# COS 433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

# Announcements/Reminders

- HW1 due September 15
- PR1 due March 6

OH:

- Mark: Mondays 10-11am (starting next week)
- Udaya: Tuesdays 7-8pm (starting next week)
- Anunay: Wednesdays 1:30-2:30pm (starting tomorrow)

# Previously on COS 433…

# Formalizing Encryption (syntax and correctness)

**Syntax:**

- Key space **K**
- Message space **M**
- Ciphertext space **C**
- **Enc: K×M → C**
- **Dec: K×C → M**

**Correctness:**

- For all **k∈K, m∈M, Dec(k, Enc(k,m) ) = m**

# Example: Substitution Cipher

**K? Perms({a,…,z})**
**M? $\{a,\dots,z\}^*$**
**C? $\{a,\dots,z\}^*$**

**$\text{Enc}(k, m_1 m_2 \dots) = k(m_1)k(m_2)\dots$**

**$\text{Dec}(k, c_1 c_2 \dots) = k^{-1}(c_1)k^{-1}(c_2)\dots$**

Correctness: **$m_i' = k^{-1}(c_i) = k^{-1}(k(m_i)) = m_i$**

# Encryption Security?

Questions to think about:

<u>What kind of messages?</u><br>
<u>What does the adversary already know?</u><br>
<u>What information are we trying to protect?</u>

Examples:

- Messages are always either “attack at dawn” or “attack at dusk”, trying to hide which is the case
- Messages are status updates (“<person> reports <event> at <location>”). Which data is sensitive?

# Encryption Security?

Questions to think about:

<u>What kind of messages?</u>
<u>What does the adversary already know?</u>
<u>What information are we trying to protect?</u>

Goal:

Rather than design a separate system for each use case, design a system that works in all possible settings

# Today: Encryption Security, continued

# Semantic Security

Idea:

- Plaintext comes from an arbitrary distribution
- Adversary initially has some information about the plaintext
- Seeing the ciphertext should not reveal any more information
- Model unknown key by assuming it is chosen uniformly at random

# (Perfect) Semantic Security

**Definition:** A scheme **(Enc,Dec)** is **(perfectly) semantically secure** if, for all:

- Distributions **D** on **M** ← Plaintext distribution
- Functions $\mathbf{I:M\rightarrow\{0,1\}^*}$ ← Info adv gets
- Functions $\mathbf{f:M\rightarrow\{0,1\}^*}$ ← Info adv tries to learn
- Functions $\mathbf{A:C\times\{0,1\}^*\rightarrow\{0,1\}^*}$ ← Adversary

"Simulator"

There exists a function $\mathbf{S:\{0,1\}^*\rightarrow\{0,1\}^*}$ such that

$$\Pr[\ A(\ \mathrm{Enc}(k,m)\ ,\ I(m)\ )\ =\ f(m)\ ]$$
$$=\ \Pr[\ S(\ I(m)\ )\ =\ f(m)\ ]$$

where probabilities are taken over **k←K, m←D**

# Semantic Security

Captures what we want out of an encryption scheme

But, complicated, with many moving parts

Want: something simpler…

# Notation

Two random variables **X,Y** over a finite set **S** have identical distributions if, for all $\mathbf{s} \in \mathbf{S}$,

$$\Pr[\, X = s] \;=\; \Pr[\, Y = s]$$

In this case, we write

$$X \overset{d}{=} Y$$

# Perfect Secrecy [Shannon'49]

**Definition:** A scheme **(Enc,Dec)** has **perfect secrecy** if, for any two messages $\mathbf{m_0}, \mathbf{m_1} \in \mathbf{M}$

$$\mathbf{Enc(K,\ m_0)} \overset{d}{=} \underbrace{\mathbf{Enc(K,\ m_1)}}$$

Random variable corresponding to uniform distribution over **K**

Random variable corresponding to encrypting $\mathbf{m_1}$ using a uniformly random key

# Semantic Security = Perfect Secrecy

**Theorem:** A scheme **(Enc,Dec)** is semantically secure if and only if it has perfect secrecy

Intuition

Semantic Security ⟹ Perfect Secrecy

- Side information: message $\in \{m_0, m_1\}$
- Adversary trying to learn which one

Perfect Secrecy ⟹ Semantic Security

- $S(I(m)) = A(\ Enc(k,0),\ I(m)\ )$

# Perfect vs. Semantic Security

Semantic security is the “right” notion to intuitively capture the desired security goals

Perfect is much simpler and easier to reason about

Fortunately, we know both are identical
$\Rightarrow$ perfect security is almost always what is used

# Any perfectly/semantically secure schemes?

# Perfect Security of One-Time Pad

Fix any message $\mathbf{m}\in\{0,1\}^n$, ciphertext $\mathbf{c}\in\{0,1\}^n$

$$\begin{aligned}\Pr_k[\mathrm{Enc}(k,m)=c] &= \Pr_k[k\oplus m=c]\\ &= \Pr_k[k=m\oplus c]\\ &= 2^{-n}\end{aligned}$$

Therefore, for any $\mathbf{m}$, $\mathbf{Enc(K,\ m)} =$ uniform dist.

