Lecture 2: Shannon and Perfect Secrecy

Instructor: Omkant Pandey

Spring 2018 (CSE390)
Last Class

- We discussed some historical ciphers
- ...and how to break them
Last Class

- We discussed some historical ciphers
- ...and how to break them

- This class: a more formal treatment of ciphers.
- Specifically Shannon’s treatment of secure ciphers
Last Class

- We discussed some historical ciphers
- ...and how to break them

- This class: a more formal treatment of ciphers.
- Specifically Shannon’s treatment of secure ciphers

- Volunteer for today’s scribes?
Symmetric Ciphers

- A symmetric cipher consists of:

- A method for generating random keys $k$, denoted by $KG$
- Encryption algorithm: $Enc$
- Decryption algorithm: $Dec$

$Enc$ encrypts messages using a secret key:

- $Enc(k, m) \Rightarrow c$

$Enc$ may use randomness

$Dec$ should decrypt correctly:

@ $k, @ m$: $Dec(k, Enc(k, m)) \Rightarrow m$

The set of all messages $m$ is called message space $M$;
$c$ is called the ciphertext and set of all ciphertexts $C$;

The set of all keys $k$ is called the key space $K$.

Messages $m$ are also known as plaintexts.
Symmetric Ciphers

- A symmetric cipher consists of:
  - A method for generating random keys $k$, denoted by $KG$
  - Encryption algorithm: $\text{Enc}$
  - Decryption algorithm: $\text{Dec}$
A symmetric cipher consists of:
- A method for generating random keys \( k \), denoted by \( KG \)
- Encryption algorithm: \( Enc \)
- Decryption algorithm: \( Dec \)

\( Enc \) encrypts messages using a secret key:
- \( Enc(k, m) \rightarrow c \)
- \( Enc \) may use randomness
- \( c \) is called the ciphertext
A symmetric cipher consists of:
- A method for generating random keys $k$, denoted by $\text{KG}$
- Encryption algorithm: $\text{Enc}$
- Decryption algorithm: $\text{Dec}$

$\text{Enc}$ encrypts messages using a secret key:
- $\text{Enc}(k, m) \rightarrow c$
- $\text{Enc}$ may use randomness
- $c$ is called the ciphertext

$\text{Dec}$ should decrypt correctly:

Symmetric Ciphers

- A symmetric cipher consists of:
  - A method for generating random keys $k$, denoted by KG
  - Encryption algorithm: $Enc$
  - Decryption algorithm: $Dec$

- $Enc$ encrypts messages using a secret key:
  - $Enc(k, m) \rightarrow c$
  - $Enc$ may use randomness
  - $c$ is called the ciphertext

- $Dec$ should decrypt correctly:
  $$\forall k, \forall m : Dec(k, Enc(k, m)) = m.$$
Symmetric Ciphers

- A symmetric cipher consists of:
  - A method for generating random keys $k$, denoted by $\text{KG}$
  - Encryption algorithm: $\text{Enc}$
  - Decryption algorithm: $\text{Dec}$

$\text{Enc}$ encrypts messages using a secret key:
- $\text{Enc}(k, m) \rightarrow c$
- $\text{Enc}$ may use randomness
- $c$ is called the ciphertext

$\text{Dec}$ should decrypt correctly:
\[ \forall k, \forall m : \text{Dec}(k, \text{Enc}(k, m)) = m. \]

- The set of all messages $m$ is called message space $\mathcal{M}$;
- $c$ is called the ciphertext and set of all ciphertexts ciphertext space $\mathcal{C}$;
- The set of all keys $k$ is called the key space $\mathcal{K}$.
Security of a Cipher

What about security?
Security of a Cipher

What about security?

What should it mean **intuitively**?
First attempt: hide the key

- All ciphers in the frequency analysis recover the key...
First attempt: hide the key

- All ciphers in the frequency analysis recover the key...

  What if we just guarantee that key remains completely hidden?

Example from Caesar Cipher:

ATTACK = BUUBDL and DEFEND = EFGFOE

Broken by checking patterns! don't need the key!
First attempt: hide the key

• All ciphers in the frequency analysis recover the key...
  What if we just guarantee that key remains completely hidden?
• No reason why plaintext should be hidden!
• Example from Caesar Cipher:
  ATTACK = BUUBDL and DEFEND = EFGFOE

Broken by checking patterns! don’t need the key!
Second approach: hide the message

- What does it mean?
Second approach: hide the message

- What does it mean?
- Hide the full message only?

What if the ciphertext reveals the frequency of the alphabets in the plaintext?

Dangerous: May be enough to find out if the army will attack or defend?

Hide everything about the message: all possible functions of the message.

