

Claude Shannon

CSL 505

CRYPTOGRAPHY

Lecture 4

Perfect Secrecy

Instructor
Dr. Dhiman Saha

Communication Theory of Secrecy Systems*

By C. E. SHANNON

1. INTRODUCTION AND SUMMARY

THE problems of cryptography and secrecy systems furnish an interesting application of communication theory.¹ In this paper a theory of secrecy systems is developed. The approach is on a theoretical level and is intended to complement the treatment found in standard works on cryptography.² There, a detailed study is made of the many standard types of codes and ciphers, and of the ways of breaking them. We will be more concerned with the general mathematical structure and properties of secrecy systems.

The treatment is limited in certain ways. First, there are three general types of secrecy system: (1) concealment systems, including such methods as invisible ink, concealing a message in an innocent text, or in a fake covering cryptogram, or other methods in which the existence of the message is concealed from the enemy; (2) privacy systems, for example speech inver-

A solid orange circle is centered on the page. Inside the circle, the words "Cryptographic" and "Security" are written in black, stacked vertically.

Cryptographic Security

A diagram consisting of two overlapping circles. The left circle is orange and contains the text 'Cryptographic Security'. The right circle is light green and contains the text 'Computational'. The circles overlap in the center. Below the green circle is an orange rounded rectangle containing the text 'Implies Practical Impossibility'.

Cryptographic
Security

Computational

Implies Practical Impossibility

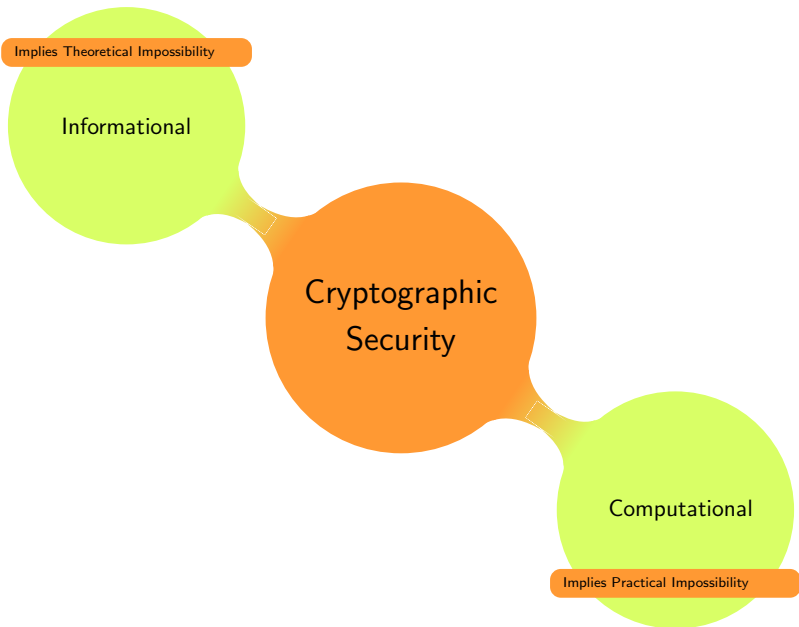
Implies Theoretical Impossibility

Informational

Cryptographic
Security


Computational

Implies Practical Impossibility



- ▶ Cryptographic Security definitions **differ** from Computer Security definitions
- ▶ Cry. Sec. is quantifiable
- ▶ Often possible to calculate the amount of effort required to break a cryptographic algorithm.
- ▶ The goal of cryptographic security is to make well-defined problems impossible to solve.

How to quantify crypto security in ways that are both theoretically sound and practically relevant?

