Introduction to Secure Communication

DSC 4110/6111: Modern Cryptography and AI/ML Security

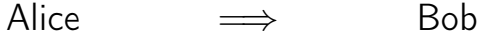
Dr. Laltu Sardar

School of Data Science,
Indian Institute of Science Education and Research Thiruvananthapuram (IISER TVM)

July 27 , 2026

Introduction to Secure Communication

Secure Communication: The Basic Scenario



- Alice wants to send a message to Bob.
- The communication channel is public.
- Anyone may observe the transmitted data.
- Goal: Bob should receive the intended message correctly.

Sending a Message over a Public Channel

Alice

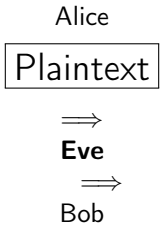
Plaintext: "Meet at 5 PM"

$\Longrightarrow$ Public Network $\Longrightarrow$

Bob

- The message is transmitted in plaintext.
- Every intermediate node can read the message.
- No confidentiality is provided.

An Adversary Appears



- Eve can eavesdrop on the communication.
- She learns the complete message.
- Privacy is completely lost.

Encrypt Before Transmission

$$\boxed{\text{Ciphertext} = \text{Enc}_K(\text{Plaintext})}$$

Alice

$$\boxed{\text{Ciphertext}}$$

$\Longrightarrow$

Eve

$\Longrightarrow$

Bob

$$\boxed{\text{Plaintext} = \text{Dec}_K(\text{Ciphertext})}$$

What Should Security Achieve?

$$\text{Alice} \Longrightarrow \text{Ciphertext} \Longrightarrow \text{Bob}$$

Even if Eve intercepts the ciphertext,

> Ciphertext reveals no useful information about the plaintext

A secure communication system should provide:

- Confidentiality
- Integrity
- Authentication
- Availability (system level)

Fundamental Security Goals

1. **Confidentiality**
   - Only authorized parties should be able to access the information.
2. **Integrity**
   - Data should not be altered without detection during storage or transmission.
3. **Authentication**
   - Verify the identity of communicating parties and the origin of the data.

Availability

- Information and services should remain accessible to authorized users when needed.

Achieving Integrity

Integrity

Confidentiality hides the message.

Integrity ensures that the message has **not been modified**.

Examples of integrity violations:

- Changing the amount in a bank transaction.
- Modifying a software update.
- Altering a medical record.

How Do We Achieve Integrity?

A cryptographic checksum is attached to every message.

$$m \xrightarrow{\text{Integrity Algorithm}} t$$

Sender transmits

$$(m, t)$$

Receiver recomputes the tag.

$$t \stackrel{?}{=} t'$$

If they match, the message is accepted; otherwise, it is rejected.

Techniques for Achieving Integrity

Technique	**Purpose**
Hash Function	Detect accidental modifications
Message Authentication Code (MAC)	Detect malicious modifications
Digital Signature	Integrity + authentication + non-repudiation

These techniques will be studied in later lectures.

Achieving Authenticity

Authentication

Authentication answers the question:

Who created this message?

Authentication assures that the sender is genuine.

Examples:

- Online banking
- Software updates
- Secure email
- Digital certificates

How Do We Achieve Authentication?

Authentication requires cryptographic evidence that only the legitimate sender can generate.

$$m \xrightarrow{\text{Authentication Algorithm}} a$$

Sender transmits

$$(m, a)$$

Receiver verifies

$$\text{Verify}(m, a)$$

Accepted only if the proof is valid.

Techniques for Achieving Authentication

Technique	**Used In**
Message Authentication Code (MAC)	Shared-secret systems
Digital Signature	Public-key systems
Digital Certificates	Identity verification

A MAC provides authentication only between parties sharing the same secret key, whereas digital signatures provide public verifiability.

Integrity vs Authentication

Integrity	Has the message been modified?
Authentication	Who generated the message?

Hash Function

$\Longrightarrow$

MAC

$\Longrightarrow$

Digital Signature

These primitives together provide secure and trustworthy communication.

Achieving Confidentiality

Achieving Confidentiality

Goal: Prevent unauthorized parties from learning the message.

$$\text{Plaintext} \xrightarrow{\text{Enc}_K} \text{Ciphertext} \xrightarrow{\text{Public Channel}} \text{Ciphertext} \xrightarrow{\text{Dec}_K} \text{Plaintext}$$

- Encryption transforms readable data into an unintelligible form.
- Only authorized users can recover the original message.