Good starting point but impossible! Something about the message may already be known! (E.g., it is in English, starts with "Hello" and today's date, etc.)
Second approach: hide the message

- What does it mean?
- Hide the full message only?
- Hide every letter of the message?
Second approach: hide the message

- What does it mean?
- Hide the full message only?
- Hide every letter of the message?
- What if the ciphertext reveals the frequency of the alphabets in the plaintext?
Second approach: hide the message

- What does it mean?
- Hide the full message only?
- Hide every letter of the message?
- What if the ciphertext reveals the frequency of the alphabets in the plaintext?
- Dangerous: May be enough to find out if the army will attack or defend?
Second approach: hide the message

- What does it mean?
- Hide the full message only?
- Hide every letter of the message?
- What if the ciphertext reveals the frequency of the alphabets in the plaintext?
- Dangerous: May be enough to find out if the army will attack or defend?
- Hide *everything* about the message: all possible functions of the message.
Second approach: hide the message

- What does it mean?
- Hide the full message only?
- Hide every letter of the message?
- What if the ciphertext reveals the frequency of the alphabets in the plaintext?
- Dangerous: May be enough to find out if the army will attack or defend?
- Hide *everything* about the message: all possible functions of the message.
  - Good starting point but impossible! Something about the message may already be known!
    (E.g., it is in English, starts with “Hello” and today’s date, etc.)
Third approach: hide everything that is not already known!

- We cannot hide what may be *a priori* known about the message.
Third approach: hide everything that is not already known!

- We cannot hide what may be *a priori* known about the message.
- The ciphertext must hide everything else!
Third approach: hide everything that is not already known!

- We cannot hide what may be \textit{a priori} known about the message.
- The ciphertext must hide everything else!
- Adversary should not learn any \textbf{NEW} information about the message after seeing the ciphertext.
Third approach: hide everything that is not already known!

- We cannot hide what may be \textit{a priori} known about the message.
- The ciphertext must hide everything else!
- Adversary should not learn any \textbf{NEW} information about the message after seeing the ciphertext.
- How to capture it mathematically?
Shannon’s Treatment

• Messages come from some \textit{distribution}; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.
Shannon’s Treatment

- Messages come from some *distribution*; let \( D \) be a random variable for sampling the messages from the message space \( M \).
- Distribution \( D \) is known to the adversary. This captures *a priori* information about the messages.
Shannon’s Treatment

- Messages come from some *distribution*; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.
- Distribution $D$ is known to the adversary. This captures *a priori* information about the messages.
- The ciphertext $c = \text{Enc}(m, k)$ depends on:
Shannon’s Treatment

- Messages come from some *distribution*; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.

- Distribution $D$ is known to the adversary. This captures *a priori* information about the messages.

- The ciphertext $c = \text{Enc}(m, k)$ depends on:
  - $m$ chosen according to $D$
Shannon’s Treatment

- Messages come from some distribution; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.
- Distribution $D$ is known to the adversary. This captures a priori information about the messages.
- The ciphertext $c = \text{Enc}(m, k)$ depends on:
  - $m$ chosen according to $D$
  - $k$ is chosen randomly (according to $\text{KG}$)
Shannon’s Treatment

- Messages come from some *distribution*; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.
- Distribution $D$ is known to the adversary. This captures *a priori* information about the messages.
- The ciphertext $c = \text{Enc}(m, k)$ depends on:
  - $m$ chosen according to $D$
  - $k$ is chosen randomly (according to $\text{KG}$)
  - $\text{Enc}$ may also use some randomness
Messages come from some distribution; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.

Distribution $D$ is known to the adversary. This captures a priori information about the messages.

The ciphertext $c = Enc(m, k)$ depends on:
- $m$ chosen according to $D$
- $k$ is chosen randomly (according to $KG$)
- $Enc$ may also use some randomness
- These induce a distribution $C$ over the ciphertexts $c$. 
Shannon’s Treatment

- Messages come from some distribution; let $D$ be a random variable for sampling the messages from the message space $\mathcal{M}$.
- Distribution $D$ is known to the adversary. This captures a priori information about the messages.
- The ciphertext $c = \text{Enc}(m, k)$ depends on:
  - $m$ chosen according to $D$
  - $k$ is chosen randomly (according to $\mathcal{K}$)
  - $\text{Enc}$ may also use some randomness
  - These induce a distribution $C$ over the ciphertexts $c$.
- The adversary only observes $c$
  (for some $m \overset{D}{\leftarrow} \mathcal{M}$ and $k \overset{\mathcal{K}}{\leftarrow} \mathcal{K}$, but $m, k$ themselves)
Shannon’s Treatment (continued)

- Knowledge about $m$ **before** observing the output of $C$ is captured by: $D$

- Knowledge about $m$ **after** observing the output of $C$ is captured by:

**Shannon secrecy**: distribution $D$ and $D | C$ must be identical.

Intuitively, this means that:

$C$ contains no NEW information about $m$ ...in the standard sense of information theory.
Shannon’s Treatment (continued)

- Knowledge about $m$ before observing the output of $C$ is captured by: $D$
- Knowledge about $m$ after observing the output of $C$ is captured by: $D|C$
Shannon’s Treatment (continued)

- Knowledge about $m$ before observing the output of $C$ is captured by: $D$

- Knowledge about $m$ after observing the output of $C$ is captured by: $D|C$

- **Shannon secrecy**: distribution $D$ and $D|C$ must be identical.
Knowledge about \( m \) \textbf{before} observing the output of \( C \) is captured by: \( D \)

Knowledge about \( m \) \textbf{after} observing the output of \( C \) is captured by: \( D|C \)

\textbf{Shannon secrecy}: distribution \( D \) and \( D|C \) must be \textit{identical}.

