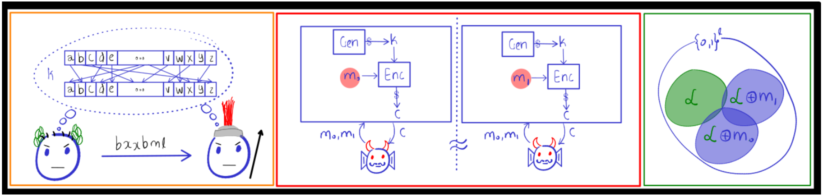



# CS783: Theoretical Foundations of Cryptography

Lecture 2 (02/Aug/24)

Instructor: Chethan Kamath

# Recall from Last Lecture...

MODULE 1 (Shared keys)

MODULE 2 (Public keys)

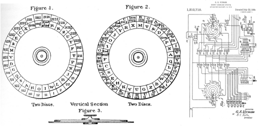
MODULE 3 (Secure comp.)

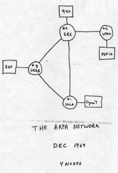
MODULE 4 (Adv. tasks)

For a large part of history

Advent of internet

Ubiquity of computing



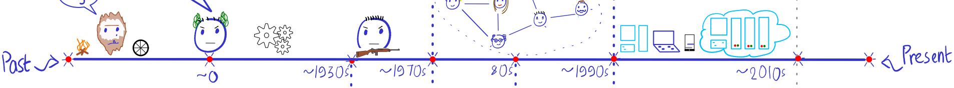

*

**

```
2 #include <iostream>
3
4 int main(int argc, char *argv[])
5 {
6     std::cout << "Not prime.\n";
7 }
```

ooga

Ego quoque sum conscius secreti




Birth of information theory & CS

Foundations of modern cryptography

Birth of "provable security"

Credit for images: Wikipedia (*User:DARPA)(**User:Saliko)

# Plan for this Lecture...

MODULE 1
(Shared keys)

For a large part of history

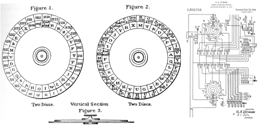
MODULE 2
(Public keys)

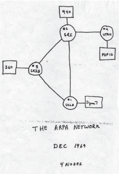
Advent of internet

MODULE 3
(Secure comp.)

Ubiquity of computing

MODULE 4
(Adv. tasks)

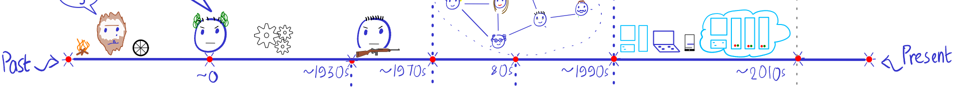



*

**

```
2 #include <iostream>
3
4 int main(int argc, char *argv[])
5 {
6     std::cout << "Not prime.\n";
7 }
```




Birth of information theory & CS
Foundations of modern cryptography

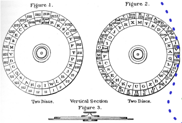
Birth of "provable security"

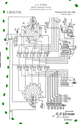
Credit for images: Wikipedia (*User:DARPA)(**User:Saliko)

MODULE 1

(Shared keys)

For a large part of history



Some historical ciphers

Perfect secrecy and one-time pad

Ego quoque sum conscius secreti

ooga

Past

~0

~1930s

~1970s

Birth of information theory & CS

Foundations of modern cryptography

Credit for images: Wikipedia (*User:DARPA)(**User:Saliko)

# Recall from Last 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 $\Pi$
4. Formally prove that $\Pi$ in secure in model $M$

# Plan for this Lecture...

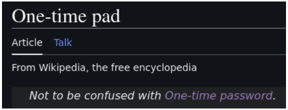
General *template*: secret communication with shared keys

1. Identify the task
   Perfect secrecy
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 $\Pi$ One-time pad
4. Formally prove that $\Pi$ in secure in model $M$

# Plan for this Lecture...

1 Syntax of Shared/Symmetric-Key Encryption (SKE)

2 Classical ciphers




+First proof

3 Perfect Secrecy and One-Time Pad (OTP)

# Plan for this Lecture

1. Syntax of Shared/Symmetric-Key Encryption (SKE)

2. Classical ciphers

3. Perfect Secrecy and One-Time Pad (OTP)

# Some Notation and Conventions

- Sets:
  - Denoted using calligraphic font: e.g., $\mathcal{M}$, $\mathcal{C}$
  - Sampling *uniformly at random* from a set is denoted using '$\leftarrow$': e.g., $k \leftarrow \{0, 1\}^{\ell}$

# Some Notation and Conventions

- Sets:
  - Denoted using calligraphic font: e.g., $\mathcal{M}$, $\mathcal{C}$
  - Sampling *uniformly at random* from a set is denoted using '$\leftarrow$': e.g., $k \leftarrow \{0,1\}^{\ell}$
- Algorithms
  - Algorithms will be denoted using straight font: e.g., $\mathsf{A}$, $\mathsf{Eve}$ ...
  - For a randomised algorithm $\mathsf{A}$, $y \leftarrow \mathsf{A}(x)$ denotes running $\mathsf{A}$ on input $x$ to get a (random) output $y$

# Some Notation and Conventions

- Sets:
  - Denoted using calligraphic font: e.g., $\mathcal{M}$, $\mathcal{C}$
  - Sampling *uniformly at random* from a set is denoted using '$\leftarrow$': e.g., $k \leftarrow \{0,1\}^{\ell}$
- Algorithms
  - Algorithms will be denoted using straight font: e.g., $\mathsf{A}$, Eve ...
  - For a randomised algorithm $\mathsf{A}$, $y \leftarrow \mathsf{A}(x)$ denotes running $\mathsf{A}$ on input $x$ to get a (random) output $y$
- Probability notation:
  - For a distribution $M$ over a set $\mathcal{M}$ and element $m \in \mathcal{M}$, $m = M$ denotes the *event*: 'a random sample from $M$ equals $m$"
  - Following denotes probability that $\mathsf{A}(x) = 1$ when $x \leftarrow \{0,1\}^n$:

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathsf{A}(x) = 1]$$

# Syntax of Shared/Symmetric-Key Encryption

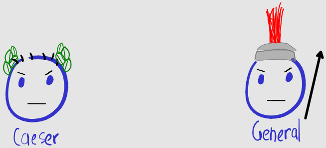
**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

# Syntax of Shared/Symmetric-Key Encryption

**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

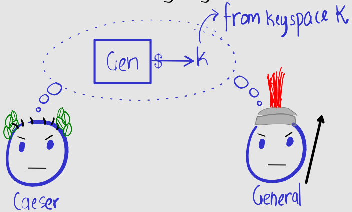
An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



# Syntax of Shared/Symmetric-Key Encryption

**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

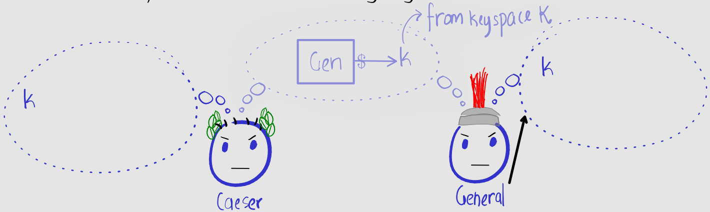
An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



# Syntax of Shared/Symmetric-Key Encryption

**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

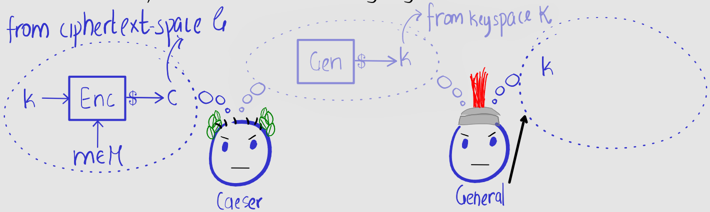
An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



# Syntax of Shared/Symmetric-Key Encryption

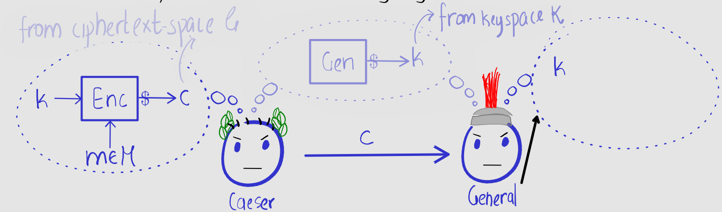
**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



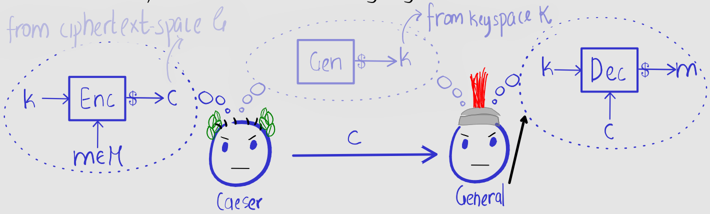
**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms (Gen, Enc, Dec) with the following syntax:



**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

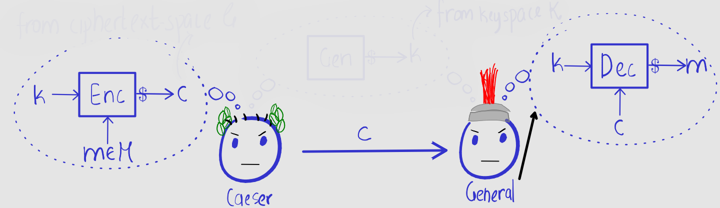
An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms (Gen, Enc, Dec) with the following syntax:



# Syntax of Shared/Symmetric-Key Encryption

**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



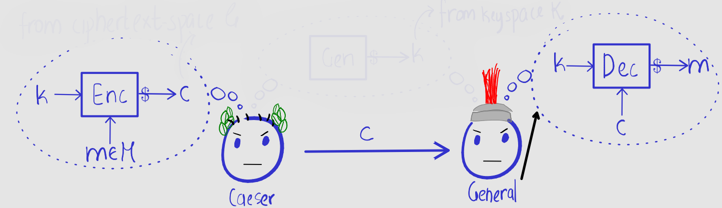

- Correctness of decryption: for all message $m \in \mathcal{M}$,

$$\Pr_{k \leftarrow \mathsf{Gen}, c \leftarrow \mathsf{Enc}(k,m)}[\mathsf{Dec}(k, c) = m] = 1$$

# Syntax of Shared/Symmetric-Key Encryption

**Definition 1 (Shared/Symmetric-Key Encryption (SKE))**

An SKE $\Pi$ for message space $\mathcal{M}$ is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



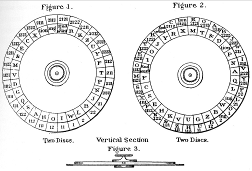

- Correctness of decryption: for all message $m \in \mathcal{M}$,

$$\Pr_{k \leftarrow \mathsf{Gen}, c \leftarrow \mathsf{Enc}(k,m)}[\mathsf{Dec}(k, c) = m] = 1$$

Why can we assume that $\mathsf{Dec}$ is *deterministic* w.l.o.g.?

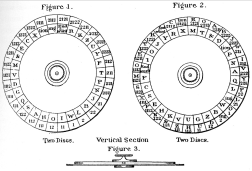
# Plan for this Lecture

# Shift Cipher (Caeser Cipher)

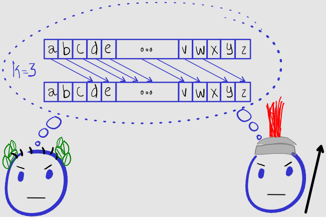



## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)

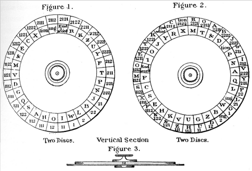
# Shift Cipher (Caeser Cipher)



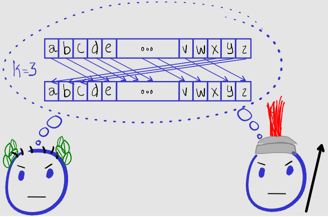

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



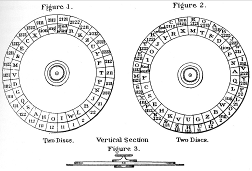

# Shift Cipher (Caeser Cipher)



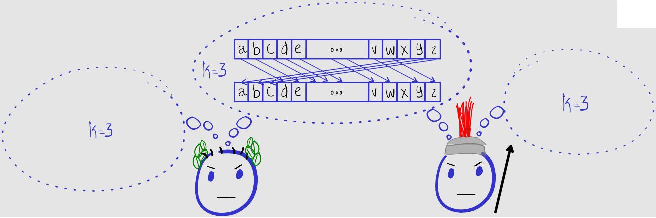

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



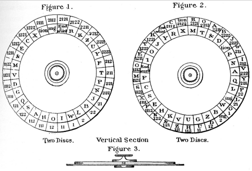

# Shift Cipher (Caeser Cipher)



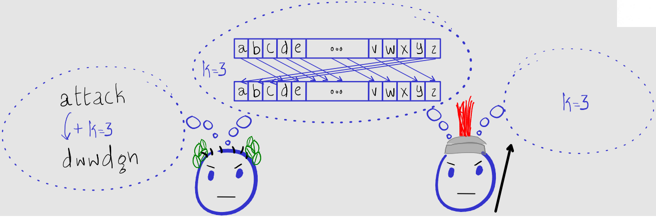

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)



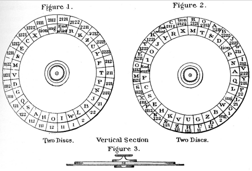

# Shift Cipher (Caeser Cipher)



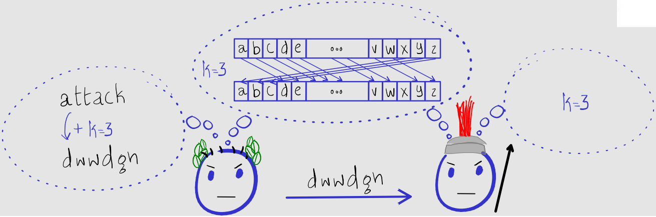

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



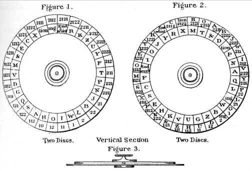

# Shift Cipher (Caeser Cipher)



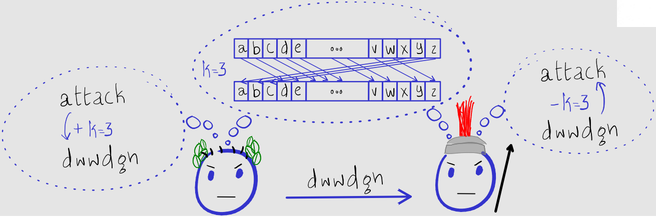

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)



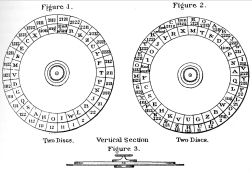

# Shift Cipher (Caeser Cipher)



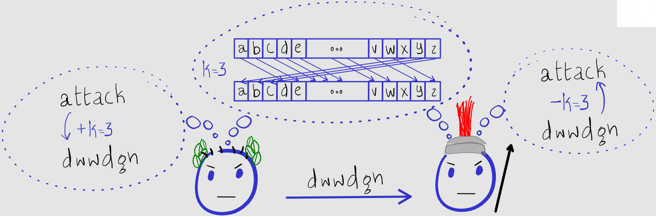

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)




# Shift Cipher (Caeser Cipher)



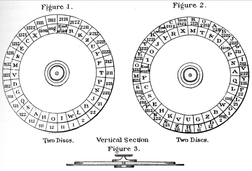

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



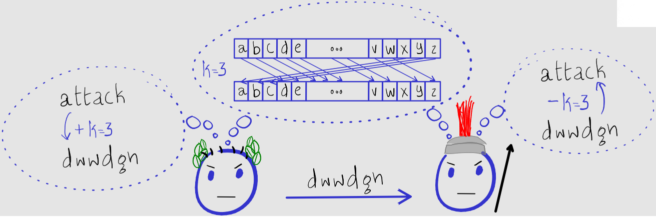

## Pseudocode 1 (Message space $\{0, \cdots, 25\}^{\ell} \leftrightarrow \{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)

- Key generation, **Gen**: output $k \leftarrow \{0, \cdots, 25\}$

# Shift Cipher (Caeser Cipher)



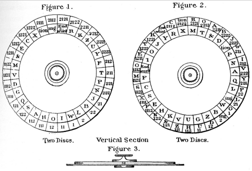

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)



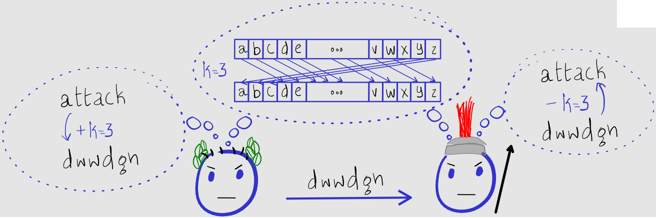

## Pseudocode 1 (Message space $\{0, \cdots, 25\}^\ell \leftrightarrow \{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)

- Key generation, **Gen**: output $k \leftarrow \{0, \cdots, 25\}$
- Encryption, $\mathsf{Enc}(k, m = m_1 \cdots m_\ell)$:
  - Output $c := c_1 \cdots c_\ell$, where $c_i := m_i + k \bmod 26$

# Shift Cipher (Caeser Cipher)



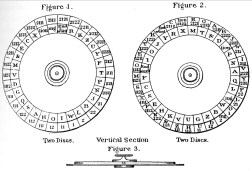

## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)