Encryption and Decryption

Let

$$m = \text{message}, \qquad c = \text{ciphertext}, \qquad k = \text{secret key}.$$

Encryption:

$$c = \mathrm{Enc}_k(m)$$

Decryption:

$$m = \mathrm{Dec}_k(c)$$

Correctness requires

$$\boxed{\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m.}$$

Examples Encryption Schemes (Symmetric-Key)

Scheme	Type	Typical Key Size
DES	Block Cipher	56 bits
3DES	Block Cipher	112/168 bits
AES	Block Cipher	128/192/256 bits
ChaCha20	Stream Cipher	256 bits
Salsa20	Stream Cipher	256 bits

- AES is the current standard for symmetric encryption.
- DES is no longer considered secure due to its short key length.
- ChaCha20 is widely used in software and mobile applications.

Symmetric-Key Encryption

Alice $\Longleftrightarrow$ Bob

Shared Secret Key k

Encryption:

$$c = \mathrm{Enc}_k(m)$$

Decryption:

$$m = \mathrm{Dec}_k(c)$$

- Alice and Bob use the same secret key k.
- The security of the scheme relies on keeping k secret.
- Anyone knowing k can both encrypt and decrypt messages.

The Key Distribution Problem

Symmetric encryption assumes that Alice and Bob already share a secret key.

How can Alice and Bob establish k securely?

Possible solutions:

- Physical exchange of the key.
- Trusted third party.
- Public-key cryptography.

Public-Key Encryption

Each user possesses a pair of keys:

$$(\mathsf{pk}, \mathsf{sk})$$

- Public key (pk): Shared with everyone.
- Secret key (sk): Kept private by the owner.

Encryption:

$$c = \mathrm{Enc}_{\mathsf{pk}}(m)$$

Decryption:

$$m = \mathrm{Dec}_{\mathsf{sk}}(c)$$

Symmetric vs Public-Key Encryption

	Symmetric	**Public-Key**
Encryption Key	Secret key k	Public key pk
Decryption Key	Secret key k	Secret key sk
Speed	Fast	Slower
Key Distribution	Difficult	Easier
Typical Use	Data Encryption	Key Exchange

- Modern secure communication protocols combine both approaches.

Roadmap

So far, we have seen *what* encryption achieves. Next, we study *how* encryption schemes are constructed mathematically.

The next part of the course will cover:

- Mathematical model of an encryption scheme.
- Security definitions for encryption.
- Construction of symmetric-key encryption schemes.
- Key distribution: How two parties establish a shared secret key.

Encryption Scheme

A symmetric-key encryption scheme consists of three algorithms:

$$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$$

Key generation:
The key generation algorithm outputs a secret key.

$$k \leftarrow \mathsf{Gen}(1^\lambda)$$

- λ is the security parameter.
- k is known only to legitimate users.

Encryption:

$$c \leftarrow \mathsf{Enc}_k(m),\ m \in \mathcal{M}:\ \text{message space}$$

Decryption:

$$m \leftarrow \mathsf{Dec}_k(c),\ c \in \mathcal{C}:\ \text{ciphertext space}$$

IISER THIRUVANANTHAPURAM

Correctness

An encryption scheme must satisfy

$$\boxed{\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m}$$

for every

$$k \leftarrow \mathsf{Gen}(1^\lambda), \qquad m \in \mathcal{M}.$$

Example Construction of Symmetric key Encryption Scheme.

Evolution of Classical Encryption

$$\text{Caesar Cipher} \Longrightarrow \text{Monoalphabetic Cipher} \Longrightarrow \text{Vigenère Cipher} \Longrightarrow \text{One-Time Pad}$$

Each scheme attempts to overcome a weakness of its predecessor:

- Caesar: Very small key space.
- Monoalphabetic: Vulnerable to frequency analysis.
- Vigenère: Hides frequencies using multiple alphabets.
- One-Time Pad: Achieves perfect secrecy.

Caesar Cipher

Each letter is shifted by a fixed amount k.

$$c = (m + k) \bmod 26$$

$$m = (c - k) \bmod 26$$

Example ($k = 3$):

$$\texttt{HELLO} \rightarrow \texttt{KHOOR}$$

Problem:

- Only 25 possible keys.
- Easily broken by brute force.

Monoalphabetic Substitution Cipher

Each plaintext letter is replaced by a fixed permutation.

$$c = \pi(m), \qquad m = \pi^{-1}(c)$$

where π is a permutation of the alphabet.

Example:

$$A \mapsto Q,\ B \mapsto X,\ C \mapsto M, \ldots$$

Improvement:
Much larger key space.

Problem:

- Letter frequencies are preserved.
- Broken using frequency analysis.