Intuitively, this means that:

\( C \) contains \textbf{no NEW information} about \( m \)

...in the standard sense of information theory.
Definition (Shannon Secrecy)

A cipher \((\mathcal{M}, \mathcal{K}, \mathcal{KG}, \text{Enc}, \text{Dec})\) is **Shannon secure w.r.t a distribution** \(D\) over \(\mathcal{M}\) if for all \(m' \in \mathcal{M}\) and for all \(c\),

\[
\Pr[m \leftarrow D : m = m'] =
\]
Definition (Shannon Secrecy)

A cipher \((\mathcal{M}, \mathcal{K}, KG, Enc, Dec)\) is **Shannon secure w.r.t a distribution** \(D\) over \(\mathcal{M}\) if for all \(m' \in \mathcal{M}\) and for all \(c\),

\[
\Pr[m \leftarrow D : m = m'] = \\
\Pr[k \leftarrow KG, m \leftarrow D : m = m'|Enc(m, k) = c]
\]
Shannon Secrecy

Definition (Shannon Secrecy)

A cipher \((\mathcal{M}, \mathcal{K}, \mathcal{KG}, \text{Enc}, \text{Dec})\) is **Shannon secure w.r.t a distribution** \(D\) over \(\mathcal{M}\) if for all \(m' \in \mathcal{M}\) and for all \(c\),

\[
\Pr [m \leftarrow D : m = m'] = \Pr [k \leftarrow \mathcal{KG}, m \leftarrow D : m = m'|\text{Enc}(m, k) = c]
\]

It is **Shannon secure** if it is Shannon secure w.r.t. all distributions \(D\) over \(\mathcal{M}\).
Questions?
Perfect Secrecy

- Suppose you have two messages: \( m_1 \in \mathcal{M} \) and \( m_2 \in \mathcal{M} \).
Perfect Secrecy

- Suppose you have two messages: $m_1 \in \mathcal{M}$ and $m_2 \in \mathcal{M}$.
- What is the distribution of ciphertexts for $m_1$?
Perfect Secrecy

• Suppose you have two messages: \( m_1 \in \mathcal{M} \) and \( m_2 \in \mathcal{M} \).
• What is the distribution of ciphertexts for \( m_1 \)?

\[
C_1 := \{ k \leftarrow \text{KG}, \text{ output } \text{Enc}(m_1, k) \}
\]
Perfect Secrecy

- Suppose you have two messages: $m_1 \in \mathcal{M}$ and $m_2 \in \mathcal{M}$.
- What is the distribution of ciphertexts for $m_1$?

$$C_1 := \{k \leftarrow \text{KG}, \ \text{output} \ \text{Enc}(m_1, k)\}$$

- Likewise, for $m_2$, the ciphertext distribution is:

$$C_2 := \{k \leftarrow \text{KG}, \ \text{output} \ \text{Enc}(m_2, k)\}$$

Perfect secrecy: $C_1$ and $C_2$ must be identical for every pair of $m_1$, $m_2$.

Ciphertexts are independent of the plaintext(s)!
Perfect Secrecy

- Suppose you have two messages: \( m_1 \in \mathcal{M} \) and \( m_2 \in \mathcal{M} \).
- What is the distribution of ciphertexts for \( m_1 \)?

\[
C_1 := \{k \leftarrow \text{KG}, \text{output } \text{Enc}(m_1, k)\}
\]

- Likewise, for \( m_2 \), the ciphertext distribution is:

\[
C_2 := \{k \leftarrow \text{KG}, \text{output } \text{Enc}(m_2, k)\}
\]

- **Perfect secrecy:**
  \( C_1 \) and \( C_2 \) must be identical for every pair of \( m_1, m_2 \).
Perfect Secrecy

- Suppose you have two messages: \( m_1 \in \mathcal{M} \) and \( m_2 \in \mathcal{M} \).

- What is the distribution of ciphertexts for \( m_1 \)?

  \[ C_1 := \{ k \leftarrow \text{KG}, \ \text{output} \ \text{Enc}(m_1, k) \} \]

- Likewise, for \( m_2 \), the ciphertext distribution is:

  \[ C_2 := \{ k \leftarrow \text{KG}, \ \text{output} \ \text{Enc}(m_2, k) \} \]

- **Perfect secrecy:**
  \( C_1 \) and \( C_2 \) must be **identical for every pair** of \( m_1, m_2 \).

  \[ \Rightarrow \text{Ciphertexts are independent of the plaintext(s)!} \]
Definition (Perfect Secrecy)

Scheme \((M, K, KG, Enc, Dec)\) is **perfectly secure** for every pair of messages \(m_1, m_2\) in \(M\) and for all \(c\),

\[
\Pr[k \in KG : Enc_p(m_1, k) = c] = \Pr[k \in KG : Enc_p(m_2, k) = c]
\]

So much simpler than Shannon Secrecy!

No mention of distributions, a priori or posteriori.