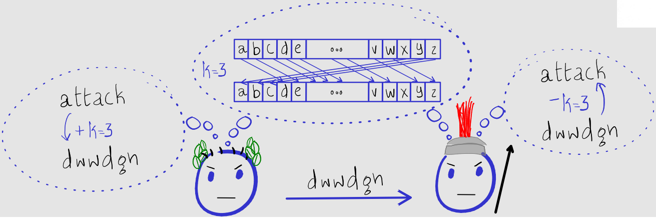

## Pseudocode 1 (Message space $\{0, \cdots, 25\}^\ell \leftrightarrow \{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)

- Key generation, $\mathsf{Gen}$: output $k \leftarrow \{0, \cdots, 25\}$
- Encryption, $\mathsf{Enc}(k, m = m_1 \cdots m_\ell)$:
  - Output $c := c_1 \cdots c_\ell$, where $c_i := m_i + k \bmod 26$
- Decryption, $\mathsf{Dec}(k, c = c_1 \cdots c_\ell)$:
  - Output $m := m_1 \cdots m_\ell$, where $m_i := c_i - k \bmod 26$

# Shift Cipher (Caeser Cipher)




## Construction 1 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)



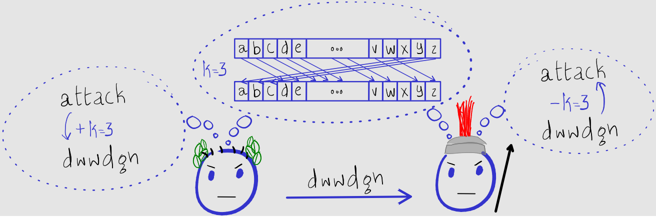

## Pseudocode 1 (Message space $\{0, \cdots, 25\}^\ell \leftrightarrow \{\mathtt{a}, \cdots, \mathtt{z}\}^\ell$)

- Key generation, $\mathsf{Gen}$: output $k \leftarrow \{0, \cdots, 25\}$
- Encryption, $\mathsf{Enc}(k, m = m_1 \cdots m_\ell)$:
  - Output $c := c_1 \cdots c_\ell$, where $c_i := m_i + k \bmod 26$
- Decryption, $\mathsf{Dec}(k, c = c_1 \cdots c_\ell)$:
  - Output $m := m_1 \cdots m_\ell$, where $m_i := c_i - k \bmod 26$

? Why does correctness of decryption hold?

# Shift Cipher (Caeser Cipher)...

## Construction 2 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)

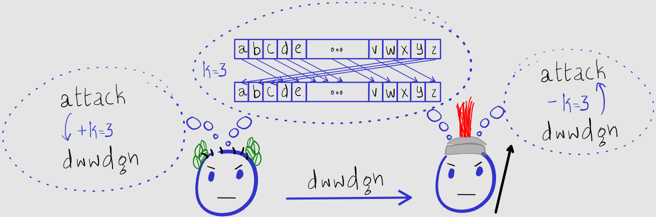



## Exercise 1

1. *What is the key-space? What is the ciphertext-space?*
2. *What is the probability that* $k = 10$*? What is* $\mathsf{Enc}(10, \mathtt{attack})$*?*

# Shift Cipher (Caeser Cipher)...

## Construction 2 (for message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)

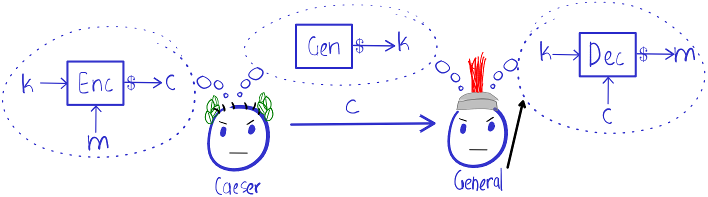



## Exercise 1

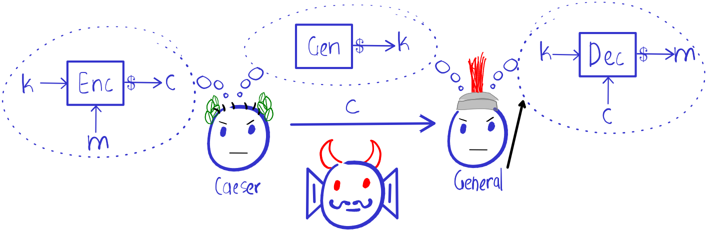
1. *What is the key-space? What is the ciphertext-space?*
2. *What is the probability that $k = 10$? What is* $\mathsf{Enc}(10, \mathtt{attack})$*?*

   *Assume that Caeser only sends either* `attack` *or* `defend`*.*
3. *What is the probability that the ciphertext is* `kddkmu`*, (resp.* `kddkmw`*)?*
4. *If ciphertext is* `kddkmu`*, is it possible that message is* `defend`*?*

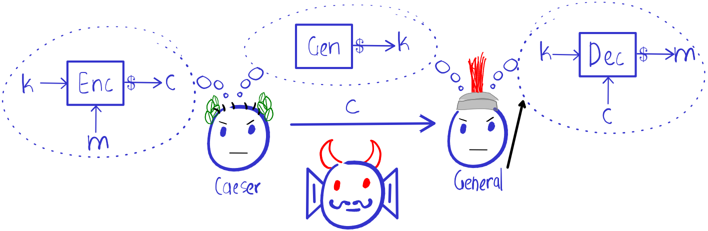
# First Let's Try to Model our Eavesdropper Eve



# First Let's Try to Model our Eavesdropper Eve

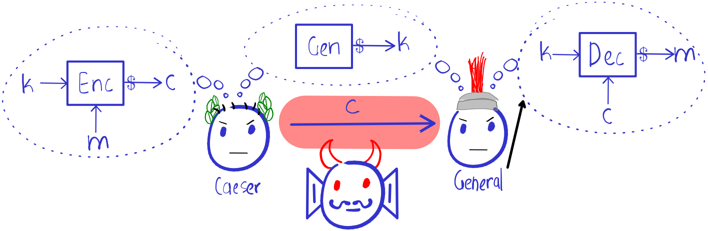


# First Let's Try to Model our Eavesdropper Eve



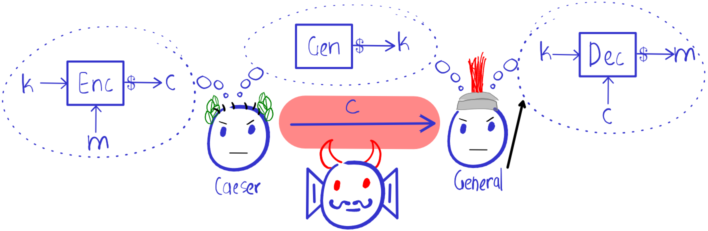

- Can be modelled as an algorithm

# First Let's Try to Model our Eavesdropper Eve




- Can be modelled as an algorithm

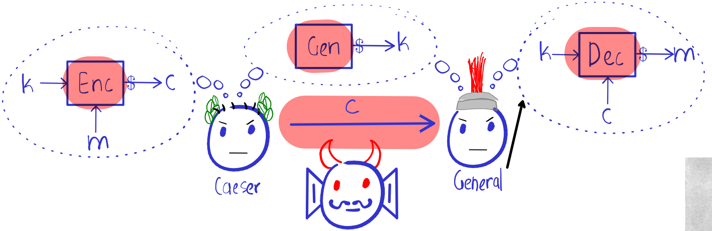
? What does Eve have access to?




- Can be modelled as an algorithm

? What does Eve have access to?

  - Description of the algorithms?



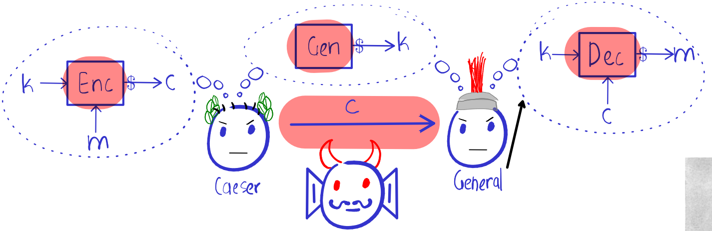

- Can be modelled as an algorithm

? What does Eve have access to?

- Description of the algorithms? Yes, Kerckhoffs' principle:

  *'One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.'*

# First Let's Try to Model our Eavesdropper Eve




- Can be modelled as an algorithm

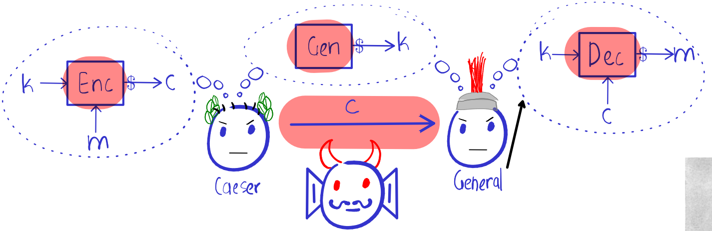
? What does Eve have access to?

