow D_1}[\text{ct} = 0^\lambda] = 1/2^\lambda.$$

Encryption: Perfect Security

- An alternative idea for defining security of encryption schemes.
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.
  - ~~The ciphertext looks uniformly random to Eve.~~ Encryptions of m_0 look like encryptions of m_1 to Eve.

Encryption: Perfect Security

- An alternative idea for defining security of encryption schemes.
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.
  - ~~The ciphertext looks uniformly random to Eve.~~ Encryptions of m_0 look like encryptions of m_1 to Eve.

<u>(One-Time) Perfect Security</u>

An encryption scheme is one-time perfectly secure if $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$$

Encryption: Perfect Security

- An alternative idea for defining security of encryption schemes.
  - The secret key should be kept hidden from Eve.
  - The key is only used to encrypt one plaintext.
  - ~~The ciphertext looks uniformly random to Eve.~~ Encryptions of m_0 look like encryptions of m_1 to Eve.

(One-Time) Perfect Security

An encryption scheme is one-time perfectly secure if $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$$

From Eve's view, the ciphertext carries no information about the plaintext.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathscr{M}$,
$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathcal{M}$, $$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

We want to show that $\forall m_0, m_1 \in \mathcal{M}$, $$D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}.$$

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathcal{M}$, $\quad D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}.$

We want to show that $\forall m_0, m_1 \in \mathcal{M}$, $\quad D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}.$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

We want to show that $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

Our goal is to show that $H_0 \equiv H_2$. We will do this in two steps using the "intermediate" hybrid H_1.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathscr{M}$,
$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

We want to show that $\forall m_0, m_1 \in \mathscr{M}$,
$$D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

We want to show that $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0' = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\}$$

$H_0 \equiv H_1$ because of one-time uniform ciphertext security.

$$H_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$$

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof: We are given that $\forall m \in \mathscr{M}$, $D_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}$.

We want to show that $\forall m_0, m_1 \in \mathscr{M}$, $D_0' = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \text{ct} : \text{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \text{ct} : \begin{array}{c} k \leftarrow \text{KeyGen}() \\ \text{ct} \leftarrow \text{Enc}(k, m_1) \end{array} \right\}$$

$H_0 \equiv H_1$ because of one-time uniform ciphertext security.

$H_1 \equiv H_2$ because of one-time uniform ciphertext security.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof: We are given that $\forall m \in \mathcal{M}$,
$$D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}.$$

We want to show that $\forall m_0, m_1 \in \mathcal{M}$,
$$D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathscr{C} \right\}$$

$$H_2 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

$H_0 \equiv H_1$ because of one-time uniform ciphertext security.

$H_1 \equiv H_2$ because of one-time uniform ciphertext security.

By transitivity, $H_0 \equiv H_2$.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Proof:

We are given that $\forall m \in \mathcal{M}$, $\quad D_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{array} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathcal{C} \right\}$.

We want to show that $\forall m_0, m_1 \in \mathcal{M}$, $\quad D_0' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\} \equiv D_1' = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$

We will consider the following sequence of distributions, called hybrids.

$$H_0 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{array} \right\}$$

$$H_1 = \left\{ \mathsf{ct} : \mathsf{ct} \xleftarrow{\$} \mathcal{C} \right\}$$

$$H_2 = \left\{ \mathsf{ct} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{array} \right\}$$

$H_0 \equiv H_1$ because of one-time uniform ciphertext security.

$H_1 \equiv H_2$ because of one-time uniform ciphertext security.

By transitivity, $H_0 \equiv H_2$.

The hybrid technique is very common in cryptographic proofs.
We will use it repeatedly throughout the course.

Comparing Both Security Notions

Claim: If an encryption scheme is one-time uniform ciphertext secure, then it is also perfectly secure.

Corollary: One-time pad is perfectly secure.