Much easier to work with...
Definition (Perfect Secrecy)

Scheme $(\mathcal{M}, K, KG, Enc, Dec)$ is **perfectly secure** for every pair of messages $m_1, m_2$ in $\mathcal{M}$ and for all $c$,

$$\Pr[k \leftarrow KG : Enc(m_1, k) = c] = \Pr[k \leftarrow KG : Enc(m_2, k) = c]$$
Perfect Secrecy (continued)

**Definition (Perfect Secrecy)**

Scheme \((\mathcal{M}, \mathcal{K}, KG, Enc, Dec)\) is **perfectly secure** for every pair of messages \(m_1, m_2\) in \(\mathcal{M}\) and for all \(c\),

\[
\Pr [k \leftarrow KG : Enc(m_1, k) = c] = \Pr [k \leftarrow KG : Enc(m_2, k) = c]
\]

- So much simpler than Shannon Secrecy!
- No mention of distributions, a priori or posteriori.
- Much easier to work with...
Which notion is better?

- OK, so we have two definitions: perfect secrecy and Shannon secrecy.
- Both of them **intuitively** seem to guarantee great security!
Which notion is better?

- OK, so we have two definitions: perfect secrecy and Shannon secrecy.
- Both of them intuitively seem to guarantee great security!

- Is one better than the other?
- If our intuition is right, shouldn’t they offer “same level” of security?
Theorem (Equivalence Theorem)

A private-key encryption scheme is perfectly secure if and only if it is Shannon secure.
Proof: Simplifying Notation

- We drop \( KG \) and \( D \) when clear from context.
Proof: Simplifying Notation

- We drop $KG$ and $D$ when clear from context.
- $Enc_k(m)$ will be shorthand for $Enc(m, k)$
Proof: Simplifying Notation

- We drop $KG$ and $D$ when clear from context.
- $\text{Enc}_k(m)$ will be shorthand for $\text{Enc}(m, k)$
- For example:
  - $\text{Pr}_m[\ldots]$ means $\text{Pr}[m \leftarrow D : \ldots]$
  - $\text{Pr}_k[\ldots]$ means $\text{Pr}[k \leftarrow KG : \ldots]$
  - $\text{Pr}_{k,m}[\ldots]$ means $\text{Pr}[k \leftarrow KG, m \leftarrow D : \ldots]$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy

Given:

$P \rightarrow m_1, m_2 \rightarrow q \in \mathcal{M}$ and every $c \in \mathcal{C}$:

$\Pr[k, r] \rightarrow Enc(k, p) \rightarrow m_1 \rightarrow q \rightarrow c \rightarrow s$

Show: for every $D$ over $\mathcal{M}$, $m_1 \in \mathcal{M}$, and $c \in \mathcal{C}$:

$\Pr[k, m] \rightarrow m \rightarrow c \rightarrow Enc(k, p) \rightarrow m_1 \rightarrow q \rightarrow c \rightarrow s$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy

Given: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and every $c \in \mathcal{C}$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy

Given: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and every $c \in \mathcal{C}$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$

Show: for every $D$ over $\mathcal{M}$, $m' \in \mathcal{M}$, and $c \in \mathcal{C}$:

$$\Pr_{k,m}[m = m'|\text{Enc}_k(m) = c] = \Pr_m[m = m']$$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

L.H.S. $= \Pr_{k,m}[m = m' | \text{Enc}_k(m) = c]$
Proof: Perfect Secrecy ⇔ Shannon Secrecy (continued)

\[
\text{L.H.S.} = \Pr_{k,m}[m = m'|\text{Enc}_k(m) = c]
\]

\[
= \frac{\Pr_{k,m}[m=m' \cap \text{Enc}_k(m)=c]}{\Pr_{k,m}[\text{Enc}_k(m)=c]}
\]
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

$$\text{L.H.S.} = \Pr_{k,m}[m = m'|\text{Enc}_k(m) = c]$$

$$= \frac{\Pr_{k,m}[m = m' \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\Pr_{k,m}[m = m' \cap \text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

L.H.S.  $= \ Pr_{k,m}[m = m'|\text{Enc}_k(m) = c]$

$= \frac{Pr_{k,m}[m=m' \land \text{Enc}_k(m) = c]}{Pr_{k,m}[\text{Enc}_k(m) = c]}$

$= \frac{Pr_{k,m}[m=m' \land \text{Enc}_k(m') = c]}{Pr_{k,m}[\text{Enc}_k(m) = c]}$

$= \frac{Pr_m[m=m'] \cdot Pr_k[\text{Enc}_k(m') = c]}{Pr_{k,m}[\text{Enc}_k(m) = c]}$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

L.H.S. $= \Pr_{k,m}[m = m' | \text{Enc}_k(m) = c]$

$= \frac{\Pr_{k,m}[m=m' \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$

$= \frac{\Pr_{k,m}[m=m' \cap \text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$

$= \frac{\Pr_m[m=m'] \cdot \Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$

$= \text{R.H.S. } \times \frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

Show:

$$\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]} = 1$$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

Show:

\[
\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]} = 1
\]

Proof:

\[
\Pr_{k,m}[\text{Enc}_k(m) = c] = \]

Instructor: Omkant Pandey  Lecture 2: Shannon and Perfect Secrecy  Spring 2018 (CSE390)
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