  - Description of the algorithms? Yes, Kerckhoffs' principle:

    *'One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.'*

  - What about the key?

# First Let's Try to Model our Eavesdropper Eve




- Can be modelled as an algorithm

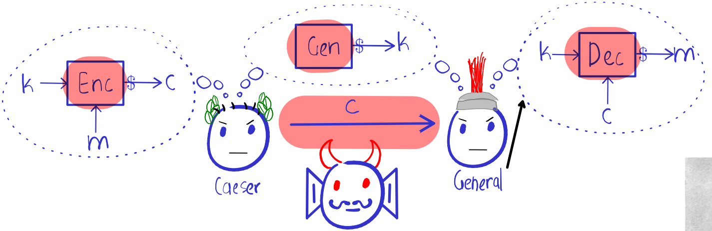
? What does Eve have access to?

  - Description of the algorithms? Yes, Kerckhoffs' principle:

    *'One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.'*

  - What about the key? No, then everything is open
  - Randomness used to derive the key?

# First Let's Try to Model our Eavesdropper Eve




- Can be modelled as an algorithm

? What does Eve have access to?

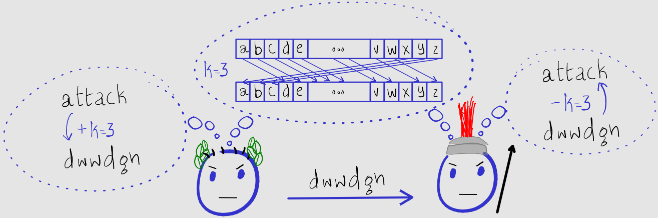
  - Description of the algorithms? Yes, Kerckhoffs' principle:

    *'One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.'*

  - What about the key? No, then everything is open
  - Randomness used to derive the key? No, can then rederive key
  - Randomness used to encrypt?

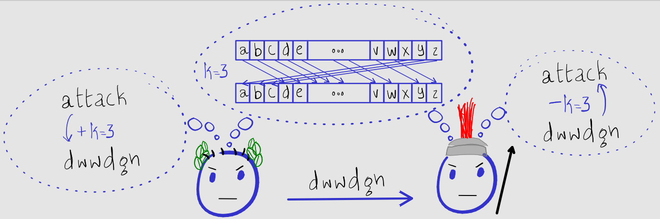
# Shift Cipher (Caeser Cipher)...

## Construction 3




- What can Eve learn?

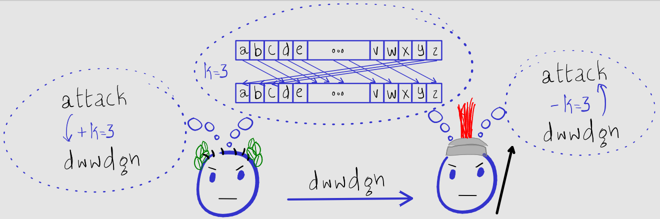
## Construction 3




What can Eve learn?

- Whole message, by exhaustive key search (brute force).

### Construction 3

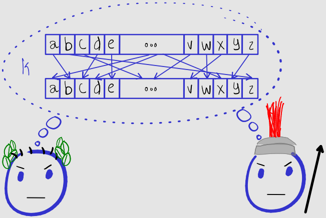



- What can Eve learn?
  - Whole message, by exhaustive key search (brute force).
- What have *we* learnt?
  - *Large-enough* key-space is necessary to thwart *brute force*

### Exercise 2

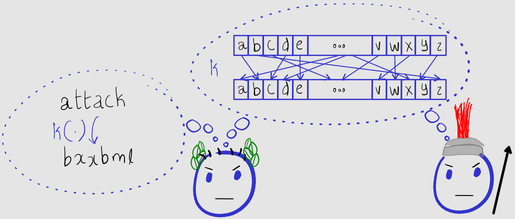
*What happens if the length of the message* $\ell = 1$*?*

**Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)**



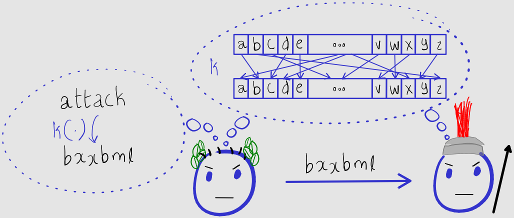

- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



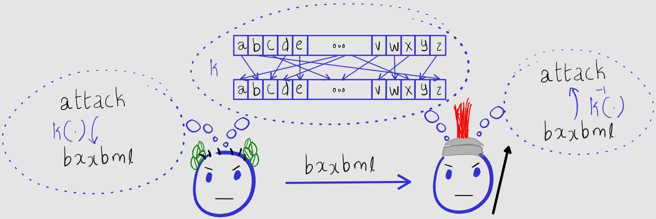

- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

## Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



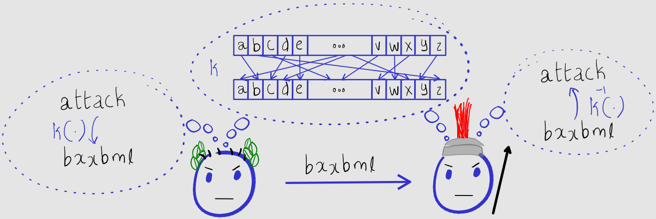

- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

**Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)**




- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

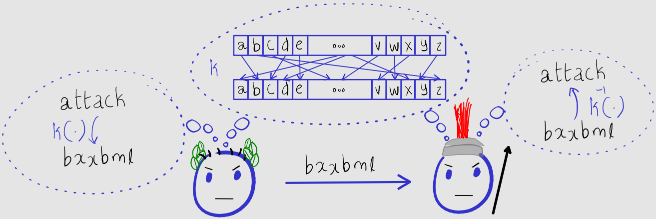
## Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)




- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

? What is the key-space? How large is it?

## Construction 4 (Message space $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$)



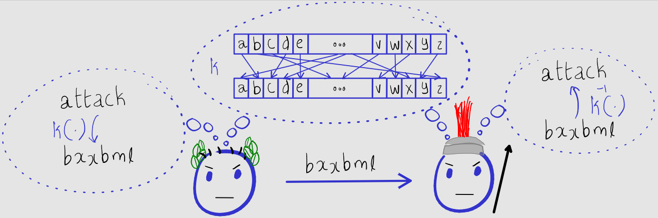

- Key is a *permutation* of $\{\mathtt{a}, \cdots, \mathtt{z}\}$.

? What is the key-space? How large is it?

## Exercise 3

- *Write down the pseudocode for substitution cipher.*
- *Why does correctness of decryption hold?*

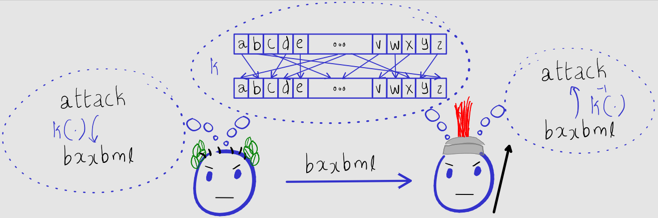
## Construction 5




What can Eve learn?

# Substitution Cipher...

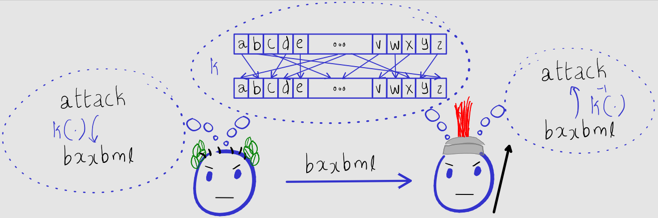
## Construction 5




What can Eve learn?

- Can easily *distinguish* certain messages

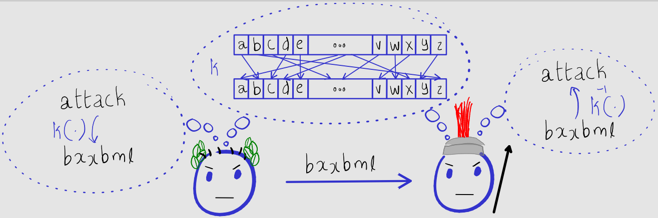
## Construction 5




What can Eve learn?

- Can easily *distinguish* certain messages
- Can recover key with a bit more effort (frequency analysis)

## Construction 5

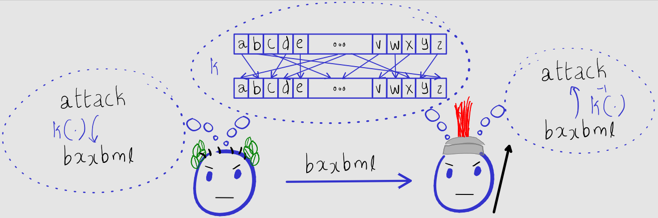



What can Eve learn?