Key Space of the Monoalphabetic Substitution Cipher

Each plaintext letter is mapped to a unique ciphertext letter.

The secret key is a permutation of the alphabet.

$$\begin{array}{cccccccc} A & B & C & D & E & \cdots & Y & Z \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ Q & L & M & X & P & \cdots & C & H \end{array}$$

Number of possible keys:

$$26! \approx 4.03 \times 10^{26} \approx 2^{88}$$

- Far larger than the Caesar cipher (25 keys).
- Exhaustive search is computationally infeasible.

Why Does Frequency Analysis Still Work?

Although the substitution changes the letters, it does *not* change their frequencies.

Plaintext Letter	E	T	A	O	N	...
Frequency	High	High	High	High	High	
Ciphertext Letter	X	Q	M	J	K	...
Frequency	High	High	High	High	High	

An attacker compares ciphertext letter frequencies with known language statistics to recover the substitution.

Large key space alone does not guarantee security.

Language Statistics Differ Across Languages

Letter frequencies are characteristic of a language and are remarkably stable over large texts.

Rank	1	2	3	4	5
English	E	T	A	O	I
French	E	A	S	I	T

Common words also reveal the language.

English	the, and, of, to, in
French	le, de, et, la, les

Frequency analysis exploits language statistics, not just the cipher.

How to Fix it?

Vigenère Cipher

Use multiple Caesar ciphers controlled by a keyword.

For keyword $k_1 k_2 \cdots k_t$,

$$c_i = (m_i + k_i) \bmod 26.$$

Example:

ATTACKATDAWN

Keyword:

LEMONLEMONLE

Improvement:
Letter frequencies are spread across multiple alphabets.

Problem:

- Repeated key introduces patterns.
- Broken using Kasiski and Friedman attacks.

How Does the Vigenère Cipher Improve Security?

The Vigenère cipher uses multiple Caesar ciphers instead of a single substitution.

$$\text{Ciphertext}_i = (m_i + k_i) \bmod 26$$

where the key is repeated periodically.

Position	1	2	3	4	5	6	7	8
Key	L	E	M	O	N	L	E	M

Consequences:

- The same plaintext letter may encrypt to different ciphertext letters.
- Letter frequencies are dispersed.
- Simple frequency analysis no longer works.

Key Space of the Vigenère Cipher

Suppose the key has length ℓ.

Each key position can be any of the 26 letters.

$$|\mathcal{K}| = 26^{\ell}$$

Key Length	Number of Keys
14	$26^{14} \approx 2^{64}$
28	$26^{28} \approx 2^{128}$
55	$26^{55} \approx 2^{256}$

A longer key significantly increases the key space.

Why Was the Vigenère Cipher Eventually Broken?

Although frequency analysis is hidden,

the key repeats periodically.

Repeated key

$$\underbrace{LEMON}_{\text{key}}\ \underbrace{LEMON}_{\text{repeats}}\ \underbrace{LEMON}_{\text{repeats}}$$

creates repeated encryption patterns in long messages.

Once the key length is discovered,

- split the ciphertext into ℓ groups,
- each group becomes a Caesar cipher,
- apply frequency analysis to each group separately.

Kasiski and Friedman Attacks

This slide can be skipped

Two classical cryptanalytic techniques estimate the key length.

Kasiski Examination

- Find repeated ciphertext fragments.
- Measure distances between repetitions.
- Compute the greatest common divisor (GCD) of these distances.
- The GCD often reveals the key length.

Friedman Test

- Measure the *Index of Coincidence (IC)* of the ciphertext.
- Estimate the key length statistically.

After estimating the key length, frequency analysis breaks each Caesar cipher independently.

IISER THIRUVANANTHAPURAM

One-Time Pad

Replace the repeating keyword with a truly random key.

- Message: $m \in \{0,1\}^{\ell}$
- Key: $k \xleftarrow{\$} \{0,1\}^{\ell}$, where $|k| = |m|$
- Encryption:

$$c = m \oplus k$$

- Decryption:

$$m = c \oplus k.$$

Advantages

- Perfect secrecy.
- Information-theoretically secure.

Problem

- Key must be random.
- Key length equals message length.
- Key can never be reused.

References

Textbook:

1. Douglas R. Stinson and Maura Paterson. *Cryptography: Theory and Practice*. 4th Edition. Chapman & Hall/CRC Press, 2019. ISBN: 9780367148591.

Dr. Laltu Sardar, Assistant Professor, IISER Thiruvananthapuram
laltu.sardar@iisertvm.ac.in, laltu.sardar.crypto@gmail.com
Course webpage: https://laltu-sardar.github.io/courses/26_crypto.html.