

Introduction to Cryptographic Primitives

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 29 , 2026

Cryptographic Primitives: An Introduction

Recap from the Previous Lecture

One-Time Pad (OTP)

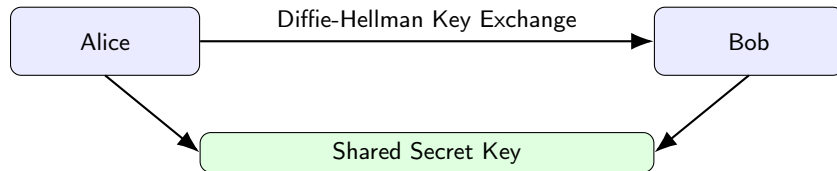
OTP provides **information-theoretic security** only if

- Sender and receiver already share a secret key.
- Key length equals message length.
- Key is perfectly random.
- Key is never reused.

Challenge

How can two users securely establish such a long shared secret over an insecure network?

Establishing the Shared Secret



- Public-key cryptography (e.g., Diffie-Hellman) establishes a shared secret.
- This secret can then be used for symmetric encryption.
- However, OTP still requires a key as long as the message.

Diffie-Hellman Key Exchange (DHKE)

Public Parameters

Everyone knows

$$G = \langle g \rangle, \quad |G| = q$$

where

- G is a cyclic group,
- g is a generator of G .

Alice

Chooses secret $a \xleftarrow{\$} \mathbb{Z}_q$

Computes $A = g^a$

Sends A

Bob

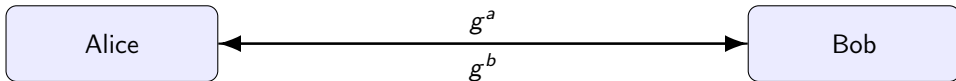
Chooses secret $b \xleftarrow{\$} \mathbb{Z}_q$

Computes $B = g^b$

Sends B

Only g^a and g^b travel through the network.

Why Both Parties Obtain the Same Secret?



Alice computes

$$K = (g^b)^a = g^{ab}$$

Bob computes

$$K = (g^a)^b = g^{ab}$$

$$(g^b)^a = g^{ab} = g^{ba} = (g^a)^b$$

Key Idea

Alice and Bob independently obtain the same shared secret

$$K = g^{ab}$$

without ever transmitting a , b , or g^{ab} .

Motivation for Pseudorandom Generators

The Challenge

Diffie-Hellman Key Exchange (DHKE) establishes only a short shared secret,

$$K = g^{ab}.$$

Suppose Alice and Bob want to encrypt a 10 GB file using a One-Time Pad. They would need a secret key of length 10 GB.

How can we efficiently expand the short shared secret g^{ab} into a long cryptographic keystream?

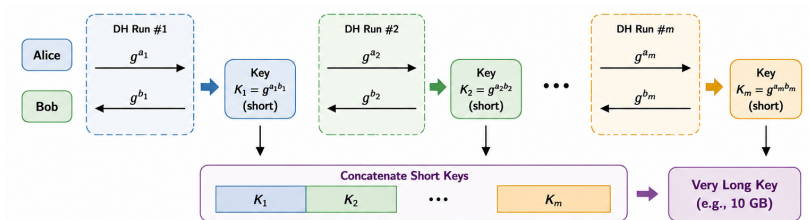
Short Secret $\implies$ Very Long Secure Keystream ?

Can We Generate Very Long Keys?

Suppose the message is several gigabytes long.

Option 1: Repeat Diffie-Hellman

- Generate many short keys.
- Concatenate them together.
- Repeat until the required key length is reached.

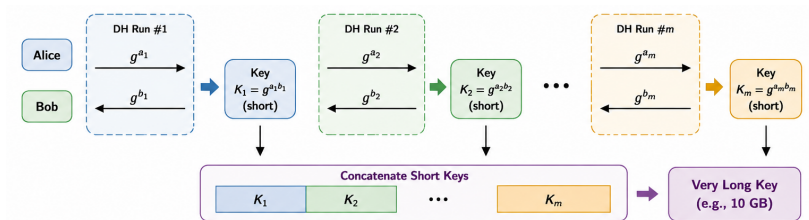


Can We Generate Very Long Keys?

Suppose the message is several gigabytes long.

Option 1: Repeat Diffie-Hellman

- Generate many short keys.
- Concatenate them together.
- Repeat until the required key length is reached.

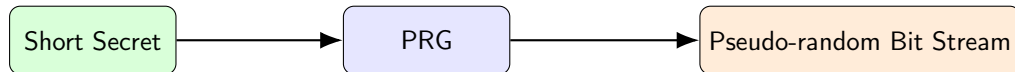


Problem

Exponentiation in large cyclic groups is computationally expensive.

Generating millions of key bits this way is significantly slower than the encryption itself.