- Can easily *distinguish* certain messages
- Can recover key with a bit more effort (frequency analysis)

- What have *we* learnt?
  - Large key-space maybe necessary, but is not *sufficient*

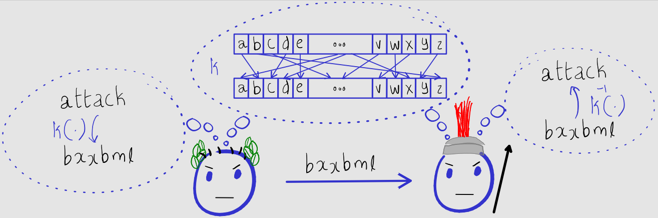
## Construction 5




- What can Eve learn?
  - Can easily *distinguish* certain messages
  - Can recover key with a bit more effort (frequency analysis)

- What have *we* learnt?
  - Large key-space maybe necessary, but is not *sufficient*
  - Must *hide* simple *statistical properties* of the plaintext
    - Should not map a plaintext character to same ciphertext character

## Construction 5




What can Eve learn?

- Can easily *distinguish* certain messages
- Can recover key with a bit more effort (frequency analysis)

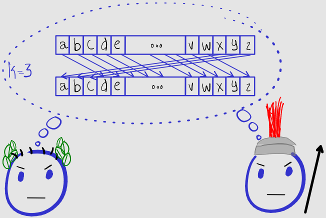
- What have *we* learnt?
  - Large key-space maybe necessary, but is not *sufficient*
  - Must *hide* simple *statistical properties* of the plaintext
    - Should not map a plaintext character to same ciphertext character

# *Polyalphabetic* Ciphers...

- Let's map a plaintext character to different ciphertext characters

- Let's map a plaintext character to different ciphertext characters

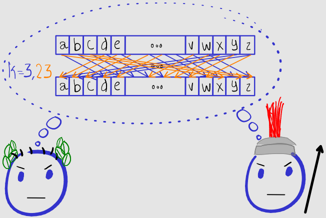
**Construction 6 (*Polyalphabetic* shift cipher (Vignère cipher))**



# *Polyalphabetic* Ciphers...

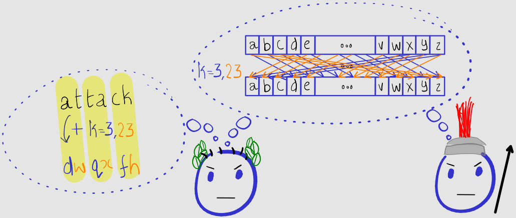
- Let's map a plaintext character to different ciphertext characters

**Construction 6** (*Polyalphabetic* shift cipher (Vignère cipher))



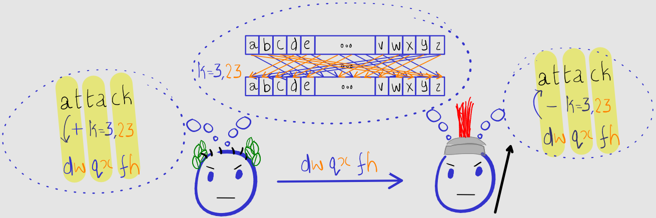
- Let's map a plaintext character to different ciphertext characters

**Construction 6** (*Polyalphabetic* shift cipher (Vignère cipher))



- Let's map a plaintext character to different ciphertext characters

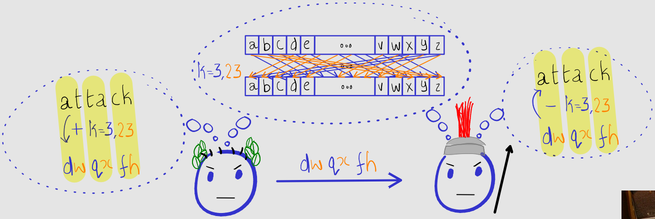
**Construction 6** (*Polyalphabetic* shift cipher (Vignère cipher))



# *Polyalphabetic* Ciphers...

- Let's map a plaintext character to different ciphertext characters

## Construction 6 (*Polyalphabetic* shift cipher (Vignère cipher))

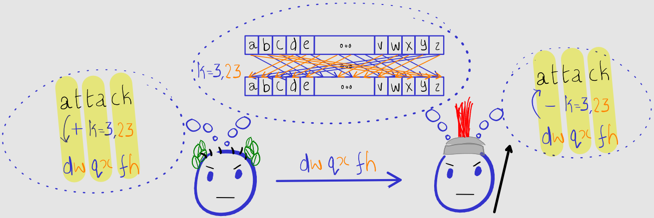



## Exercise 4

1. *Write down the pseudocode for polyalphabetic shift cipher.*
2. *Work out the details of polyalphabetic substitution cipher.*

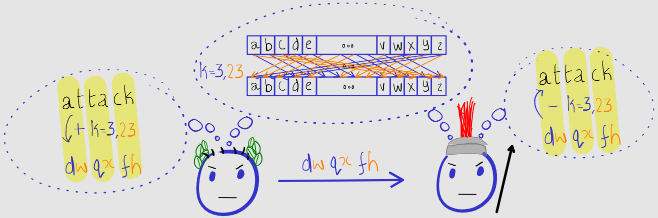
# *Polyalphabetic* Ciphers...

**Construction 7** (*Polyalphabetic* shift cipher (Vignère cipher))



# *Polyalphabetic* Ciphers...

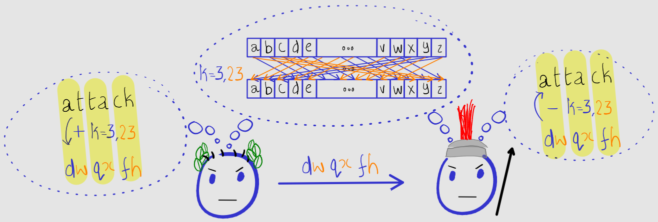
## Construction 7 (*Polyalphabetic* shift cipher (Vignère cipher))




What can Eve learn?

# *Polyalphabetic* Ciphers...

Construction 7 (*Polyalphabetic* shift cipher (Vignère cipher))

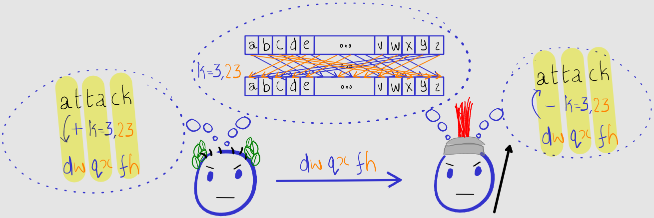



What can Eve learn?

- Can still *distinguish* certain messages. Any guesses?

# *Polyalphabetic* Ciphers...

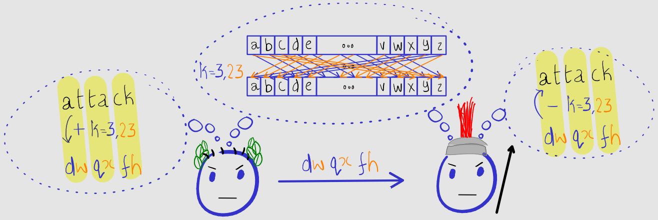
**Construction 7** (*Polyalphabetic* shift cipher (Vignère cipher))




What can Eve learn?

- Can still *distinguish* certain messages. Any guesses?
- Can still recover key (more complicated frequency analysis)

## Construction 7 (*Polyalphabetic* shift cipher (Vignère cipher))




- What can Eve learn?
  - Can still *distinguish* certain messages. Any guesses?
  - Can still recover key (more complicated frequency analysis)
- What have we learnt?
  - Must hide *all* statistical patterns of the plaintext
  - Equivalently: Eve must *learn no information* about the plaintext

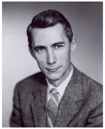
# Plan for this Lecture

# How to Model 'No Information Learnt'?

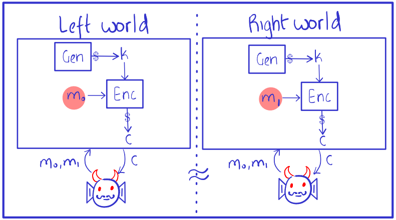
- We will look at two ways:

"Information theoretic"

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$



"Imitation game"



# Modelling 'No Information Learnt': Shannon's Take...

- Intuition: '*observing a ciphertext must have no effect on Eve's knowledge about the message being sent*'

# Modelling 'No Information Learnt': Shannon's Take...

**Definition 2 (Shannon'49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$

- Intuition: '*observing a ciphertext must have no effect on Eve's knowledge about the message being sent*'

# Modelling 'No Information Learnt': Shannon's Take...

**Definition 2 (Shannon'49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

→ Ciphertext distribution induced by M, Gen & Enc

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$

- Intuition: '*observing a ciphertext must have no effect on* Eve*'s knowledge about the message being sent*'

# Modelling ‘No Information Learnt’: Shannon’s Take...

### Definition 2 (Shannon’49)

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

Ciphertext distribution induced by M, Gen & Enc

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | \underbrace{C = c^*}] = \Pr[M = m^*]$$

- Intuition: ‘*observing a ciphertext must have no effect on* Eve*’s knowledge about the message being sent*’

**Definition 2 (Shannon'49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

Ciphertext distribution induced by M, Gen & Enc

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$

- Intuition: '*observing a ciphertext must have no effect on Eve's knowledge about the message being sent*'
- Definition essentially says $M$ and $C$ are *independent* random variables
- Definition *does not* refer to Eve at all!