Show:

\[
\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]} = 1
\]

Proof:

\[
\Pr_{k,m}[\text{Enc}_k(m) = c] = \sum_{m'' \in \mathcal{M}} \Pr[m = m''] \Pr_k[\text{Enc}_k(m'') = c]
\]
Proof: Perfect Secrecy ⇒ Shannon Secrecy (continued)

Show:

\[
\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]} = 1
\]

Proof:

\[
\Pr_{k,m}[\text{Enc}_k(m) = c] = \sum_{m'' \in M} \Pr_k(m = m'') \Pr_k[\text{Enc}_k(m'') = c]
\]

\[
= \sum_{m'' \in M} \Pr_k(m = m'') \Pr_k[\text{Enc}_k(m') = c]
\]
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

Show:

$$\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_k, m[\text{Enc}_k(m) = c]} = 1$$

Proof:

$$\Pr_{k,m}[\text{Enc}_k(m) = c] = \sum_{m'' \in \mathcal{M}} \Pr[m = m''] \Pr_k[\text{Enc}_k(m'') = c]$$

$$= \sum_{m'' \in \mathcal{M}} \Pr[m = m''] \Pr_k[\text{Enc}_k(m') = c]$$

$$= \Pr_k[\text{Enc}_k(m') = c] \cdot \sum_{m'' \in \mathcal{M}} \Pr[m = m'']$$
Proof: Perfect Secrecy $\Rightarrow$ Shannon Secrecy (continued)

Show:
\[
\frac{\Pr_k[\text{Enc}_k(m') = c]}{\Pr_k,m[\text{Enc}_k(m) = c]} = 1
\]

Proof:
\[
\Pr_{k,m}[\text{Enc}_k(m) = c] = \sum_{m'' \in \mathcal{M}} \Pr_m[m = m''] \Pr_k[\text{Enc}_k(m'') = c]
\]
\[
= \sum_{m'' \in \mathcal{M}} \Pr_m[m = m''] \Pr_k[\underline{\text{Enc}_k(m')} = c]
\]
\[
= \Pr_k[\text{Enc}_k(m') = c] \cdot \sum_{m'' \in \mathcal{M}} \Pr_m[m = m'']
\]
\[
= \Pr_k[\text{Enc}_k(m') = c] \times 1. \quad \text{(QED)}
\]
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show:

$$\forall p_m^1, m^2 \in \mathbb{M}^\ast$$ and $$\forall c$$:

$$\Pr_{k, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_m^1 \mathbf{r})] = \Pr_{k, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_m^2 \mathbf{r})] = \frac{1}{2}$$.

By definition, the scheme is Shannon secure w.r.t. this $$\mathcal{D}$$. Therefore,

$$\Pr_{k, m, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_m \mathbf{r})] = \Pr_{k, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_{m^1} \mathbf{r})]$$,

and

$$\Pr_{k, m, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_m \mathbf{r})] = \Pr_{k, \mathbf{r}}[\mathbf{c} = s \mid \text{Enc}_k(p_{m^2} \mathbf{r})]$$.

Instructor: Omkant Pandey
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and $\forall c$: 

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and $\forall c$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$

Fix any $m_1, m_2, c$ as above.
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and $\forall c$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$

Fix any $m_1, m_2, c$ as above. Let $D$ be the uniform distribution over $\{m_1, m_2\}$ so that:

$$\Pr_m[m = m_1] = \Pr_m[m = m_2] = 1/2.$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in M \times M$ and $\forall c$:

$$\Pr_{k}[\text{Enc}_k(m_1) = c] = \Pr_{k}[\text{Enc}_k(m_2) = c]$$

Fix any $m_1, m_2, c$ as above.
Let $D$ be the uniform distribution over $\{m_1, m_2\}$ so that:

$$\Pr_m[m = m_1] = \Pr_m[m = m_2] = 1/2.$$ 

By definition, the scheme is Shannon secure w.r.t. this $D$. 

Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and $\forall c$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$

Fix any $m_1, m_2, c$ as above. Let $D$ be the uniform distribution over $\{m_1, m_2\}$ so that:

$$\Pr_m[m = m_1] = \Pr_m[m = m_2] = 1/2.$$  

By definition, the scheme is Shannon secure w.r.t. this $D$. Therefore,

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_m[m = m_1],$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy

We have to show: $\forall (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$ and $\forall c$:

$$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$$

Fix any $m_1, m_2, c$ as above.
Let $D$ be the uniform distribution over $\{m_1, m_2\}$ so that:

$$\Pr_m[m = m_1] = \Pr_m[m = m_2] = 1/2.$$

By definition, the scheme is Shannon secure w.r.t. this $D$. Therefore,

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_m[m = m_1], \text{ and}$$

$$\Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c] = \Pr_m[m = m_2]$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c]$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] =$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1 | \text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2 | \text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1 | \text{Enc}_k(m) = c] = \frac{\Pr_{k,m}[m = m_1 \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \frac{\Pr_{k,m}[m = m_1 \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\Pr_m[m = m_1] \cdot \Pr_k\text{Enc}_k(m_1) = c}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1 | \text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2 | \text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1 | \text{Enc}_k(m) = c] = \frac{\Pr_{k,m}[m = m_1 \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\Pr_{m}[m = m_1] \cdot \Pr_{k} \text{Enc}_k(m_1) = c}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\frac{1}{2} \cdot \Pr_{k} \text{Enc}_k(m_1) = c}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \frac{\Pr_{k,m}[m = m_1 \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\Pr_m[m = m_1] \cdot \Pr_k[\text{Enc}_k(m_1) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\frac{1}{2} \cdot \Pr_k[\text{Enc}_k(m_1) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