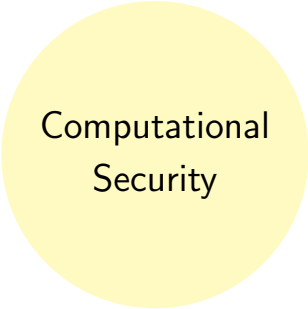
- ▶ Cipher's Security $\xleftarrow{\text{Impossibility}}$ Attacker's Capability
- ▶ What is "Impossibility" in this context? 
- ▶ Two notions of impossibility in crypto

Informational Security

- ▶ Theoretical Impossibility
- ▶ Does not quantify security : Cipher is either secure or insecure

Computational Security

- ▶ Practical Impossibility
- ▶ Quantifiable measure of the strength of a cipher



Computational Security



Security in Practice

- ▶ “Hardness” of breaking a cipher
- ▶ In/with **reasonable** time/resources

Comp. Sec. typically expressed in terms of two values

- ▶ t , which is a limit on the number of operations that an attacker will carry out
- ▶ ϵ , which is a limit on the probability of success of an attack

Idea 

Quantifying security of an adversary running with certain time by bounding the **maximum success probability**

- ▶ “Hardness” of breaking a cipher
- ▶ In/with **reasonable** time/resources

Comp. Sec. typically expressed in terms of two values

- ▶ t , which is a limit on the number of operations that an attacker will carry out
- ▶ ϵ , which is a limit on the probability of success of an attack

Idea 

Quantifying security of an adversary running with certain time by bounding the **maximum success probability**

- ▶ “Hardness” of breaking a cipher
- ▶ In/with **reasonable** time/resources

Comp. Sec. typically expressed in terms of two values

- ▶ t , which is a limit on the number of operations that an attacker will carry out
- ▶ ϵ , which is a limit on the probability of success of an attack

Idea 

Quantifying security of an adversary running with certain time by bounding the **maximum success probability**

- ▶ “Hardness” of breaking a cipher
- ▶ In/with **reasonable** time/resources

Comp. Sec. typically expressed in terms of two values

- ▶ t , which is a limit on the number of operations that an attacker will carry out
- ▶ ϵ , which is a limit on the probability of success of an attack

Idea 

Quantifying security of an adversary running with certain time by bounding the **maximum success probability**

- ▶ “Hardness” of breaking a cipher
- ▶ In/with **reasonable** time/resources

Comp. Sec. typically expressed in terms of two values

- ▶ t , which is a limit on the number of operations that an attacker will carry out
- ▶ ϵ , which is a limit on the probability of success of an attack

Idea 

Quantifying security of an adversary running with certain time by bounding the **maximum success probability**

“A crypto-scheme is $(t, \epsilon) - \text{secure}$ if **every** adversary performing at most t operations succeeds in breaking the scheme with probability at most ϵ ”

- ▶ Note: t and ϵ are just limits
- ▶ Implying no attacker performing fewer than t operations will succeed (with probability ϵ).
- ▶ So what does this **not** imply?

We say t is a **lower bound** on the computational effort needed.

- ▶ When can we say that this bound is **tight**? 

“A crypto-scheme is $(t, \epsilon) - \text{secure}$ if **every** adversary performing at most t operations succeeds in breaking the scheme with probability at most ϵ ”

- ▶ Note: t and ϵ are just limits
- ▶ Implying no attacker performing fewer than t operations will succeed (with probability ϵ).
- ▶ So what does this **not** imply?


We say t is a **lower bound** on the computational effort needed.

- ▶ When can we say that this bound is **tight**? 

“A crypto-scheme is $(t, \epsilon) - \text{secure}$ if **every** adversary performing at most t operations succeeds in breaking the scheme with probability at most ϵ ”

- ▶ Note: t and ϵ are just limits
- ▶ Implying no attacker performing fewer than t operations will succeed (with probability ϵ).
- ▶ So what does this **not** imply?

We say t is a **lower bound** on the computational effort needed.

- ▶ When can we say that this bound is **tight**? 