# Modelling 'No Information Learnt': Shannon's Take...

**Definition 3 (Shannon'49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$

- Let's see why shift cipher is not perfectly secure.

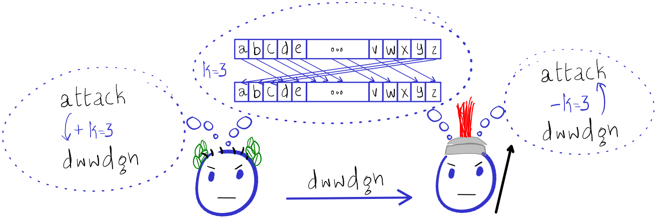
# Modelling 'No Information Learnt': Shannon's Take...

**Definition 3 (Shannon'49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is perfectly-secure if *for any* message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$

- Let's see why shift cipher is not perfectly secure.



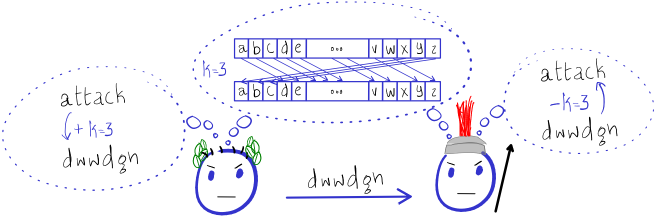
# Modelling ‘No Information Learnt’: Shannon’s Take...

**Definition 3 (Shannon’49)**

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is not perfectly-secure if there exists ~~for any~~ message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] \neq \Pr[M = m^*]$$

- Let’s see why shift cipher is not perfectly secure.



# Modelling 'No Information Learnt': Shannon's Take...

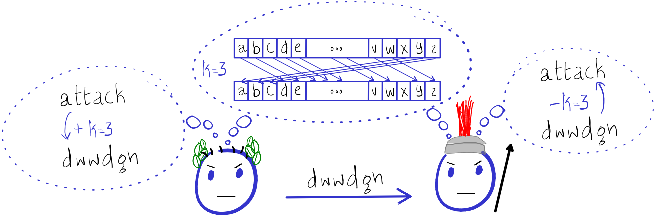
**Definition 3 (Shannon'49)** $\Pr[\text{attack}] = \frac{1}{2} = \Pr[\text{defend}]$

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is not perfectly-secure if ~~for any~~ there exists message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

defend

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] \neq \Pr[M = m^*]$$

- Let's see why shift cipher is not perfectly secure.



# Modelling 'No Information Learnt': Shannon's Take...

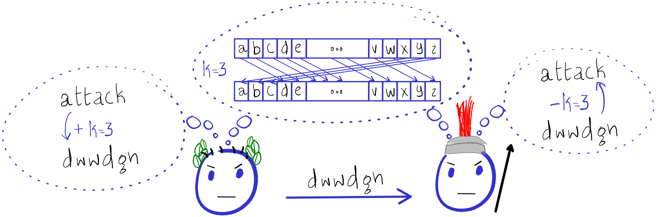
**Definition 3 (Shannon'49)** $\Pr[\text{attack}] = \frac{1}{2} = \Pr[\text{defend}]$

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is not perfectly-secure if there exists ~~for any~~ message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

defend  dwwdgn

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] \neq \Pr[M = m^*]$$

- Let's see why shift cipher is not perfectly secure.



# Modelling 'No Information Learnt': Shannon's Take...

**Definition 3 (Shannon'49)** $\Pr[\text{attack}] = \frac{1}{2} = \Pr[\text{defend}]$

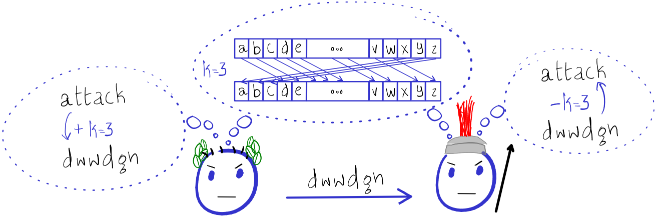
Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an SKE with message space $\mathcal{M}$.
$\Pi$ is not perfectly-secure if ~~*for any*~~ there exists message distribution $M$ over $\mathcal{M}$, message $m^* \in \mathcal{M}$ and ciphertext $c^* \in \mathcal{C}$ (in support):

$m^*$: defend  $c^*$: dwwdgn

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] \neq \Pr[M = m^*]$$

$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*]$: 0!  $\Pr[M = m^*]$: 1/2

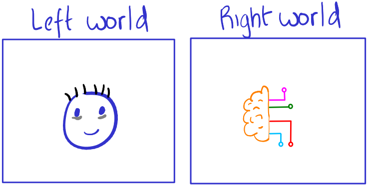
- Let's see why shift cipher is not perfectly secure.



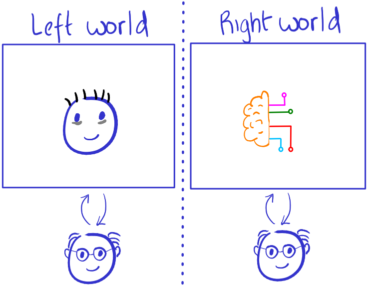
# Modelling ‘No Information Learnt’: Imitation Game...

# Modelling 'No Information Learnt': Imitation Game...

- Turing's Imitation Game (Turing Test)

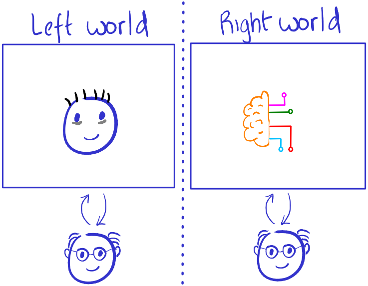


- Turing's Imitation Game (Turing Test)



# Modelling ‘No Information Learnt’: Imitation Game...

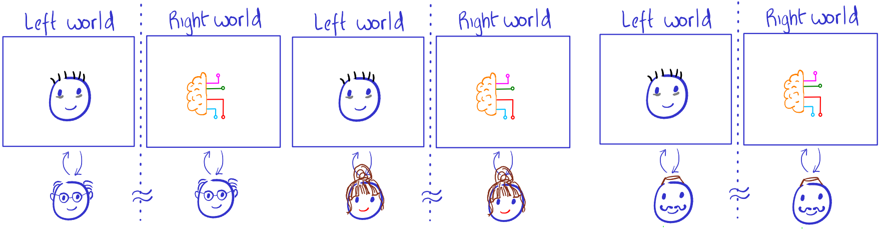
- Turing’s Imitation Game (Turing Test)




- Turing, on artificial intelligence: *"Are there imaginable digital computers which would do well in the imitation game?"*

# Modelling 'No Information Learnt': Imitation Game...

- Turing's Imitation Game (Turing Test)

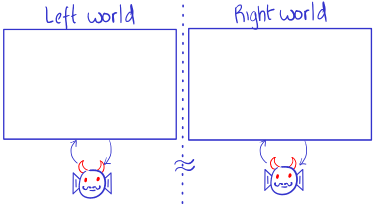



- Turing, on artificial intelligence: *"Are there imaginable digital computers which would do well in the imitation game?"*
- To paraphrase: sign of artificial (human) intelligence if no human can tell the two worlds apart.

≈

# Modelling 'No Information Learnt': Imitation Game...

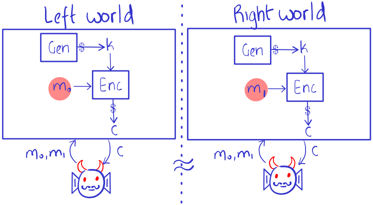
What are our two worlds?



# Modelling ‘No Information Learnt’: Imitation Game...

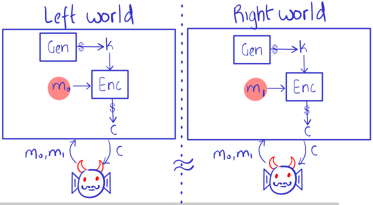
What are our two worlds?

- ‘Left” world: always encrypt $m_0$
  “Right” world: always encrypt $m_1$



# Modelling ‘No Information Learnt’: Imitation Game...

- What are our two worlds?
- ‘Left” world: always encrypt $m_0$
  “Right” world: always encrypt $m_1$




**Definition 4**

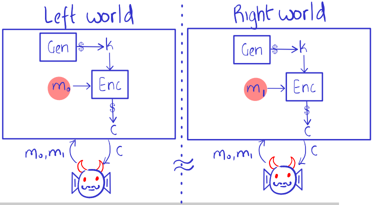
An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly-secure if *for any* eavesdropper *Eve* and messages $(m_0, m_1) \in \mathcal{M}$:

$$\Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_0)}}[\mathsf{Eve}(c) \text{ outputs ‘left'}] = \Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_1)}}[\mathsf{Eve}(c) = \text{ outputs ‘left'}]$$

# Modelling 'No Information Learnt': Imitation Game...

What are our two worlds?

