E0 235: Cryptography

Naman Mishra

August 2024

Contents

The course

Course webpage

MS Teams:

Lecture 1.
Tuesday
August 06

Timeline

Symmetric/Private/Shared key cryptography will be covered by Prof. Arpita Patra till 18th September, and then resume near the end of November. **Asymmetric/Public key cryptography** will be covered by Prof. Sanjit Chatterjee in the middle.

Evaluation

Both professors may have different evaluation techniques, each with a 50% weightage.

Prof. Arpita Patra

TBD

Prof. Sanjit Chatterjee

- Midterm:
- Two assignments:
- Endterm:

Chapter 1

Introduction

1.1 History

The 1980s mark a transition from classical to modern cryptography. Classical cryptography dealt only with secure communication, with ad-hoc codes and ciphers. When a code is broken, it is fixed, or another one is created. Creative, intellectually challenging, but not scientifically rigorous.

Modern cryptographic problems:

- Authenticated message transmission: An adversary tries to tamper with the message. The receiver should be able to detect tampering.
- Electronic voting and auctions: Return the highest bid without revealing the other bids. In elections, make the votes anonymous and secret till the end.
- Activism with safety: Deniable encryption
- Secure storage, secret sharing, broadcast, zero-knowledge proofs
- Secure information retrieval
- Secure outsourcing to the cloud
- **Secure computation:** The holy grail of cryptography. An abstraction of everything else.

1.2 Secure (multiparty) computation

Year	Technique
110 BC	Caesar Cipher
WWII	Enigma
1980s	Boom!

Table 1.1: Timeline of cryptography

Problem 1.1. There are n parties, $P_1, P_2, \ldots, P_n$, who do not trust each other. Each party P_i has a private input x_i to a common n-input function f.

The goal is to compute $f(x_1, x_2, \ldots, x_n)$ while revealing nothing about the inputs except the output.

Applications:

- Satellite collision avoidance
- Secure set intersection
- Privacy-preserving everything

Modern cryptography follows three principles:

- A *formal definition* of security capturing requirement.
- *Precise* and *well-studied* assumptions.
- Mathematical proof of security.

Examples.

- Prime factorization
- Subset sum: Given a set of integers S and a target t, is there a subset of S that sums to t?

Our primary objective will be secure communication: turning insecure channels into secure ones. We expect (i) privacy, (ii) integrity and (iii) authenticity.

Integrity is the ability to detect tampering. Authenticity is the ability to detect impersonation.

1.3 Classical examples

In the private key setting, a message m is encrypted with a key k to get a ciphertext c. The ciphertext is decrypted with the same key k to get the message m. We will use the following notation:

- The key-generation algorithm Gen is a randomized algorithm that outputs a key k according to some probability distribution.
- The encryption algorithm Enc_k is parameterized by the key k, and takes a message m to output a ciphertext c. This may be deterministic or randomized.
- The decryption algorithm Dec_k is parameterized by the key k, and takes a ciphertext c to output a message m.
- The key space $\mathcal{K}$ is the set of all possible keys.
- The message space $\mathcal{M}$ is the set of all possible messages.
- The cipher-text space $\mathcal{C}$ is the set of all possible ciphertexts.

Example (Caesar cipher). The Caesar cipher has $\mathcal{M} = \mathcal{C} = \{0, 1, \ldots, 25\}$. $\text{Enc}(m) = m + 3 \bmod 26$ and $\text{Dec}(c) = c - 3 \bmod 26$. It is a keyless cipher.

A keyed cipher needs to be private.

> The security of a cryptographic system should not depend on the secrecy of the algorithm, but only on the secrecy of the key. – Kerckhoffs, 1883

Here are some arguments for Kerckhoffs' principle:

- Maintaining the secrecy of the algorithm is far more difficult than maintaining the secrecy of the key.
- Replacing the algorithm is infinitely harder than replacing the key.
- For cryptography to be an everyday tool, secret algorithms would need to be devised for every pair of communicating parties.

Example (Shift cipher). The obvious generalization of the Caesar cipher using a key k.

26 is an embarrassing size for a key space.

Example (Mono-alphabetic substitution). The key is a permutation of the alphabet. $\text{Enc}_k = k$ and $\text{Dec}_k = k^{-1}$.

The key space has size 26!. Brute force is infesible, but *frequency analysis* destroys it.

Example (Vigenère cipher). The key is a random t-tuple of shifts. The message is divided into blocks of length t, and the key is applied to each block using the shift cipher.