Likewise, the RHS is:

$$\Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c] = \frac{\frac{1}{2} \cdot \Pr_k[\text{Enc}_k(m_2) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$
Proof: Perfect Secrecy $\iff$ Shannon Secrecy (continued)

Therefore: $\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c]$

Consider the LHS:

$$\Pr_{k,m}[m = m_1|\text{Enc}_k(m) = c] = \frac{\Pr_{k,m}[m = m_1 \cap \text{Enc}_k(m) = c]}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{\Pr_{m}[m = m_1] \cdot \Pr_{k} \text{Enc}_k(m_1) = c}{\Pr_{k,m}[\text{Enc}_k(m) = c]}$$

$$= \frac{1}{2} \cdot \Pr_{k} \text{Enc}_k(m_1) = c$$

Likewise, the RHS is:

$$\Pr_{k,m}[m = m_2|\text{Enc}_k(m) = c] = \frac{1}{2} \cdot \Pr_{k} \text{Enc}_k(m_2) = c$$

Cancel and rearrange. (QED)
Should we go over this proof again?
The *One Time Pad*: A perfect secure scheme

Let \( n \) be an integer = length of the plaintext messages.

**Message space** \( M \):

- \( t_0, t_1 \ldots t_n \) (bit-strings of length \( n \))

**Key space** \( K \):

- \( t_0, t_1 \ldots t_n \) (keys too are length \( n \) bit-strings)

The key is as long as the message.

**Algorithms:**

- \( KG \): samples a key uniformly at random
- \( Enc_p(m, k) \): XOR bit-by-bit.

Let \( m = m_1 m_2 \ldots m_n \) and \( k = k_1 k_2 \ldots k_n \);

Output \( c = c_1 c_2 \ldots c_n \) where \( c_i = m_i \oplus k_i \) for every \( i \).

- \( Dec_p(c, k) \): XOR bit-by-bit.

Return \( m \) where \( m_i = c_i \oplus k_i \) for every \( i \).
The One Time Pad: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
The One Time Pad: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
- Message space $\mathcal{M} := \{0, 1\}^n$ (bit-strings of length $n$)
The One Time Pad: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
- Message space $\mathcal{M} := \{0, 1\}^n$ (bit-strings of length $n$)
- Key space $\mathcal{K} := \{0, 1\}^n$ (keys too are length $n$ bit-strings)
The One Time Pad: A perfect secure scheme

- Let \( n \) be an integer = length of the plaintext messages.
- Message space \( \mathcal{M} := \{0, 1\}^n \) (bit-strings of length \( n \))
- Key space \( \mathcal{K} := \{0, 1\}^n \) (keys too are length \( n \) bit-strings)
- The key is as long as the message
The One Time Pad: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
- Message space $\mathcal{M} := \{0, 1\}^n$ (bit-strings of length $n$)
- Key space $\mathcal{K} := \{0, 1\}^n$ (keys too are length $n$ bit-strings)
- The key is as long as the message
- The algorithms are:
  - $\text{KG}$: samples a key uniformly at random $k \leftarrow \{0, 1\}^n$
  - $\text{Enc}(m, k)$: XOR bit-by-bit,
The One Time Pad: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
- Message space $\mathcal{M} := \{0, 1\}^n$ (bit-strings of length $n$)
- Key space $\mathcal{K} := \{0, 1\}^n$ (keys too are length $n$ bit-strings)
- The key is as long as the message
- The algorithms are:
  - $\text{KG}$: samples a key uniformly at random $k \leftarrow \{0, 1\}^n$
  - $\text{Enc}(m, k)$: XOR bit-by-bit,
    Let $m = m_1 m_2 \ldots m_n$ and $k = k_1 k_2 \ldots k_n$;
The One Time Pad: A perfect secure scheme

- Let \( n \) be an integer = length of the plaintext messages.
- Message space \( \mathcal{M} := \{0, 1\}^n \) (bit-strings of length \( n \))
- Key space \( \mathcal{K} := \{0, 1\}^n \) (keys too are length \( n \) bit-strings)
- The key is as long as the message
- The algorithms are:
  - \( \text{KG} \): samples a key uniformly at random \( k \leftarrow \{0, 1\}^n \)
  - \( \text{Enc}(m, k) \): XOR bit-by-bit,
    Let \( m = m_1m_2\ldots m_n \) and \( k = k_1k_2\ldots k_n \); 
    Output \( c = c_1c_2\ldots c_n \) where \( c_i = m_i \oplus k_i \) for every \( i \in [n] \).
The *One Time Pad*: A perfect secure scheme