In particular, for any $\mathbf{m_0,m_1}$,

$$\mathrm{Enc}(K,\ m_0) \overset{d}{=} \mathrm{Enc}(K,\ m_1)$$

# Insecurity of Substitution/Transposition

$m_0$ = aa
$m_1$ = ab

$\Pr[\mathrm{Enc}(K,m_0)$ has 2 identical characters$] = 1$
$\Pr[\mathrm{Enc}(K,m_1)$ has 2 identical characters$] = 0$

# Proper Use Case for Perfect Security

- Message can come from any distribution ✓
- Adversary can know anything about message ✓
- Encryption hides anything ✓

- But, definition only says something about an adversary that sees a single message ✘

  $\Longrightarrow$ If two messages, no security guarantee

- Assumes no side-channels ✘
- Assumes key is uniformly random ✘

# Variable Length Messages

# Variable-Length Messages

OTP has message-length $\{0,1\}^n$ where $n$ is the key length

In practice, fixing the message size is often unreasonable

So instead, will allow for smaller messages to be encrypted

# Variable-Length OTP

Key space $\mathbf{K} = \{0,1\}^n$
Message space $\mathbf{M} = \{0,1\}^{\leq n}$
Ciphertext space $\mathbf{C} = \{0,1\}^{\leq n}$

$\mathbf{Enc(k, m) = k_{[1, |m|]} \oplus m}$
$\mathbf{Dec(k, c) = k_{[1, |c|]} \oplus c}$

Correctness:

$$\begin{aligned} \mathbf{Dec(k, Enc(k, m))} &= k\oplus(k\oplus m) \\ &= (k\oplus k)\oplus m \\ &= 0\oplus m \\ &= m \end{aligned}$$

Does the variable length OTP have perfect secrecy according to our definition?

# Ciphertext Size

**Theorem:** For a scheme with perfect secrecy, the expected ciphertext size for any message, $\mathbb{E}[\ |\mathbf{Enc}(K,m)|\ ]$, is at least $(\log_2 |M|) - 3$

Proof Idea:

- $|$ **Enc(**random message**)** $| \gtrsim \log_2 |M|$
- But, by security, **Enc(**any message**)** must be distributed identically to **Enc(**random message**)**

# Variable-Length Messages

For perfect secrecy of variable length messages, must have expected ciphertext length for short messages almost as long as longest messages

In practice, very undesirable

- What if I want to either send a **100mb** attachment, or just a message “How are you?”

Therefore, we usually allow message length to be revealed

# (Perfect) Semantic Security for Variable Length Messages

**Definition:** A scheme **(Enc,Dec)** is **(perfectly) semantically secure** if, for all:

- Distributions **D** on **M**
- (Probabilistic) Functions $\mathbf{I:M\rightarrow\{0,1\}^*}$
- (Probabilistic) Functions $\mathbf{f:M\rightarrow\{0,1\}^*}$
- (Probabilistic) Functions $\mathbf{A:C\times\{0,1\}^*\rightarrow\{0,1\}^*}$

There exists (probabilistic) func $\mathbf{S:\{0,1\}^*\rightarrow\{0,1\}^*}$ s.t.

$$\Pr[\,A(\,\mathrm{Enc}(k,m)\,,\,I(m)\,) = f(m)\,]$$
$$= \Pr[\,S(\,I(m),\,\boxed{|m|}\,) = f(m)\,]$$

where probabilities are taken over **k←K, m←D**

# Perfect Secrecy For Variable Length Messages

**Definition:** A scheme **(Enc,Dec)** has **perfect secrecy** if, for any two messages $\mathbf{m_0}$, $\mathbf{m_1}$ where $|m_0| = |m_1|$,

$$\mathrm{Enc}(K, m_0) \overset{d}{=} \mathrm{Enc}(K, m_1)$$

Easy to adapt earlier thm to show:

**Theorem:** A scheme **(Enc,Dec)** is semantically secure if and only if it has perfect secrecy

# Variable-Length OTP

Key space $\mathcal{K} = \{0,1\}^n$
Message space $\mathcal{M} = \{0,1\}^{\leq n}$
Ciphertext space $\mathcal{C} = \{0,1\}^{\leq n}$

$\mathsf{Enc}(k, m) = k_{[\,1,\ |m|\,]} \oplus m$
$\mathsf{Dec}(k, c) = k_{[\,1,\ |m|\,]} \oplus c$

**Theorem:** Variable-Length OTP has perfect secrecy

# Encrypting Multiple Messages

# Re-using the OTP

What if we have a **100mb** long key **k**, but encrypt only **1mb**?