- 'Left" world: always encrypt $m_0$
  "Right" world: always encrypt $m_1$



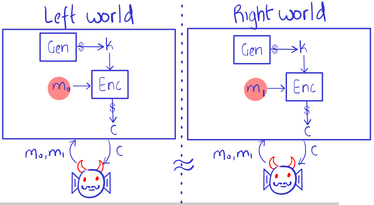

**Definition 4**

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly-secure if *for any* eavesdropper *Eve* and messages $(m_0, m_1) \in \mathcal{M}$:

$$\Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_0)}}[\mathsf{Eve}(c)\ \underbrace{\text{outputs 'left'}}_{0}] = \Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_1)}}[\mathsf{Eve}(c) = \underbrace{\text{outputs 'left'}}_{0}]$$

What are our two worlds?

- ‘Left” world: always encrypt $m_0$
  “Right” world: always encrypt $m_1$




### Definition 4

An SKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly-secure if *for any* eavesdropper *Eve* and messages $(m_0, m_1) \in \mathcal{M}$:

$$\Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_0)}} [\mathsf{Eve}(c) \text{ outputs ‘left’}] = \Pr_{\substack{k \leftarrow \mathsf{Gen} \\ c \leftarrow \mathsf{Enc}(k, m_1)}} [\mathsf{Eve}(c) = \text{ outputs ‘left’}]$$

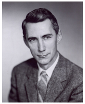
### Exercise 5

*Show that shift and substitution ciphers are not perfectly secure w.r.to above definition.*

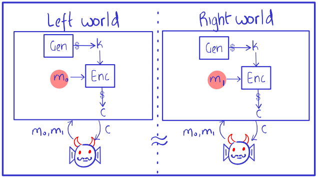
- We saw two definitions.

‘Information theoretic’

$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$



‘imitation game’

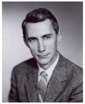


# How to Model ‘No Information Learnt’?...

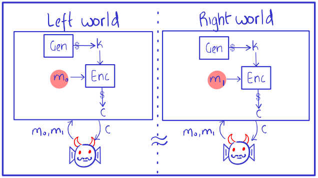
- We saw two definitions. There are two more.

‘Information theoretic’

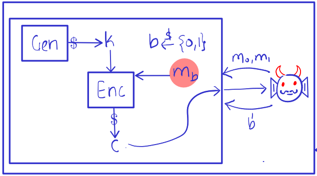
$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$



‘Imitation game’




$$\Pr_{k \leftarrow \mathsf{Gen}}[\mathsf{Enc}(k, m_0) = c^*] = \Pr_{k \leftarrow \mathsf{Gen}}[\mathsf{Enc}(k, m_1) = c^*]$$



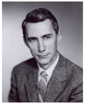

- ‘Semantic-security’: ciphertext contains no info. about plaintext
- Ciphertext indistinguishability: variant of imitation game

# How to Model 'No Information Learnt'?...

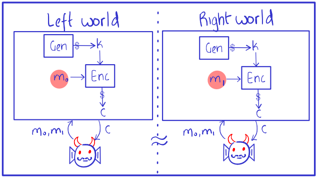
- We saw two definitions. There are two more.

'Information theoretic'

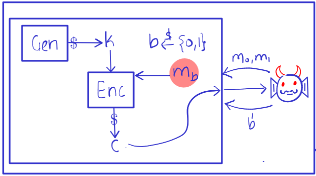
$$\Pr_{k \leftarrow \mathsf{Gen}}[M = m^* | C = c^*] = \Pr[M = m^*]$$



'Imitation game'




$$\Pr_{k \leftarrow \mathsf{Gen}}[\mathsf{Enc}(k, m_0) = c^*] = \Pr_{k \leftarrow \mathsf{Gen}}[\mathsf{Enc}(k, m_1) = c^*]$$

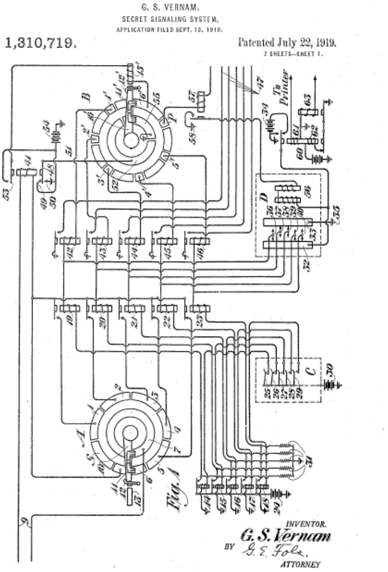



- 'Semantic-security': ciphertext contains no info. about plaintext
- Ciphertext indistinguishability: variant of imitation game

## Exercise 6

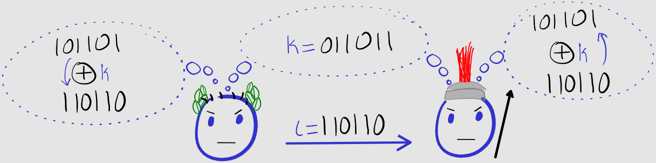
*Show equivalence of all these definitions.*

# One-Time Pad (Vernam' Cipher)...

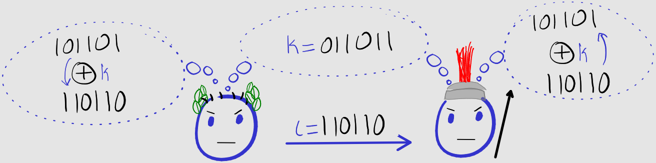


# One-Time Pad (Vernam' Cipher)...

Construction 8 (Message space $\{0,1\}^{\ell=6}$)



**Construction 8 (Message space $\{0,1\}^{\ell=6}$)**

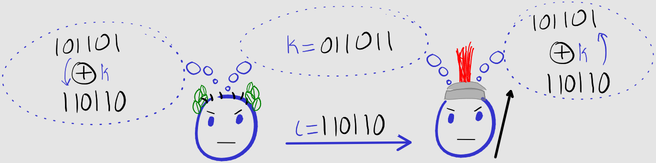



**Pseudocode 2 (Message space $\{0,1\}^{\ell}$)**

- Key generation $\mathsf{Gen}$: output $k \leftarrow \{0,1\}^{\ell}$
- Encryption $\mathsf{Enc}(k,m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k,c)$: output $m := k \oplus c$

# One-Time Pad (Vernam' Cipher)...

## Construction 8 (Message space $\{0,1\}^{\ell}$) $=6$




## Pseudocode 2 (Message space $\{0,1\}^{\ell}$)

- Key generation **Gen**: output $k \leftarrow \{0,1\}^{\ell}$
- Encryption **Enc**$(k, m)$: output $c := k \oplus m$
- Decryption **Dec**$(k, c)$: output $m := k \oplus c$

## Exercise 7

1. *Design OTP for message space* $\{\mathtt{a}, \cdots, \mathtt{z}\}^{\ell}$
2. *How is this different from polyalphabetic shift cipher?*

# One-Time Pad is Perfectly Secure...

Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

Proof.

# One-Time Pad is Perfectly Secure...

**Theorem 5 (Shannon'49)**

*One-time pad is perfectly secure.*

**Proof.**

Goal is to show: $\forall$ Eve, $\forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

# One-Time Pad is Perfectly Secure...

Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

Proof.

Goal is to show: $\forall$ Eve, $\forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \frac{1}{2^\ell} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \frac{1}{2^\ell} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

# One-Time Pad is Perfectly Secure...

Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

Proof.

Goal is to show: $\forall$ Eve, $\forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

# One-Time Pad is Perfectly Secure...

Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

Proof.

Goal is to show: $\forall \text{Eve}, \forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left|\left\{r : \text{Eve}(m_0 \oplus r) = \text{"left"}\right\}\right| = \left|\left\{r : \text{Eve}(m_1 \oplus r) = \text{"left"}\right\}\right|$$

# One-Time Pad is Perfectly Secure...

Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

Proof.

Goal is to show: $\forall \text{Eve}, \forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left|\left\{r : \text{Eve}(m_0 \oplus r) = \text{"left"}\right\}\right| = \left|\left\{r : \text{Eve}(m_1 \oplus r) = \text{"left"}\right\}\right|$$

Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{c : \text{Eve}(c) = \text{"left"}\}$

**Theorem 5 (Shannon'49)**

*One-time pad is perfectly secure.*

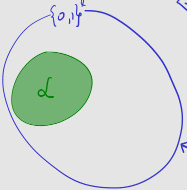
**Proof.**

Goal is to show: $\forall \text{Eve}, \forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left|\left\{r : \text{Eve}(m_0 \oplus r) = \text{"left"}\right\}\right| = \left|\left\{r : \text{Eve}(m_1 \oplus r) = \text{"left"}\right\}\right|$$




Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{c : \text{Eve}(c) = \text{"left"}\}$

# One-Time Pad is Perfectly Secure...

## Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

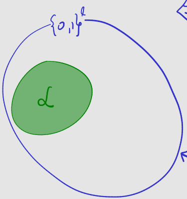
## Proof.