- Let $n$ be an integer = length of the plaintext messages.
- Message space $\mathcal{M} := \{0, 1\}^n$ (bit-strings of length $n$)
- Key space $\mathcal{K} := \{0, 1\}^n$ (keys too are length $n$ bit-strings)
- The key is as long as the message
- The algorithms are:
  - $\text{KG}$: samples a key uniformly at random $k \leftarrow \{0, 1\}^n$
  - $\text{Enc}(m, k)$: XOR bit-by-bit,
    Let $m = m_1m_2\ldots m_n$ and $k = k_1k_2\ldots k_n$;
    Output $c = c_1c_2\ldots c_n$ where $c_i = m_i \oplus k_i$ for every $i \in [n]$.
  - $\text{Dec}(c, k)$: XOR bit-by-bit.
    Return $m$ where $m_i = c_i \oplus k_i$ for every $i$. 
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.

Then:

$\text{Enc}(p, m, k) = q^a_m \oplus k$ and $\text{Dec}(p, c, k) = q^c_m \oplus k$.

Ciphertext space is $C$: $\{0, 1\}^n$. Correctness: straightforward.

Perfect secrecy: fix any $m_0, m_1 \in \{0, 1\}^n$ and $c_0, c_1 \in \{0, 1\}^n$. 

$\Pr[k \leftarrow 0, 1, \ldots, n | \text{Enc}(k, m_0) = c_0] = \Pr[k \leftarrow 0, 1, \ldots, n | \text{Enc}(k, m_1) = c_1]$. 

$\Pr[k \leftarrow 0, 1, \ldots, n | \text{Enc}(k, m_0) = c_0] = \frac{1}{2n}$.

QED
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.
Perfect Security of OTP

Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
Perfect Security of OTP

Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $\mathcal{C} := \{0, 1\}^n$. Correctness: straightforward.
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $C := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$. 
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $C := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$.

$$\Pr_{k}[\text{Enc}_k(m) = c] = \Pr[m \oplus k = c]$$
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $\mathcal{C} := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$.

\[
\Pr_k[\text{Enc}_k(m) = c] = \Pr[m \oplus k = c] = \Pr[k = m \oplus c]
\]
Theorem (Perfect security of OTP)

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $\mathcal{C} := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$.

$$\Pr_k[\text{Enc}_k(m) = c] = \Pr[m \oplus k = c] = \Pr[k = m \oplus c] = 2^{-n}.$$
Perfect Security of OTP

**Theorem (Perfect security of OTP)**

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $\mathcal{C} := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$.

$$\Pr_k[\text{Enc}_k(m) = c] = \Pr[m \oplus k = c]$$

$$= \Pr[k = m \oplus c] = 2^{-n}.$$ 

$$\Pr_k[\text{Enc}_k(m) = c] = 0 \quad (\forall c \notin \{0, 1\}^n)$$
Perfect Security of OTP

**Theorem (Perfect security of OTP)**

One Time Pad is a perfectly secure private-key encryption scheme.

- Let $a \oplus b$ for $n$-bit strings $a, b$ mean bit-wise XOR.
- Then: $\text{Enc}(m, k) = m \oplus k$ and $\text{Dec}(c, k) = c \oplus k$.
- Ciphertext space is $C := \{0, 1\}^n$. Correctness: straightforward.
- Perfect secrecy: fix any $m \in \{0, 1\}^n$ and $c \in \{0, 1\}^n$.

\[
\Pr_k[\text{Enc}_k(m) = c] = \Pr[m \oplus k = c] = \Pr[k = m \oplus c] = 2^{-n}.
\]

\[
\Pr_k[\text{Enc}_k(m) = c] = 0 \quad (\forall c \notin \{0, 1\}^n)
\]

$\Rightarrow \forall (m_1, m_2) \in \{0, 1\}^{n \times n}$ and $\forall c$:

$\Pr_k[\text{Enc}_k(m_1) = c] = \Pr_k[\text{Enc}_k(m_2) = c]$. (QED)
Some Remarks

- The One Time Pad (OTP) scheme is also known as the Vernam Cipher.
Some Remarks

- The One Time Pad (OTP) scheme is also known as the **Vernam Cipher**.
- The Caesar Cipher is just OTP for 1-alphabet messages!
Some Remarks

- The One Time Pad (OTP) scheme is also known as the Vernam Cipher.
- The Caesar Cipher is just OTP for 1-alphabet messages!
- Mathematically:
  - XOR is the same as addition modulo 2:
    \[ a + b \mod 2. \]
  - Caesar Cipher for 1-alphabet is addition modulo 26.
Some Remarks

- The One Time Pad (OTP) scheme is also known as the Vernam Cipher.
- The Caesar Cipher is just OTP for 1-alphabet messages!
- Mathematically:
  - XOR is the same as addition modulo 2:
    \[ a + b \mod 2. \]
  - Caesar Cipher for 1-alphabet is addition modulo 26.
  - You can work modulo any number \( n \).
Some Remarks

- The One Time Pad (OTP) scheme is also known as the **Vernam Cipher**.
- The Caesar Cipher is just OTP for 1-alphabet messages!
- Mathematically:
  - XOR is the same as **addition modulo 2**: 
    \[ a + b \mod 2. \]
  - Caesar Cipher for 1-alphabet is **addition modulo 26**.
  - You can work modulo any number \( n \)
- As the name suggests, one key can be used only **once**.
- The key must be:
  - sampled uniformly **every time**, and
  - be **as long as** the message.
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...