Symmetric Cipher with 128-bit Key

Ideally

Cipher should be $(t, t/2^{128})$ – *secure* for any value of t between 1 and 2^{128}

Success prob. of three possible attacks

- ▶ $t = 1$, an attacker tries one key, $\epsilon = \frac{1}{2^{128}}$
- ▶ $t = 2^{128}$, an attacker tries all 2^{128} keys, $\epsilon = 1$ (Brute-force)
- ▶ $t = 2^{64}$, an attacker tries only $t = 2^{64}$ keys and succeeds with a probability of

$$\epsilon = \frac{2^{64}}{2^{128}} = 2^{-64}$$

Designer aim: Brute-force should be the best attack

Symmetric Cipher with 128-bit Key

Ideally

Cipher should be $(t, t/2^{128})$ – *secure* for any value of t between 1 and 2^{128}

Success prob. of three possible attacks

- ▶ $t = 1$, an attacker tries one key, $\epsilon = \frac{1}{2^{128}}$
- ▶ $t = 2^{128}$, an attacker tries all 2^{128} keys, $\epsilon = 1$ (Brute-force)
- ▶ $t = 2^{64}$, an attacker tries only $t = 2^{64}$ keys and succeeds with a probability of

$$\epsilon = \frac{2^{64}}{2^{128}} = 2^{-64}$$

Designer aim: Brute-force should be the best attack

Symmetric Cipher with 128-bit Key

Ideally

Cipher should be $(t, t/2^{128})$ – *secure* for any value of t between 1 and 2^{128}

Success prob. of three possible attacks

- ▶ $t = 1$, an attacker tries one key, $\epsilon = \frac{1}{2^{128}}$
- ▶ $t = 2^{128}$, an attacker tries all 2^{128} keys, $\epsilon = 1$ (Brute-force)
- ▶ $t = 2^{64}$, an attacker tries only $t = 2^{64}$ keys and succeeds with a probability of

$$\epsilon = \frac{2^{64}}{2^{128}} = 2^{-64}$$

Designer aim: Brute-force should be the best attack

Symmetric Cipher with 128-bit Key

Ideally

Cipher should be $(t, t/2^{128})$ – *secure* for any value of t between 1 and 2^{128}

Success prob. of three possible attacks

- ▶ $t = 1$, an attacker tries one key, $\epsilon = \frac{1}{2^{128}}$
- ▶ $t = 2^{128}$, an attacker tries all 2^{128} keys, $\epsilon = 1$ (Brute-force)
- ▶ $t = 2^{64}$, an attacker tries only $t = 2^{64}$ keys and succeeds with a probability of

$$\epsilon = \frac{2^{64}}{2^{128}} = 2^{-64}$$

Designer aim: Brute-force should be the best attack

Achieving Computational Security

Idea

Proving that breaking your crypto scheme is at least as hard as solving another problem known to be hard.

Reduction

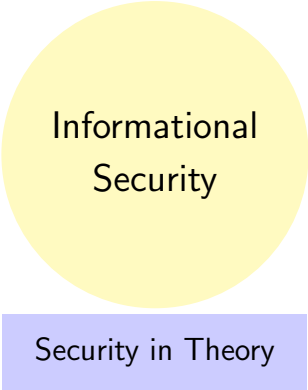
We say that breaking some cipher is reducible to problem X if any method to solve problem X also yields a method to break the cipher.

- ▶ Two directions
 - ▶ Proofs Relative to a Mathematical Problem
 - ▶ Usual approach for public-key ciphers
 - ▶ e.g. Integer-factorization
 - ▶ Proofs Relative to Another Crypto Problem
 - ▶ Usual approach for symmetric ciphers

- ▶ Not Provably-secure but Probably-secure
- ▶ “Believed” to be secure
- ▶ Due to absence of attacks even after extensive cryptanalysis
- ▶ Notion of security margin
- ▶ Round-reduced/Simplified version cryptanalysis

Example

Advanced Encryption Standard (AES)



Informational
Security

Security in Theory

- ▶ Independent of the attacker's power
- ▶ Notion of computationally unbounded adversary




A cipher is informationally secure only if, even given unlimited computation time and memory, it cannot be broken.