Goal is to show: $\forall$ Eve, $\forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left|\underbrace{\{r : \mathrm{Eve}(m_0 \oplus r) = \text{"left"}\}}_{\mathcal{L} \oplus m_0}\right| = \left|\underbrace{\{r : \mathrm{Eve}(m_1 \oplus r) = \text{"left"}\}}_{\mathcal{L} \oplus m_1}\right|$$




Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{c : \mathrm{Eve}(c) = \text{"left"}\}$

## Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

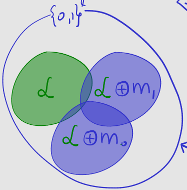
## Proof.

Goal is to show: $\forall \mathrm{Eve}, \forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\mathrm{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left| \underbrace{\{ r : \mathrm{Eve}(m_0 \oplus r) = \text{"left"} \}}_{\mathcal{L} \oplus m_0} \right| = \left| \underbrace{\{ r : \mathrm{Eve}(m_1 \oplus r) = \text{"left"} \}}_{\mathcal{L} \oplus m_1} \right|$$




Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{ c : \mathrm{Eve}(c) = \text{"left"} \}$

# One-Time Pad is Perfectly Secure...

## Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

## Proof.

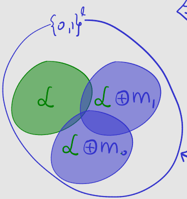
Goal is to show: $\forall \text{Eve}, \forall m_0, m_1 \in \mathcal{M}$

$$\Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \Pr_{r \leftarrow \{0,1\}^\ell}\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_0 \oplus r) = \text{"left"}\right] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr\left[\text{Eve}(m_1 \oplus r) = \text{"left"}\right]$$

$$\Leftrightarrow \left|\{r : \text{Eve}(m_0 \oplus r) = \text{"left"}\}\right| = \left|\{r : \text{Eve}(m_1 \oplus r) = \text{"left"}\}\right|$$

$$|\mathcal{L} \oplus m_0| = |\mathcal{L}| = |\mathcal{L} \oplus m_1|$$




Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{c : \text{Eve}(c) = \text{"left"}\}$

# One-Time Pad is Perfectly Secure...

## Theorem 5 (Shannon'49)

*One-time pad is perfectly secure.*

## Proof.

Goal is to show: $\forall \text{Eve}, \forall m_0, m_1 \in \mathcal{M}$

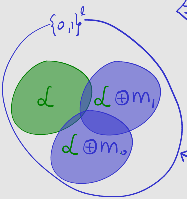
$$\Pr_{r \leftarrow \{0,1\}^\ell}[\text{Eve}(m_0 \oplus r) = \text{"left"}] = \Pr_{r \leftarrow \{0,1\}^\ell}[\text{Eve}(m_1 \oplus r) = \text{"left"}]$$

$$\Leftrightarrow \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr[\text{Eve}(m_0 \oplus r) = \text{"left"}] = \cancel{\frac{1}{2^\ell}} \sum_{r \leftarrow \{0,1\}^\ell} \Pr[\text{Eve}(m_1 \oplus r) = \text{"left"}]$$

$$\Leftrightarrow |\{r : \text{Eve}(m_0 \oplus r) = \text{"left"}\}| = |\{r : \text{Eve}(m_1 \oplus r) = \text{"left"}\}|$$

$$|\mathcal{L} \oplus m_0| = |\mathcal{L}| = |\mathcal{L} \oplus m_1|$$

since $\oplus$ is bijective




Now consider the set $\mathcal{L} \subseteq \{0,1\}^\ell := \{c : \text{Eve}(c) = \text{"left"}\}$

# OTP IRL

# OTP IRL

'Red telephone'

Radio Netherlands Archives

THE NETHERLANDS / HISTORY / AFRICA

## Operation Vula: A secret Dutch network against apartheid

Published 9th September 1999

## Moscow–Washington hotline

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from Moscow-Washington hotline)

# OTP IRL

'Red telephone'

**Moscow–Washington hotline**

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from Moscow-Washington hotline)

**Radio Netherlands Archives**

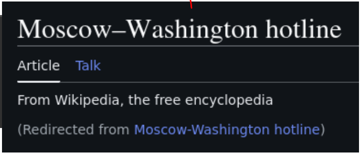
THE NETHERLANDS / HISTORY / AFRICA

## Operation Vula: A secret Dutch network against apartheid

Published 9th September 1999

Why not use OTP for all purposes?

'Red telephone'



Moscow–Washington hotline

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from Moscow-Washington hotline)

Radio Netherlands Archives

THE NETHERLANDS / HISTORY / AFRICA

Operation Vula: A secret Dutch network against apartheid

Published 9th September 1999

- Why not use OTP for all purposes?
  - Keys are as large as messages $|\mathcal{K}| = |\mathcal{M}|$
  - Why not re-use keys? Then it becomes insecure! Why?

# OTP IRL

'Red telephone'

Radio Netherlands Archives

THE NETHERLANDS / HISTORY / AFRICA

Operation Vula: A secret Dutch network against apartheid

Published 9th September 1999

Moscow–Washington hotline

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from Moscow-Washington hotline)

Why not use OTP for all purposes?

- Keys are as large as messages $|\mathcal{K}| = |\mathcal{M}|$
- Why not re-use keys? Then it becomes insecure! Why?

The Register

Declassified files reveal how pre-WW2 Brits smashed Russian crypto

Moscow's agents used one-time pads, er, two times – ой!

Venona project

Article Talk

From Wikipedia, the free encyclopedia

## To Recap

- We saw why classical ciphers are broken by modern standards.

# To Recap

- We saw why classical ciphers are broken by modern standards.
- Learnt some important design principles
  - Kerckhoffs' principles, sufficient key-space...

# To Recap

- We saw why classical ciphers are broken by modern standards.
- Learnt some important design principles
  - Kerckhoffs' principles, sufficient key-space...

# To Recap

- We saw why classical ciphers are broken by modern standards.
- Learnt some important design principles
  - Kerckhoffs' principles, sufficient key-space...
- Used lessons from breaking classical ciphers to formulate perfect secrecy
  - Saw various formulations of perfect secrecy

# To Recap

- We saw why classical ciphers are broken by modern standards.
- Learnt some important design principles
  - Kerckhoffs' principles, sufficient key-space...
- Used lessons from breaking classical ciphers to formulate perfect secrecy
  - Saw various formulations of perfect secrecy
- Saw first formal proof (OTP is perfectly secure)

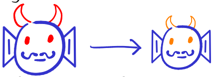
# Next Lecture

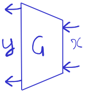
- Shannon's impossibility: $|\mathcal{K}| \geq |\mathcal{M}|$ for any perfectly-secure SKE
- What do you do in face of Shannon's impossibility?

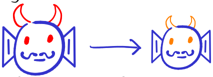
# Next Lecture

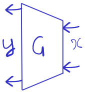
- Shannon's impossibility: $|\mathcal{K}| \geq |\mathcal{M}|$ for any perfectly-secure SKE
- What do you do in face of Shannon's impossibility?
- You compromise.
    - Kerckhoffs' principle
        - *'The system should be, if not theoretically unbreakable, unbreakable in practice.'*

# Next Lecture

- Shannon's impossibility: $|\mathcal{K}| \geq |\mathcal{M}|$ for any perfectly-secure SKE
- What do you do in face of Shannon's impossibility?
- You compromise.
  - Kerckhoffs' principle
    - *'The system should be, if not theoretically unbreakable, unbreakable in practice.'*
  - Restrict to *computationally-bounded* Eve

  - How to model computationally-bounded adversaries?
  - Pseudo-random generators (PRG)



- Shannon's impossibility: $|\mathcal{K}| \geq |\mathcal{M}|$ for any perfectly-secure SKE
- What do you do in face of Shannon's impossibility?
- You compromise.
  - Kerckhoffs' principle
    - *'The system should be, if not theoretically unbreakable, unbreakable in practice.'*
  - Restrict to *computationally-bounded* Eve 
  - How to model computationally-bounded adversaries?
  - Pseudo-random generators (PRG)




More Questions?

# References

1. [KL14, Chapters 1 and 2] for details about this lecture
2. Shannon's paper on perfect secrecy and proof of perfect secrecy one-time pad: [Sha49]
3. Turing's paper on artificial intelligence: [Tur50]

Jonathan Katz and Yehuda Lindell.
*Introduction to Modern Cryptography (3rd ed.).*
Chapman and Hall/CRC, 2014.

C. E. Shannon.
Communication theory of secrecy systems.
*The Bell System Technical Journal*, 28(4):656–715, 1949.

A. M. Turing.
Computing Machinery and Intelligence.
*Mind*, LIX(236):433–460, 10 1950.