Can’t use first **1mb** of **k** again, but remaining **99mb** is still usable

However, basic OTP definition does not allow us to re-use the key ever

# Syntax for Stateful Encryption

**Syntax:**

- Key space **K**,Message space **M**,Ciphertext space ***C***
- State Space **S**
- **Init: {} → S**
- **Enc: K×M×S → C×S**
- **Dec: K×C×S → M×S**

**$\mathrm{State}_0$ ← Init()**

**($c_0$, $\mathrm{state}_1$) ← Enc(k,$m_0$,$\mathrm{state}_0$)**

**($c_1$, $\mathrm{state}_2$) ← Enc(k,$m_1$,$\mathrm{state}_1$)**

**…**

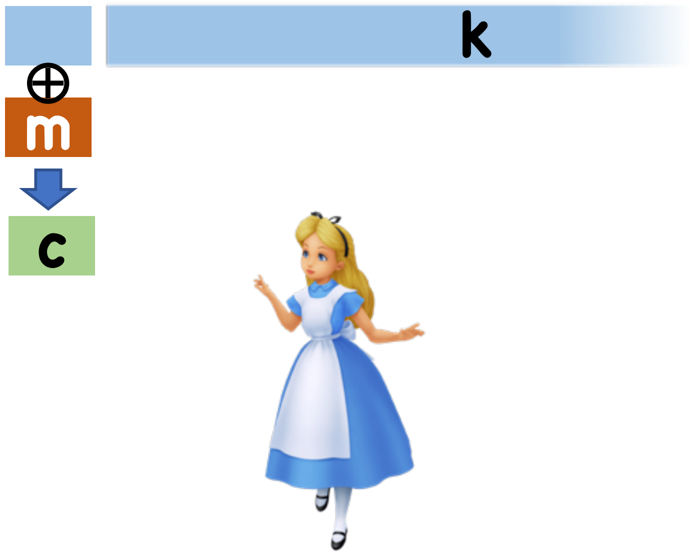