The key space has size 26^t, but if t is known, the key can be broken by frequency analysis. Consider all the characters at positions $i \pmod t$. These are all encrypted with the same key, so frequency analysis works. Repeat to get the entire word.

Even if we take the key to be a tuple of t permutations, the same method works.

Lecture 2.
Thursday
August 08

Learnings from classical SKE

- Algorithms of secure key encryption (SKE), and in all of cryptography, must be public.
- The key space must be huge.
- Definitions and proofs.

Today, we will formulate a formal definition (the threat and break model) and

Chapter 2

Secret Key Encryption

Definition 2.1. A *secret key encryption scheme* is a tuple of algorithms $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ together with three sets $\mathcal{K}, \mathcal{M}, \mathcal{C}$ called the *key space*, *message space* and *ciphertext space* where:

- $\mathrm{Gen}\colon 1 \to \mathcal{K}$ is a probabilistic algorithm;
- $\mathrm{Enc}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and $\mathrm{Dec}\colon \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ are deterministic algorithms

such that

$$\forall m \in \mathcal{M}(\mathrm{Dec}_k(\mathrm{Enc}_k(m))) = m.$$

We write Enc_k and Dec_k to denote Enc and Dec partially applied to the key k.

2.1 A formal definition of security

We consider the computationally omnipotent adversary $\mathcal{R}$ (for Ravana). Suppose they have a capability to listen to all ciphertexts. This is called *ciphertext only attack* (COA).

We will also assume that $\mathcal{R}$ can use randomness. Encryption schemes often use randomness, so why not the adversary?

Thus, the adversary has the following characteristics:

- Randomized
- All-powerful
- Ciphertext only

Let us attempt to define "security". Here are some possibilities:

	a	b	c	d
0				
1				
2				

Table 2.1: Example encryption scheme

- A scheme is secure if it does not leak the secret key. Nope, $m \mapsto m$ would be considered secure.
- A scheme is secure if it does not leak the *entire* message. Nope, revealing even a single bit is bad.
- A scheme is secure if it does not leak *any* bit. Nope. It could be that the scheme reveals the parity of the message modulo 3.
- A scheme is secure if it does not leak *any* digit in *any* base. Hmm, interesting.
- A scheme is secure if it does not leak *any* reasonable information about the message. How do we formalize this?

2.1.1 Probability review

Theorem 2.2 (Law of total probability)**.** *Let $E_1, E_2, \ldots, E_n$ be a partition of the sample space. Then for any event A,*

$$\Pr(A) = \sum_{i=1}^{n} \Pr(A \mid E_i) \Pr(E_i).$$

Theorem 2.3 (Bayes' theorem)**.** *Let A and B be events with $\Pr(B) > 0$. Then*

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \Pr(A)}{\Pr(B)}.$$

2.2 Formalizing SKE

Exercise 2.4. *Consider the encryption scheme in table 2.1 with distributions*

$$K \sim \begin{pmatrix} 0 & 1 & 2 \\ 1/2 & 1/4 & 1/4 \end{pmatrix} \qquad M \sim \begin{pmatrix} a & b & c & d \\ & & & \end{pmatrix}$$

Definition 2.5 (Perfect encryption)**.**
An encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is *perfectly secure* if for every random variable M over $\mathcal{M}$ and every $m \in \mathcal{M}$, $c \in \mathcal{C}$,

$$\Pr[M = m \mid C = c] = \Pr[M = m].$$

Remark. This is probabily the first formal definition of security, by Claude E. Shannon in Bell Systems Technical Journal, 28(4): 656–715, 1949.

Theorem 2.6 (Vernam cipher)**.** *Let* $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$*. Let* Gen *draw uniformly at random from* $\mathcal{K}$*, and let* $\mathrm{Enc} = \mathrm{Dec} = \oplus$ *(bitwise XOR).*

Then $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *is perfectly secure.*

Proof. It is easy to see that $\mathrm{Dec}_k \circ \mathrm{Enc}_k = \mathrm{id}$.

Now for any $c \in \mathcal{C}, m \in \mathcal{M}$, we have

$$\Pr[C = c \mid M = m] = \Pr[K = c \oplus m] = 2^{-n}.$$

Thus

$$\Pr[C = c] = \sum_{m \in \mathcal{M}} \Pr[C = c \mid M = m] \Pr[M = m] = 2^{-n}.$$

This gives $\Pr[C = c \mid M = m] = \Pr[C = c]$ and so C and M are independent. ■

Problems with the Vernam cipher:

- Can we reuse the key for multiple messages? Nope. We can XOR two consecutive messages to get the XOR of the two messages.