Semantic Security

Rohit Musti

CUNY - Hunter College

February 9, 2022

Overview

- Alice and Bob share a secret key k.

Overview

- Alice and Bob share a secret key k.
- Alice wants to send a message m to Bob through an untrusted medium.

Overview

- Alice and Bob share a secret key k.
- Alice wants to send a message m to Bob through an untrusted medium.
- Mechanism: Shannon Ciphers.

Who is Claude Shannon?

- Father of Information Theory

Who is Claude Shannon?

- Father of Information Theory
- Friends with Alan Turing (they actually compared notes on Turing's famous paper the Universal Turing machine)

Who is Claude Shannon?

- Father of Information Theory
- Friends with Alan Turing (they actually compared notes on Turing's famous paper the Universal Turing machine)
- Estimated the complexity of chess

Who is Claude Shannon?

- Father of Information Theory
- Friends with Alan Turing (they actually compared notes on Turing's famous paper the Universal Turing machine)
- Estimated the complexity of chess
- First to formalize the notion of security

Shannon Cipher

- **Definition:** a Shannon Cipher is a pair of functions (E, D)

Shannon Cipher

- **Definition:** a Shannon Cipher is a pair of functions (E, D)
- The E function takes in two arguments: a key k and a message m, and produces a ciphertext c

$$c = E(k, m)$$

Shannon Cipher

- **Definition:** a Shannon Cipher is a pair of functions (E, D)
- The E function takes in two arguments: a key k and a message m, and produces a ciphertext c

$$c = E(k, m)$$

- The D function takes in two arguments: a key k and a ciphertext c, and produces a plaintext m

Shannon Cipher

- **Definition:** a Shannon Cipher is a pair of functions (E, D)
- The E function takes in two arguments: a key k and a message m, and produces a ciphertext c

$$c = E(k, m)$$

- The D function takes in two arguments: a key k and a ciphertext c, and produces a plaintext m
- While E maybe random, D must be deterministic

$$m = D(k, c)$$

Shannon Cipher: Correctness

- Decryption must *undo* Encryption

Shannon Cipher: Correctness

- Decryption must *undo* Encryption

$$m = D(k, E(k, m))$$

- More formally: let K be the key space, M be the message space, and C be the ciphertext space.

Shannon Cipher: Correctness

- Decryption must *undo* Encryption

$$m = D(k, E(k, m))$$

- More formally: let K be the key space, M be the message space, and C be the ciphertext space.

$$E : K \times M \to C$$

Shannon Cipher: Correctness

- Decryption must *undo* Encryption

$$m = D(k, E(k, m))$$

- More formally: let K be the key space, M be the message space, and C be the ciphertext space.

$$E : K \times M \to C$$

$$D : K \times C \to M$$

Shannon Cipher: Correctness

- Decryption must *undo* Encryption

$$m = D(k, E(k, m))$$

- More formally: let K be the key space, M be the message space, and C be the ciphertext space.

$$E : K \times M \to C$$

$$D : K \times C \to M$$

- ε is defined over (K, M, C)

Revisit: The One Time Pad

- A one time pad is a Shannon Cipher s.t.

$$K := M := C := \{0, 1\}^L$$

Revisit: The One Time Pad

- A one time pad is a Shannon Cipher s.t.

$$K := M := C := \{0,1\}^L$$

- Encryption for a key $k \in \{0,1\}^L$ and a message $m \in \{0,1\}^L$:

$$E(k,m) := k \oplus m$$

Revisit: The One Time Pad

- A one time pad is a Shannon Cipher s.t.

$$K := M := C := \{0,1\}^L$$

- Encryption for a key $k \in \{0,1\}^L$ and a message $m \in \{0,1\}^L$:

$$E(k,m) := k \oplus m$$

- Decryption for a key $k \in \{0,1\}^L$ and a ciphertext $c \in \{0,1\}^L$:

$$D(k,c) := k \oplus c$$

Revisit: The One Time Pad

- A one time pad is a Shannon Cipher s.t.

