# Cryptography

## MAT354 - Cryptography Course

**Instructor:** Adil Akhmetov
**University:** SDU
**Semester:** Fall 2025

Press Space for next page →

# Course Description

The **"Cryptography for Programmers"** course is designed for students with computer science backgrounds and focuses on practical implementation of cryptographic algorithms in modern software systems.

## Key Focus Areas

- **Practical Programming** of cryptographic algorithms
- **Real-world Applications** in software development
- **Minimal Mathematics** - more implementation details
- **Modern Tools** and libraries

## Learning Outcomes

- Implement classical and modern ciphers
- Work with cryptographic libraries
- Create secure APIs and web applications
- Understand modern cryptographic protocols

# Learning Objectives

## Technical Skills

1. **Practical Programming of Cryptographic Algorithms**
   - Implementation of classical and modern ciphers
   - Working with cryptographic libraries
   - Creating secure APIs and web applications
2. **Understanding Modern Cryptographic Protocols**
   - TLS/SSL, HTTPS, SSH
   - Authentication and authorization
   - Digital signatures and certificates

## Security Analysis

3. **Security Analysis and Testing**
   - Penetration testing of cryptographic implementations
   - Vulnerability analysis
   - Secure programming practices

# Course Structure

# Weekly Course Plan

## Weeks 1-4: Foundations

- **Week 1:** Introduction to Cryptography for Programmers
- **Week 2:** Classical Ciphers and Implementation
- **Week 3:** Cryptanalysis and Attacks
- **Week 4:** Stream Ciphers and One-Time Pads

## Weeks 5-8: Core Algorithms

- **Week 5:** Block Ciphers and Applications
- **Week 6:** Hash Functions and Data Integrity
- **Week 7:** Asymmetric Cryptography - RSA
- **Week 8:** Key Exchange and Protocols

## Weeks 9-12: Applications

- **Week 9:** Digital Signatures
- **Week 10:** Cryptographic Protocols in Web Development
- **Week 11:** Cryptography in Mobile Applications
- **Week 12:** Cryptography in Blockchain

## Weeks 13-15: Advanced Topics

- **Week 13:** Quantum Cryptography and Post-Quantum
- **Week 14:** Practical Projects
- **Week 15:** Final Testing and Project Defense

# Week 1: Introduction to Cryptography

## Topics

- Cryptography fundamentals: terminology and concepts
- Types of attacks and security models
- Cryptographic primitives in programming

## Practical Assignments

- Create a simple cipher in Python/JavaScript
- Analyze vulnerabilities in existing code
- Set up development environment for cryptography

## Code Example

```python
# Simple XOR cipher
def xor_cipher(text, key):
    return ''.join(chr(ord(c) ^ key) for c in text)

# Usage
message = "Hello World"
encrypted = xor_cipher(message, 5)
print(f"Encrypted: {encrypted}")
```

# Week 2: Classical Ciphers

## Topics

- Caesar cipher and its variations
- Substitution ciphers
- Transposition ciphers

## Practical Assignments

- Implement Caesar cipher with different keys
- Create frequency analysis program
- Break simple ciphers using brute force

## Code Example

```
def caesar_cipher(text, shift):
    result = ""
    for char in text:
        if char.isalpha():
            ascii_offset = 65 if char.isupper() else 97
            result += chr((ord(char) - ascii_offset + shift
        else:
            result += char
    return result
```

# Assessment Methods

## Quizzes (60%)

- **3 Quizzes** throughout the semester
- Cover practical programming and theory
- Hands-on coding challenges
- Real-world problem solving

## Final Project (40%)

- **Comprehensive cryptographic project**
- Choose from suggested topics or propose your own
- Implement a complete cryptographic system
- Documentation and presentation required

**Suggested Final Projects:** - Secure messaging application - File encryption system - Blockchain implementation - Web authentication system

# Technical Requirements

## Software

- **Python 3.8+** with libraries:
  - `cryptography`
  - `pycryptodome`
  - `requests`
- **Node.js** for web development
- **Git** for version control
- **Docker** for containerization

## Recommended IDEs

- **Visual Studio Code** with cryptography extensions
- **PyCharm Professional**
- **IntelliJ IDEA**

## Getting Started

```
pip install cryptography pycryptodome requests
npm install -g @slidev/cli
git clone [course-repository]
```

# Reading List

## Primary Literature

1. **"Real-World Cryptography"** - David Wong (2021)
2. **"Cryptography Engineering"** - Niels Ferguson, Bruce Schneier, Tadayoshi Kohno (2010)
3. **"Serious Cryptography"** - Jean-Philippe Aumasson (2017)

## Online Resources

- OWASP Cryptographic Storage Cheat Sheet
- Cryptopals Crypto Challenges
- NIST Cryptographic Standards
- CryptoHack

# Practice Tools

## Network Analysis

**Wireshark** - Network traffic analysis - Protocol inspection - Security monitoring

## Web Testing

**Burp Suite** - Web application testing - Vulnerability scanning - Security assessment

## Password Security

**John the Ripper** - Password cracking - Security testing - Vulnerability assessment

# Contact Information

## Instructor

**Adil Akhmetov**
📧 adil.akhmetov@sdu.edu.kz
🏢 [Office Address]
🕒 [Office Hours]

## Course Information

- **Course Code:** MAT354
- **Course Name:** Cryptography
- **Semester:** Fall 2025
- **University:** SDU

**Important:** This syllabus is adapted for students with computer science backgrounds and focuses on practical application of cryptography in modern software development.

# Questions?

Thank you for your attention! ❤️