# Reusing the OTP

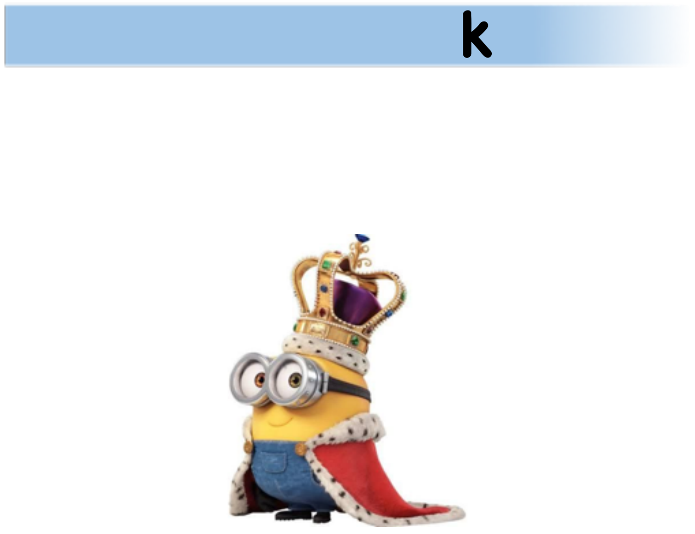
k

k

m

# Reusing the OTP





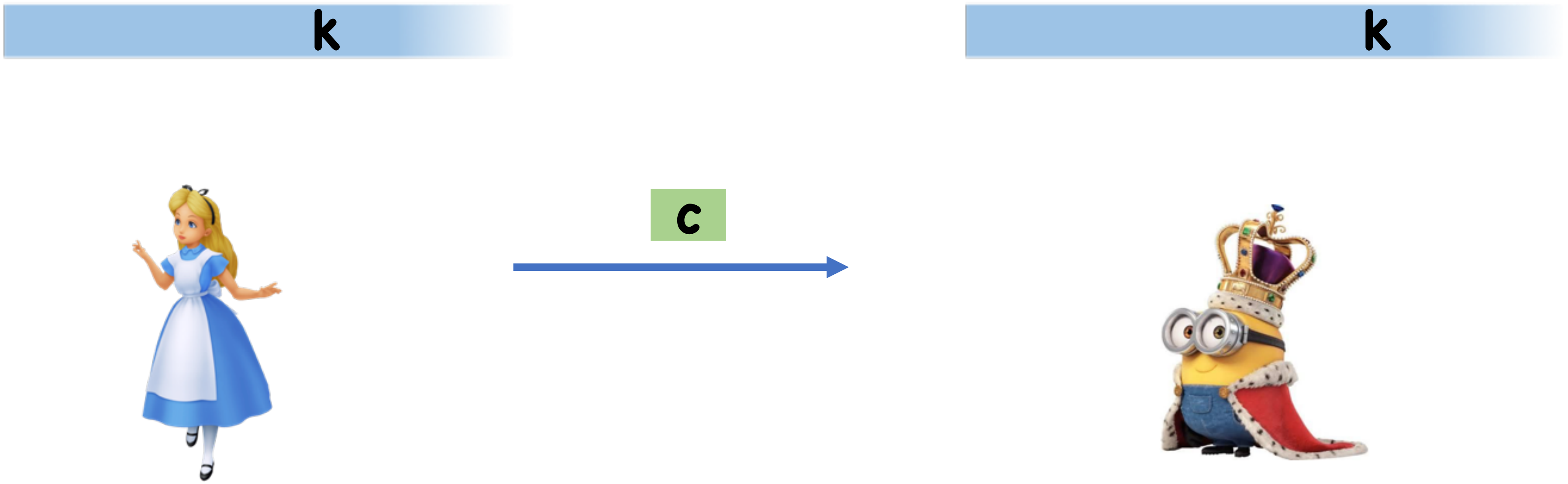
# Reusing the OTP

k

k

c

# Reusing the OTP



k

c

k

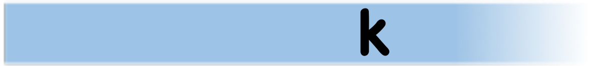
# Reusing the OTP

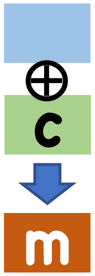
k

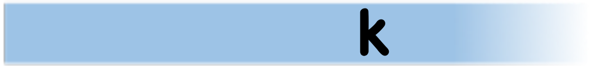
k

c

# Reusing the OTP







# Reusing the OTP

k

m

k

# Reusing the OTP

k

k

m′

# Reusing the OTP

k

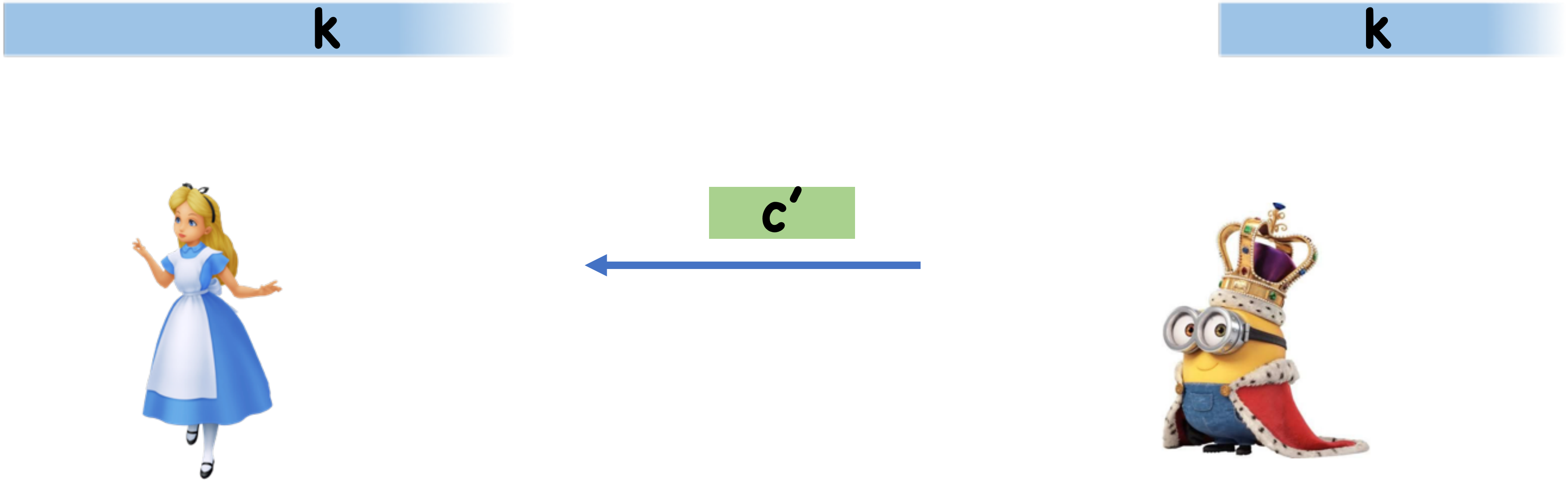
$\oplus$

m'

c'

k

# Reusing the OTP



k

k

c′

# Reusing the OTP

k

c′

k

# Reusing the OTP

k

⊕

c

m'