Instructor: Omkant Pandey
Lecture 2: Shannon and Perfect Secrecy Spring 2018 (CSE390)
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...
  - 80 GB long key to encrypt 80 GB data
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...
  - 80 GB long key to encrypt 80 GB data
  - 80 GB space to store this key in a safe place
    (other than your hard drive)
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...
  - 80 GB long key to encrypt 80 GB data
  - 80 GB space to store this key in a safe place (other than your hard drive)
  - Key for OTP is uniform, so it cannot be compressed either!
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...
  - 80 GB long key to encrypt 80 GB data
  - 80 GB space to store this key in a safe place (other than your hard drive)
  - Key for OTP is uniform, so it cannot be compressed either!
  - This is never done in practice...
Key Length in Perfectly Secure Encryption

- If the key has to be as long as the message, it is a serious problem!
- Imagine encrypting your machine’s hard drive with a OTP...
  - 80 GB long key to encrypt 80 GB data
  - 80 GB space to store this key in a safe place (other than your hard drive)
  - Key for OTP is uniform, so it cannot be compressed either!
    - This is never done in practice...
- OTP looks naïve, quite elementary: can’t we design a more sophisticated scheme with shorter keys?
Shannon’s Theorem

Theorem (Shannon’s Theorem)

For every perfectly secure cipher $\text{Enc}, \text{Dec}$ with message space $M$ and key space $K$, it holds that $|K| \leq |M|$.

Some Remarks:

Message length is $n = \log |M|$ and key length is $\ell = \log |K|$.

It follows that $\ell \leq n$, i.e., keys must be as long as the messages.
Theorem (Shannon’s Theorem)

For every perfectly secure cipher \((\text{Enc, Dec})\) with message space \(\mathcal{M}\) and key space \(\mathcal{K}\), it holds that \(|\mathcal{K}| \geq |\mathcal{M}|\).
Shannon’s Theorem

Theorem (Shannon’s Theorem)

For every perfectly secure cipher \((\text{Enc}, \text{Dec})\) with message space \(\mathcal{M}\) and key space \(\mathcal{K}\), it holds that \(|\mathcal{K}| \geq |\mathcal{M}|\).

Some Remarks:

- Message length is \(n = \lg |\mathcal{M}|\) and key length is \(\ell = \lg |\mathcal{K}|\).
- It follows that \(\ell \geq n\), i.e., keys must be as long as the messages.
Exercise: Reusing OTP

What could go wrong if you re-use a OTP anyway?

If we could re-use then we could encrypt longer messages with shorter keys.

Simply break the message in shorter parts.

Therefore, by Shannon's Theorem, the resulting scheme will not be perfectly secure.

Even worse — it will be open to the frequency attack! (just like Vigènere Cipher)

In fact, lots of neat examples where reusing OTP leaks clear patterns.

Can you construct such examples?

Instructor: Omkant Pandey
Exercise: Reusing OTP

What could go wrong if you re-use a OTP anyway?
Exercise: Reusing OTP

- What could go wrong if you re-use a OTP anyway?
- If we could re-use then we could encrypt longer messages with shorter keys.
- Simply break the message in shorter parts.
Exercise: Reusing OTP

- What could go wrong if you re-use a OTP anyway?
- If we could re-use then we could encrypt longer messages with shorter keys.
- Simply break the message in shorter parts.
- Therefore, by Shannon’s Theorem, the resulting scheme will not be perfectly secure.
Exercise: Reusing OTP

- What could go wrong if you re-use a OTP anyway?
- If we could re-use then we could encrypt longer messages with shorter keys.
- Simply break the message in shorter parts.
- Therefore, by Shannon’s Theorem, the resulting scheme will not be perfectly secure.
- Even worse — it will be open to the frequency attack! (just like Vigènere Cipher)
- In fact, lots of neat examples where reusing OTP leaks clear patterns.
Exercise: Reusing OTP

- What could go wrong if you re-use a OTP anyway?
- If we could re-use then we could encrypt longer messages with shorter keys.
- Simply break the message in shorter parts.
- Therefore, by Shannon’s Theorem, the resulting scheme will not be perfectly secure.
- Even worse — it will be open to the frequency attack! (just like Vigènere Cipher)
- In fact, lots of neat examples where reusing OTP leaks clear patterns.
- Can you construct such examples?
Shannon’s Theorem on key length is pretty bad news for perfect ciphers.
Shannon’s Theorem on key length is pretty bad news for perfect ciphers.

It means we really have to give up on perfect secrecy for practical applications, unless we absolutely need it.
Shannon’s Theorem on key length is pretty bad news for perfect ciphers.

It means we really have to give up on perfect secrecy for practical applications, unless we absolutely need it.

This is really the dawn of modern cryptography: we want to construct something that is “just as good for practical purposes.”