$$K := M := C := \{0,1\}^L$$

- Encryption for a key $k \in \{0,1\}^L$ and a message $m \in \{0,1\}^L$:

$$E(k,m) := k \oplus m$$

- Decryption for a key $k \in \{0,1\}^L$ and a ciphertext $c \in \{0,1\}^L$:

$$D(k,c) := k \oplus c$$

- Future HW: Write out a mathmatical justification that the one time pad meets the earlier correctness requirement (Decryption *undoes* Encryption)

Security Requirements 1

- key assumption: adversaries know the encryption mechanism and distribution of K

Security Requirements 1

- key assumption: adversaries know the encryption mechanism and distribution of K
- If Alice encrypts a message m with a key k and an adversary obtains the ciphertext c, the key k needs to be hard to guess

Security Requirements 1

- key assumption: adversaries know the encryption mechanism and distribution of K
- If Alice encrypts a message m with a key k and an adversary obtains the ciphertext c, the key k needs to be hard to guess (if k is easy to guess, then the adversary will just guess until they discover k).

Security Requirements 1

- key assumption: adversaries know the encryption mechanism and distribution of K
- If Alice encrypts a message m with a key k and an adversary obtains the ciphertext c, the key k needs to be hard to guess (if k is easy to guess, then the adversary will just guess until they discover k).
- Thus, k must be chosen uniformly & random from a large keyspace K

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.
- Supposed $m_0 = meet\_at\_lunch\_time$ and $m_1 = meet\_at\_snack\_time$,

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.
- Supposed $m_0 = meet\_at\_lunch\_time$ and $m_1 = meet\_at\_snack\_time$, our adversary has a 50% chance of guessing correctly.

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.
- Supposed $m_0 = meet\_at\_lunch\_time$ and $m_1 = meet\_at\_snack\_time$, our adversary has a 50% chance of guessing correctly.
- A secure cipher text should not increase this probability of guessing correctly.

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.
- Supposed $m_0 = meet\_at\_lunch\_time$ and $m_1 = meet\_at\_snack\_time$, our adversary has a 50% chance of guessing correctly.
- A secure cipher text should not increase this probability of guessing correctly.
- Suppose there are 90 keys k_0 s.t. $E(k_0, m_0) = c$ and 10 keys k_1 s.t. $E(k_1, m_1) = c$,

Security Requirements 2

- key assumption: adversaries may have some knowledge of the message being encrypted.
- Supposed $m_0 = meet\_at\_lunch\_time$ and $m_1 = meet\_at\_snack\_time$, our adversary has a 50% chance of guessing correctly.
- A secure cipher text should not increase this probability of guessing correctly.
- Suppose there are 90 keys k_0 s.t. $E(k_0, m_0) = c$ and 10 keys k_1 s.t. $E(k_1, m_1) = c$, the probability that a given message c is m_0 is $90/(90 + 10) = 90\%$, increasing our adversaries odds of guessing correctly.

Perfect Security

- Assumption: a key is drawn uniformly and randomly from a large key space: $\mathbf{k}$.

Perfect Security

- Assumption: a key is drawn uniformly and randomly from a large key space: **k**.
- Let (E, D) be a Shannon Cipher defined over (K, M, C)

Perfect Security

- Assumption: a key is drawn uniformly and randomly from a large key space: **k**.
- Let (E, D) be a Shannon Cipher defined over (K, M, C)
- If for all $m_0, m_1 \in M$, for all $c \in C$, and $\mathbf{k} \xleftarrow{R} K$ we have

$$Pr[E(\mathbf{k}, m_0) = c] = Pr[E(\mathbf{k}, m_1) = c]$$

One Time Pad: Perfectly Security

- A result of our definition of perfect security is

$$|\{k \xleftarrow{R} K : E(k, m) = c\}| = N_c$$

One Time Pad: Perfectly Security

- A result of our definition of perfect security is

$$|\{k \xleftarrow{R} K : E(k, m) = c\}| = N_c$$

- For any message $m \in \{0,1\}^l$ and cipher text $c \in \{0,1\}^l$,

$$k \oplus m = c$$

One Time Pad: Perfectly Security

- A result of our definition of perfect security is

$$|\{k \xleftarrow{R} K : E(k,m) = c\}| = N_c$$

- For any message $m \in \{0,1\}^l$ and cipher text $c \in \{0,1\}^l$,

$$k \oplus m = c$$

thus, $N_c = 1$, satisfiying the above result.