k

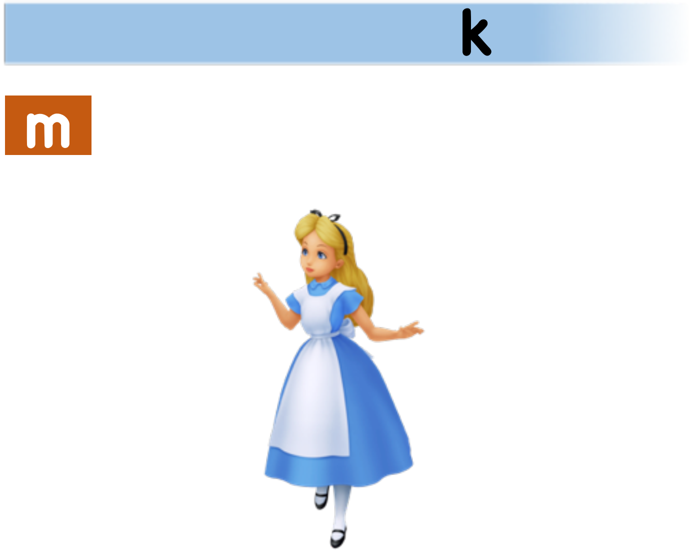
# Problem

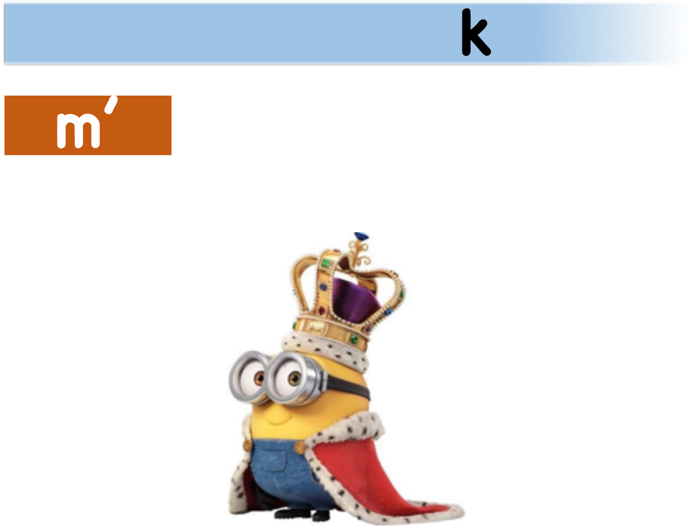
In real world, messages aren’t always synchronous

What happens if Alice and Bob try to send message at the same time?

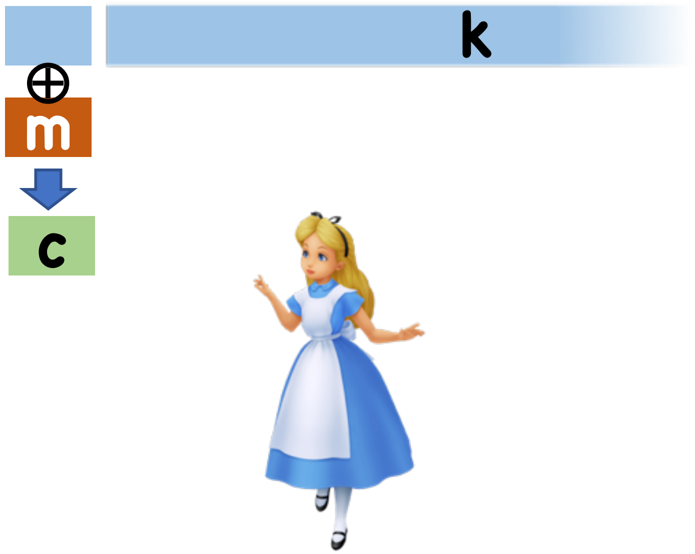
**They will both use the same part of the key!**

# Problem





# Problem

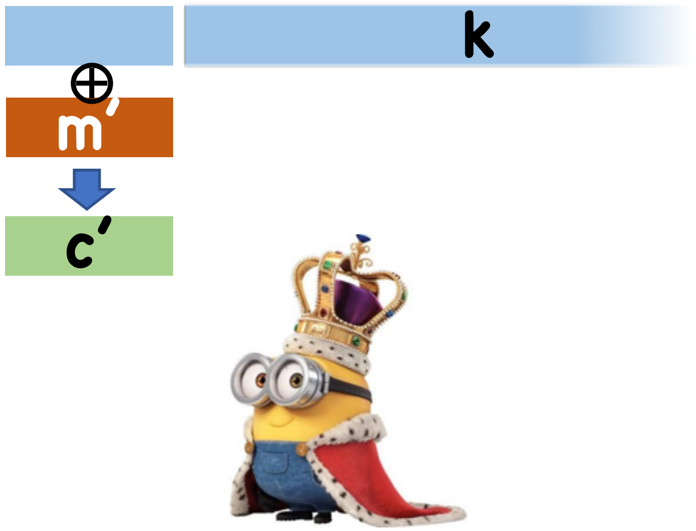


k

$\oplus$

m

c



k

$\oplus$

m’

c’