- ▶ Eve can obtain no information about the plaintext by obtaining the ciphertext.

Definition


A cryptosystem has perfect secrecy if $\Pr[x|y] = \Pr[x]$ for all $x \in \mathcal{P}, y \in \mathcal{C}$.

- ▶ Implies: The **a posteriori** probability that the plaintext is x , given that the ciphertext is y is observed, is identical to the **a priori** probability that the plaintext is x
- ▶ Does the Shift-Cipher provide perfect secrecy? 
- ▶ If all 26 keys are used with equal probability
- ▶ That is, a new random key is used to encrypt every plaintext character

- ▶ Eve can obtain no information about the plaintext by obtaining the ciphertext.

Definition


A cryptosystem has perfect secrecy if $\mathbf{Pr}[x|y] = \mathbf{Pr}[x]$ for all $x \in \mathcal{P}, y \in \mathcal{C}$.

- ▶ Implies: The **a posteriori** probability that the plaintext is x , given that the ciphertext is y is observed, is identical to the **a priori** probability that the plaintext is x
- ▶ Does the Shift-Cipher provide perfect secrecy? 
- ▶ If all 26 keys are used with equal probability
- ▶ That is, a new random key is used to encrypt every plaintext character

- ▶ Eve can obtain no information about the plaintext by obtaining the ciphertext.

Definition


A cryptosystem has perfect secrecy if $\mathbf{Pr}[x|y] = \mathbf{Pr}[x]$ for all $x \in \mathcal{P}, y \in \mathcal{C}$.

- ▶ Implies: The **a posteriori** probability that the plaintext is x , given that the ciphertext is y is observed, is identical to the **a priori** probability that the plaintext is x
- ▶ Does the Shift-Cipher provide perfect secrecy? 
- ▶ If all 26 keys are used with equal probability
- ▶ That is, a new random key is used to encrypt every plaintext character

- ▶ Eve can obtain no information about the plaintext by obtaining the ciphertext.


Definition

A cryptosystem has perfect secrecy if $\mathbf{Pr}[x|y] = \mathbf{Pr}[x]$ for all $x \in \mathcal{P}, y \in \mathcal{C}$.

- ▶ Implies: The **a posteriori** probability that the plaintext is x , given that the ciphertext is y is observed, is identical to the **a priori** probability that the plaintext is x
- ▶ Does the Shift-Cipher provide perfect secrecy? 
- ▶ If all 26 keys are used with equal probability
- ▶ That is, a new random key is used to encrypt every plaintext character

Theorem

Suppose $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is a cryptosystem where $|\mathcal{K}| = |\mathcal{P}| = |\mathcal{C}|$, then the cryptosystem provides perfect secrecy if and only if every key is used with equal probability $1/|\mathcal{K}|$, and for every $x \in \mathcal{P}$ and $y \in \mathcal{C}$ there is a unique K such that $e_K(x) = y$.

- ▶ Proof: HW-Problem 
- ▶ Note: $|\mathcal{K}| = |\mathcal{P}|$
- ▶ What is the practical implication?

Let $n \geq 1$ be an integer, and take $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$. For, $K \in (\mathbb{Z}_2)^n$, $e_K(x)$ is the xor of x and K .

If $x = (x_1, \dots, x_n)$ and $K = (K_1, \dots, K_n)$, then

$$e_K(x) = (x_1 \oplus K_1, \dots, x_n \oplus K_n)$$

If $y = (y_1, \dots, y_n)$, then decryption

$$d_K(y) = (y_1 \oplus K_1, \dots, y_n \oplus K_n)$$

The cipher is informationally secure because given a ciphertext and unlimited time to try all possible keys, K , and compute the corresponding plaintext, x , you would still be unable to identify the right K because there are as many possible x s as there are K s. 