Problems with Perfect Security

- Let (E, D) be a Shannon Cipher defined over (K, M, C), if this cipher is perfectly secure, then $|K| \geq |M|$

Problems with Perfect Security

- Let (E, D) be a Shannon Cipher defined over (K, M, C), if this cipher is perfectly secure, then $|K| \geq |M|$
- It is impractical to have keys that are at least as large as the size of the message we are transmitting .

Problems with Perfect Security

- Let (E, D) be a Shannon Cipher defined over (K, M, C), if this cipher is perfectly secure, then $|K| \geq |M|$
- It is impractical to have keys that are at least as large as the size of the message we are transmitting .
- Really, what we are concerned about are real world, computationally bounded adversaries.

Problems with Perfect Security

- Let (E, D) be a Shannon Cipher defined over (K, M, C), if this cipher is perfectly secure, then $|K| \geq |M|$
- It is impractical to have keys that are at least as large as the size of the message we are transmitting .
- Really, what we are concerned about are real world, computationally bounded adversaries.
- We are also interested in efficient algorithms for encrypting and decrypting. By efficient, we mean polynomial functions.

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.

Experiment b

- Let $\mathcal{E} = (E, D)$ be a Cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.
Experiment b

- Let $\mathcal{E} = (E, D)$ be a Cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- $\mathcal{A}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.
Experiment b

- Let $\mathcal{E} = (E, D)$ be a Cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- $\mathcal{A}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.
Experiment b

- Let $\mathcal{E} = (E, D)$ be a Cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- $\mathcal{A}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to adversary

Semantic Security

Intuition: the probability that a computationally bounded adversary can learn anything about a message m given its cipher text c is negligible.

Experiment b

- Let $\mathcal{E} = (E, D)$ be a Cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- $\mathcal{A}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to adversary
- $\mathcal{A}$ outputs b^*, guessing which experiment was selected by the Challenger

Semantic Security Advantage

- The *Semantic Security Advantage (SSA)* of the adversary $\mathcal{A}$, is

Semantic Security Advantage

- The *Semantic Security Advantage (SSA)* of the adversary $\mathcal{A}$, is

$$SSA[\mathcal{A}, \mathcal{E}] := |Pr[W_0] - Pr[W_1]|$$

where W_b is the probability that $\mathcal{A}$ outputs 1 in experiment b

Semantic Security Advantage

- The *Semantic Security Advantage (SSA)* of the adversary $\mathcal{A}$, is

$$SSA[\mathcal{A}, \mathcal{E}] := |Pr[W_0] - Pr[W_1]|$$

  where W_b is the probability that $\mathcal{A}$ outputs 1 in experiment b

- At random, $\mathcal{A}$ has a 50% probability of outputting the correct bit, yielding

$$SSA[\mathcal{A}, \mathcal{E}] := |Pr[W_0] = Pr[W_1]| = 0$$

Semantic Security Advantage

- The *Semantic Security Advantage (SSA)* of the adversary $\mathcal{A}$, is

$$SSA[\mathcal{A}, \mathcal{E}] := |Pr[W_0] - Pr[W_1]|$$

  where W_b is the probability that $\mathcal{A}$ outputs 1 in experiment b

- At random, $\mathcal{A}$ has a 50% probability of outputting the correct bit, yielding

$$SSA[\mathcal{A}, \mathcal{E}] := |Pr[W_0] = Pr[W_1]| = 0$$

- In order to be semantically secure, the advantage has to be negligible. We can think of negligible as $1 + \lambda \approx 1, \lambda = 2^{-100}$

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text.