# Problem

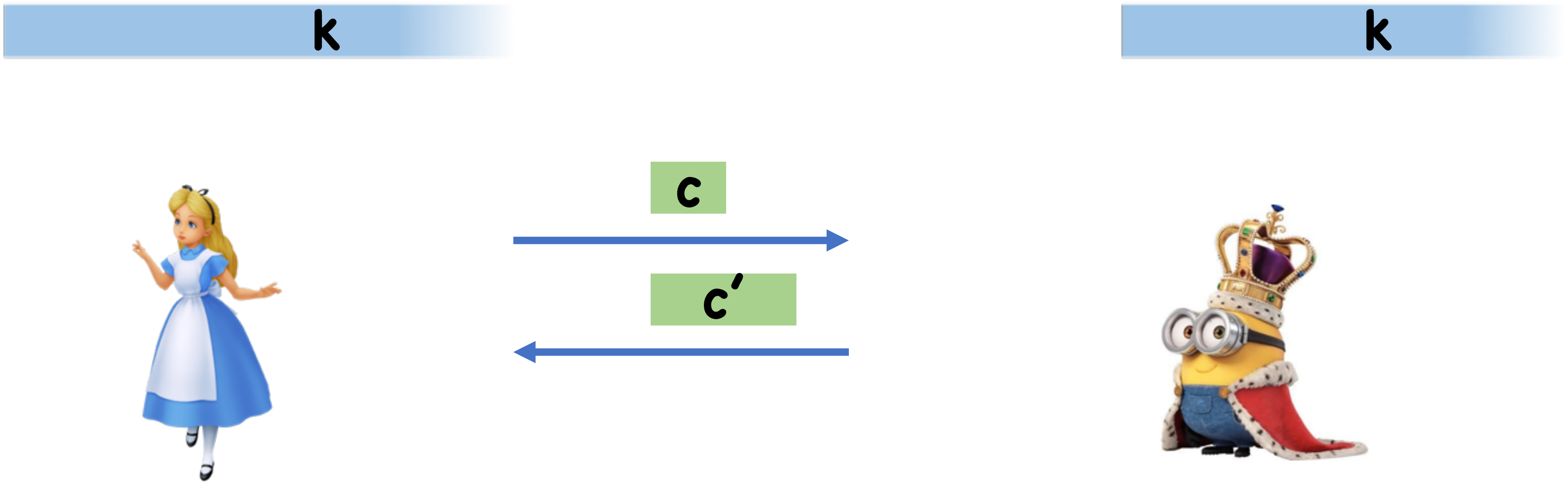
k

c

k

c′

# Problem



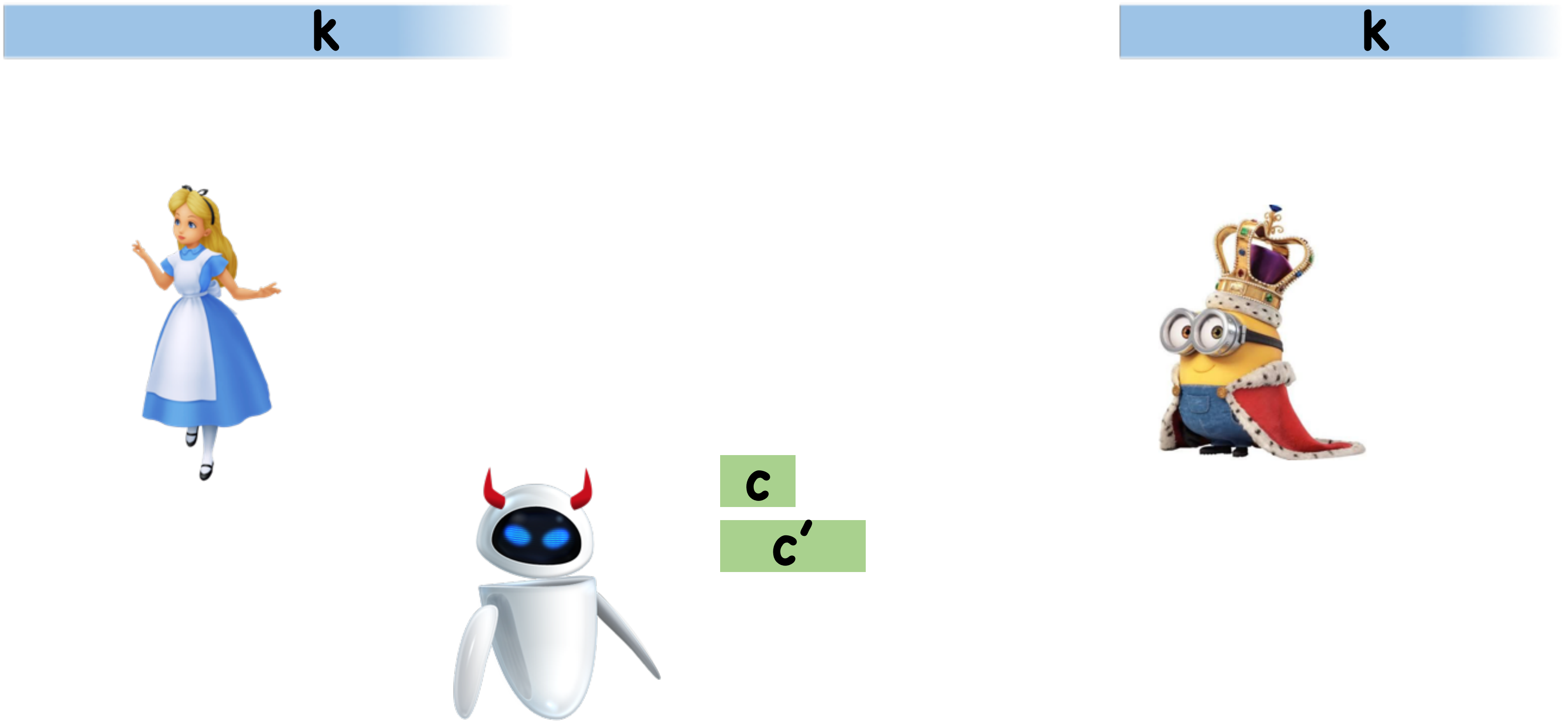
k

k

c

c′

# Problem



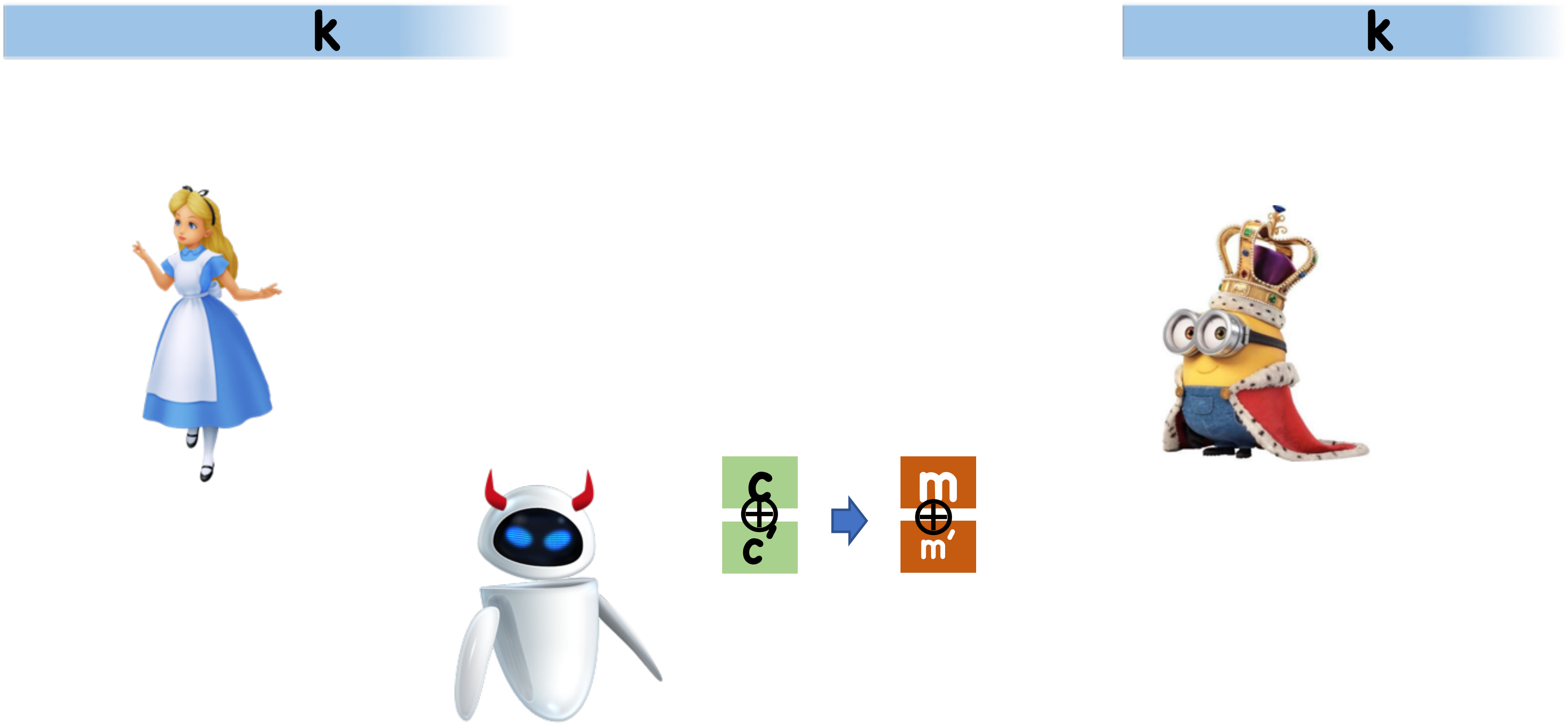
k

k

c

c′

# Problem



k

k

$c \oplus c'$ → $m \oplus m'$

# Solution

Alice and Bob have two keys

- One for communication from Alice to Bob
- One for communication from Bob to Alice

Can obtain two logical keys from one by splitting key in half

- Ex: odd bits form $\mathbf{k}_{A\rightarrow B}$, even bits form $\mathbf{k}_{B\rightarrow A}$

# Reusing the OTP

$k_{A\to B}$

$k_{B\to A}$

$k_{A\to B}$

$k_{B\to A}$

# Still A Problem

In real world, messages aren’t always synchronous

Also, sometimes messages arrive out of order or get dropped

- Need to be very careful to make sure decryption succeeds

And, what if more users?