A Better Idea: Expand One Short Secret



Instead of generating a long key directly,

Short Secret $\longrightarrow$ PRG $\longrightarrow$ Long Pseudo-random Key

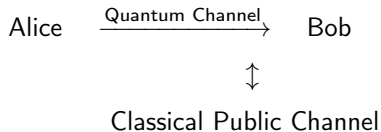
Advantages

- Only one expensive key establishment.
- Encryption becomes extremely efficient.
- Standard approach in almost every modern secure protocol.

Establishing Short Secret

Quantum Key Distribution (QKD)

Unlike classical key establishment, QKD derives its security from the laws of quantum mechanics.



- Uses quantum states (typically single photons) to establish a shared secret key.
- Any eavesdropping attempt introduces detectable disturbances.
- The generated key is subsequently used with symmetric encryption (e.g., AES or One-Time Pad).

QKD: Advantages and Limitations

Advantages

- Information-theoretic security.
- Detects eavesdropping.
- Secure against quantum computers.

Limitations

- Requires specialized quantum hardware.
- Limited communication distance and key generation rate.
- Does not encrypt data; it only establishes secret keys.

QKD is a key establishment protocol, not an encryption scheme.

If QKD were to become efficient and scalable, it would solve the key agreement problem. However, it has not yet reached that stage.

What Changes in the Quantum Era?

Problem 1

- 1 Classical Diffie-Hellman relies on the **Discrete Logarithm Problem (DLP)**.
- 2 Large-scale quantum computers can solve DLP efficiently using Shor's algorithm.
- 3 QKD is not yet practical for widespread deployment.

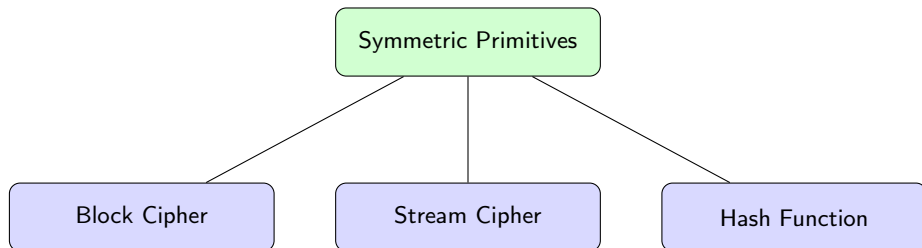
Problem 2

Many theoretical constructions of PRGs and PRFs are also derived from classical hardness assumptions such as

- Discrete Logarithm
- Integer Factorization

These constructions are therefore not suitable in the post-quantum setting.

Modern Cryptography: What Remains Reliable?

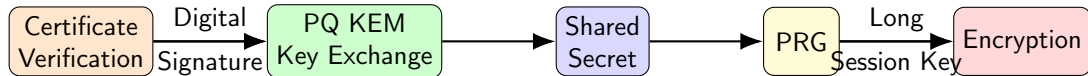


We also have modern post-quantum public-key primitives

- Digital Signatures
- Key Encapsulation Mechanisms (KEMs)

These form the foundation of today's secure communication systems.

How Secure Communication Works Today



- Public keys are authenticated using digital signatures.
- PQ KEM establishes one short shared secret.
- Symmetric primitives expand it into a long keystream.
- Encryption then proceeds efficiently.

Two Different Philosophies

Classical Constructions

Security relies on

- Discrete logarithm
- Integer factorization
- Number-theoretic assumptions

These are essentially

Mathematical Hard Problems

Modern Symmetric Designs

Security relies on

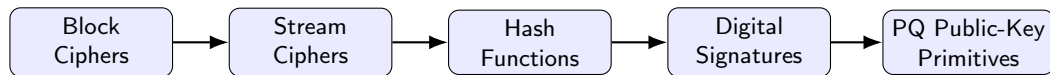
- Careful engineering
- Years of public cryptanalysis
- Resistance against known attacks

These are essentially

Engineering Hard Problems

Symmetric primitives remain the workhorse of practical cryptography because of their exceptional efficiency and long history of cryptanalytic evaluation.

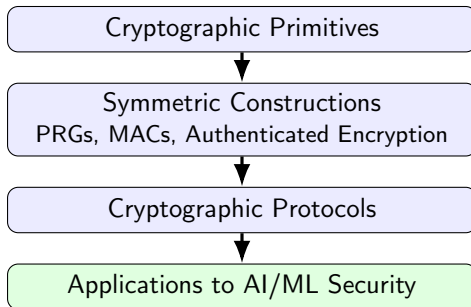
Roadmap for the Next Few Lectures



Goal

Study the fundamental cryptographic primitives, understand why they are secure, and learn how they are combined to build authenticated encryption, modern cryptographic protocols, and secure AI/ML systems.

Course Roadmap



Learning Journey

We first study the fundamental cryptographic primitives, then learn how they are composed into symmetric constructions such as authenticated encryption. These constructions form the basis of modern cryptographic protocols, which ultimately enable secure and privacy-preserving AI/ML systems.

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.