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text. Here is our security game for message recovery.

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text. Here is our security game for message recovery.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text. Here is our security game for message recovery.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- Challenger generates $k \xleftarrow{R} \mathcal{K}$, $m \xleftarrow{R} \mathcal{M}$ and computes $E(k, m) = c$ and send c to the adversary $\mathcal{A}$

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text. Here is our security game for message recovery.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- Challenger generates $k \xleftarrow{R} \mathcal{K}$, $m \xleftarrow{R} \mathcal{M}$ and computes $E(k, m) = c$ and send c to the adversary $\mathcal{A}$
- $\mathcal{A}$ outputes message m^*

Semantic Security to Message Recovery

Semantic Security guarantees that an adversary cannot recover a message from a cipher text. Here is our security game for message recovery.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{A}$ be computationally bounded
- Challenger generates $k \xleftarrow{R} \mathcal{K}$, $m \xleftarrow{R} \mathcal{M}$ and computes $E(k, m) = c$ and send c to the adversary $\mathcal{A}$
- $\mathcal{A}$ outputes message m^*

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to $\mathcal{B}$

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to $\mathcal{B}$
- $\mathcal{B}$ sends c to $\mathcal{A}$, simulating the message recovery game.

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to $\mathcal{B}$
- $\mathcal{B}$ sends c to $\mathcal{A}$, simulating the message recovery game.
- $\mathcal{A}$ returns m^*. If $m^* = m_1$, $\mathcal{B}$ returns 1, else it returns 0.

Semantic Security to Message Recovery: cont

Now let's use our adversary $\mathcal{A}$ to break semantic security, violating our original assumption.

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ and let our adversary $\mathcal{B}$ be computationally bounded
- $\mathcal{B}$ picks $m_0, m_1 \in \mathcal{M}$ and sends to challenger
- Challenger selects $k \xleftarrow{R} \mathcal{K}$
- Challenger computes $E(k, m_b) := c$ and sends c to $\mathcal{B}$
- $\mathcal{B}$ sends c to $\mathcal{A}$, simulating the message recovery game.
- $\mathcal{A}$ returns m^*. If $m^* = m_1$, $\mathcal{B}$ returns 1, else it returns 0.

Thus, if we can recover the message, we have broken semantic security.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)
- Alice must first share k_0 with $router_0$ and k_1 with $router_1$.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)
- Alice must first share k_0 with $router_0$ and k_1 with $router_1$.
- Then Alice sends $E(k_0, E(k_1, m))$ to router 1.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)
- Alice must first share k_0 with $router_0$ and k_1 with $router_1$.
- Then Alice sends $E(k_0, E(k_1, m))$ to router 1.
- $router_1$ decrypts and sends the message $E(k_1, m)$ to $router_2$ in random order.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)
- Alice must first share k_0 with $router_0$ and k_1 with $router_1$.
- Then Alice sends $E(k_0, E(k_1, m))$ to router 1.
- $router_1$ decrypts and sends the message $E(k_1, m)$ to $router_2$ in random order.
- $router_2$ decrypts and sends the message m to Bob in random order.

Derive Anonymous Routing from Semantic Security

- Let $\mathcal{E} = (E, D)$ be a semantically secure cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$
- Suppose Alice wants to send information to Bob, but doesn't want Bob to know that the message came from her. (example: you want a radiologist to review a patient's scans without revealing the patients identity)
- Alice must first share k_0 with $router_0$ and k_1 with $router_1$.
- Then Alice sends $E(k_0, E(k_1, m))$ to router 1.
- $router_1$ decrypts and sends the message $E(k_1, m)$ to $router_2$ in random order.
- $router_2$ decrypts and sends the message m to Bob in random order.

Alice's security is 1 n where n is the number of messages sent along the network.