These difficulties exist in any stateful encryption

- For this course, we will generally consider only **stateless** encryption schemes

# Perfect Security for Multiple Messages

**Definition:** A stateless scheme **(Enc,Dec)** has **perfect secrecy for n messages** if, for any two sequences of messages $(m_0^{(i)})_{i\in[n]}$ , $(m_1^{(i)})_{i\in[n]} \in M^n$

$$\left(\mathrm{Enc}(K,\ m_0^{(i)})\right)_{i\in[n]} \overset{d}{=} \left(\mathrm{Enc}(K,\ m_1^{(i)})\right)_{i\in[n]}$$

Notation: $(\ f(i)\ )_{i\in[n]} = (\ f(1),\ f(2),\ \ldots,\ f(n)\ )$

# Stateless Encryption with Multiple Messages

Ex:

**M = C**
**K = Perms(M)** (never mind that key is enormous)
**Enc( K, m) = K(m)**
**Dec( K, c) = $K^{-1}$(c)**

Q: Is this perfectly secure for two messages?

**Theorem:** No stateless deterministic encryption scheme can have perfect security for multiple messages

Proof Idea: can always tell if two messages were the same or different

# Randomized Encryption

**Syntax:**

- Key space **K**
- Message space **M**
- Ciphertext space **C**
- **Enc: K×M → C**, potentially probabilistic
- **Dec: K×C → M** (usually deterministic)

**Correctness:**

- ~~For all **k∈K, m∈M, Dec(k, Enc(k,m) ) = m**~~

# Randomized Encryption

**Syntax:**

- Key space **K**
- Message space **M**
- Ciphertext space **C**
- **Enc: K×M → C**, potentially probabilistic
- **Dec: K×C → M** (usually deterministic)

**Correctness:**

- For all $\mathbf{k}\in\mathbf{K}$, $\mathbf{m}\in\mathbf{M}$,

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

# Stateless Encryption with Multiple Messages

Ex:

$r \leftarrow R$

$C = M \times R$

$K = \mathrm{Perms}(C)$

$\mathrm{Enc}(K, m) = K(m,r)$

$\mathrm{Dec}(K, c) = (m',r') \leftarrow K^{-1}(c)$, output $m'$

Q: Is this perfectly secure for two messages?

even randomized

**Theorem:** No stateless ~~deterministic~~ encryption scheme can have perfect security for multiple messages

# Proof of Easy Case

Let **(Enc,Dec)** be stateless, deterministic

Let $\mathbf{m_0^{(0)} = m_0^{(1)}}$
Let $\mathbf{m_1^{(0)} \neq m_1^{(1)}}$

Consider distributions of encryptions:

- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_0^{(0)}\ ),\ Enc(K,\ m_0^{(1)}\ )\ )}$
  $\Rightarrow \mathbf{c^{(0)} = c^{(1)}}$ (by determinism)

- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_1^{(0)}\ ),\ Enc(K,\ m_1^{(1)}\ )\ )}$
  $\Rightarrow \mathbf{c^{(0)} \neq c^{(1)}}$ (by correctness)

# Generalize to Randomized Encryption

Let **(Enc,Dec)** be stateless, ~~deterministic~~

Let $\mathbf{m_0^{(0)} = m_0^{(1)}}$
Let $\mathbf{m_1^{(0)} \neq m_1^{(1)}}$

Consider distributions of encryptions:

- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_0^{(0)}\ ),\ Enc(K,\ m_0^{(1)}\ )\ )}$
  $\Rightarrow$ **????**
- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_1^{(0)}\ ),\ Enc(K,\ m_1^{(1)}\ )\ )}$
  $\Rightarrow \mathbf{c^{(0)} \neq c^{(1)}}$ (by correctness)

# Generalize to Randomized Encryption

$( c^{(0)} , c^{(1)} ) = ( \mathrm{Enc}(K, m), \mathrm{Enc}(K, m) )$

$\Pr[c^{(0)} = c^{(1)}]$ ?

- Fix **k**
- Conditioned on **k**, $c^{(0)}$ and $c^{(1)}$ are two independent samples from same distribution $\mathrm{Enc}(k, m)$

**Lemma:** Given any distribution **D** over a finite set **X**, $\Pr[Y=Y' : Y \leftarrow D, Y' \leftarrow D] \geq 1/|X|$

- Therefore, $\Pr[c^{(0)} = c^{(1)}]$ is non-zero

# Generalize to Randomized Encryption

Let **(Enc,Dec)** be stateless, deterministic

Let $\mathbf{m_0^{(0)} = m_0^{(1)}}$
Let $\mathbf{m_1^{(0)} \neq m_1^{(1)}}$

Consider distributions of encryptions:

- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_0^{(0)}\ ),\ Enc(K,\ m_0^{(1)}\ )\ )}$

  $\Rightarrow \mathbf{Pr[c^{(0)} = c^{(1)}] > 0}$

- $\mathbf{(\ c^{(0)}\ ,\ c^{(1)}\ ) = (\ Enc(K,\ m_1^{(0)}\ ),\ Enc(K,\ m_1^{(1)}\ )\ )}$

  $\Rightarrow \mathbf{Pr[c^{(0)} = c^{(1)}] = 0}$

# What do we do now?

Tolerate tiny probability of distinguishing

- If $\Pr[c^{(0)} = c^{(1)}] = 2^{-128}$, in reality never going to happen

How small is ok?

- Discuss next time