Secure Communications (DFCS H3013) - Course Summary

Course Overview

Module: Secure Communications (DFCS H3013)
Instructor: Dr. Jin Xu (jin.xu@tudublin.ie)
Assessment: 20% Labs, 40% Project, 40% In-Class Test

Week 1: Introduction to Cryptography & Classic Ciphers

Key Concepts

- **Four Core Goals of Cryptography (CIAN)**
  - Confidentiality: Keep information secret
  - Integrity: Ensure data hasn't been altered
  - Authentication: Verify sender/receiver identity
  - Non-repudiation: Sender cannot deny sending

Classic Ciphers

- **Substitution Ciphers**: Replace letters (e.g., Caesar cipher - shift by 3)
- **Transposition Ciphers**: Rearrange characters (e.g., Rail Fence)
- **Polyalphabetic Ciphers**: Multiple substitution alphabets (e.g., Vigenère)

Key Principle

Kerckhoffs's Principle: Strength should depend only on key secrecy, not algorithm secrecy

---

Week 2: Encoding, Hashing & Passwords

Encoding vs Encryption

- **Encoding**: Data representation (Base64, ASCII, Hex) - always reversible, NOT encryption
- **Encryption**: Requires a key to decrypt

Hashing

- **One-way function**: Easy to compute, impossible to reverse
- **Properties**: Fixed output length, collision-resistant, avalanche effect

- **Common Algorithms**:
  - MD5 (128-bit) - INSECURE, legacy only
  - SHA-1 (160-bit) - BROKEN (SHAttered attack 2017)
  - SHA-2 (224-512 bit) - SECURE, current standard
  - SHA-3 (224-512 bit) - Secure, limited adoption

Password Security

- **Never store plaintext passwords**
- **Salting**: Add random data before hashing (prevents rainbow table attacks)
- **Key Stretching**: Apply hash function repeatedly to slow brute force
- **KDF (Key Derivation Function)**: Hash + Salt + Key Stretching

Attack Methods

- Dictionary attacks: Common words/passwords
- Brute force: Try all combinations
- Rainbow tables: Precomputed hash databases

Week 3: Keys, Symmetric & Asymmetric Encryption

Key Concepts

- **Keyspace**: Total possible keys (2^n for n-bit key)
- **Birthday Paradox**: Collisions occur at √N, not N
- **Random Key Generation**: Requires high complexity and unpredictability

Symmetric Encryption

- **Same key for encryption and decryption**
- **Fast and efficient**
- **Types**:
  - Block ciphers: DES, 3DES, AES (encrypt fixed blocks)
  - Stream ciphers: RC4, SEAL (encrypt bit/byte at a time)

Key Algorithms

- **DES**: 56-bit key - INSECURE (brute-forceable)

- **3DES**: Applies DES 3 times - slower but stronger
- **AES**: 128/192/256-bit keys - current standard, very secure

Asymmetric Encryption

- **Different keys**: Public (encrypt) and Private (decrypt)
- **Slower but solves key distribution problem**
- **Examples**: RSA, Diffie-Hellman, ECC

Week 4: Key Exchange & Diffie-Hellman

The Key Exchange Problem

- **One-Time Pad**: Perfect secrecy but impractical (key distribution)
- **Trusted Third Party**: Easier but requires full trust
- **Pairwise Keys**: $n(n-1)/2$ keys needed - doesn't scale

Mathematical Foundation

- **Modular Arithmetic**: Like a clock (wrapping)
- **Primitive Root**: Generates all residues in mod p
- **Discrete Logarithm Problem**: Hard to reverse $g^x \bmod p$

Diffie-Hellman Key Exchange

Process:

1. Agree on public p (prime) and g (primitive root)
2. Alice picks private a, computes $A = g^a \bmod p$
3. Bob picks private b, computes $B = g^b \bmod p$
4. Exchange A and B publicly
5. Both compute shared secret: $K = B^a \bmod p = A^b \bmod p$

Weakness: Vulnerable to Man-in-the-Middle attacks (no authentication)

Week 5: RSA, ECC & Digital Signatures

RSA (Rivest-Shamir-Adleman)

Security: Based on difficulty of factoring large primes

Key Generation:

1. Choose primes p and q
2. Compute $n = p \times q$
3. Compute $\varphi(n) = (p-1)(q-1)$
4. Choose e (public exponent)
5. Compute d (private exponent)
6. Public key: (n, e), Private key: (n, d)

Encryption/Decryption:

- Encrypt: $C = M^e \bmod n$
- Decrypt: $M = C^d \bmod n$

Elliptic Curve Cryptography (ECC)

- **Same security with smaller keys**: 256-bit ECC ≈ 3072-bit RSA
- **Faster, more efficient** (ideal for mobile/IoT)
- **Based on**: Elliptic Curve Discrete Logarithm Problem (ECDLP)
- **Used in**: TLS 1.3, Signal, Bitcoin

Digital Signatures

Purpose: Authentication + Integrity + Non-repudiation

Process:

1. Hash the message
2. Encrypt hash with sender's private key (= signature)
3. Receiver decrypts with sender's public key
4. Compare hashes to verify

Hash vs MAC vs Digital Signature:

- Hash: Integrity only (anyone can create)

- MAC: Integrity + Authentication (shared secret)
- Digital Signature: Integrity + Authentication + Non-repudiation (public/private keys)

Week 6: PKI, Certificates & SSL/TLS

Digital Certificates

- **Bind public key to identity**
- **Contain**: Public key, owner info, validity dates, CA signature
- **Format**: X.509 v3 standard

Certificate Authority (CA)

Validation Types:

- **DV (Domain Validation)**: Confirms domain ownership (lowest trust)
- **OV (Organization Validation)**: Verifies business registration
- **EV (Extended Validation)**: Deep legal/operational verification (highest trust)

PKI Trust Models

1. **Single-Root**: One CA (simple but risky)
2. **Cross-Certified**: Peer-to-peer CAs (flexible, complex)
3. **Hierarchical**: Root → Intermediate → End-entity (most common)

Certificate Formats

- **.PEM**: Base64, text format (-----BEGIN CERTIFICATE-----)
- **.DER**: Binary format
- **.CRT/.CER**: Certificate files
- **.PFX/.P12**: Certificate + private key bundle
- **.CSR**: Certificate Signing Request

SSL/TLS

- **SSL**: Developed by Netscape (1990s) - NOW DEPRECATED
  - Vulnerable to POODLE, Heartbleed
- **TLS**: Modern successor
  - TLS 1.2 (2008) - widely used

- TLS 1.3 (2018) - current standard

TLS Handshake:

1. Client Hello (versions, cipher suites, random)
2. Server Hello (chosen cipher, certificate, random)
3. Key Exchange (derive shared secret)
4. Change Cipher Spec (switch to encryption)
5. Encrypted communication begins

Week 8: Cryptanalysis & Attacks

Types of Attacks

By Motivation:

- Criminal: Fraud, scams, destruction
- Publicity: Fame/recognition
- Legal: Exploit legal loopholes

By Security Property:

- Interception: Confidentiality (eavesdropping)
- Fabrication: Authentication (fake data)
- Modification: Integrity (alter data)
- Interruption: Availability (DoS)

Passive vs Active Attacks

- **Passive**: Observe/copy data (eavesdropping, traffic analysis) - hard to detect
- **Active**: Modify/disrupt data (modification, impersonation, DoS) - easier to detect

Cryptanalysis Methods

- **Brute Force**: Try all possible keys
- **Frequency Analysis**: Examine letter/symbol patterns
- **Known-Plaintext**: Have plaintext-ciphertext pairs
- **Chosen-Plaintext**: Choose what to encrypt

- **Differential/Linear**: Mathematical analysis of patterns
- **Side-Channel**: Exploit timing, power consumption

Algorithm-Specific Vulnerabilities

- **DES**: 56-bit key too small, brute-forceable
- **AES**: Still secure when properly implemented
- **RSA**: Vulnerable if key size < 2048 bits, quantum threat (Shor's algorithm)
- **DSA**: Weak if random number reused
- **Diffie-Hellman**: Man-in-the-Middle, small subgroup attacks

---

Week 9: Wireless Security

802.11 Standards

- **802.11a**: 54 Mbps, 5 GHz
- **802.11b**: 11 Mbps, 2.4 GHz
- **802.11g**: 54 Mbps, 2.4 GHz
- **802.11n (Wi-Fi 4)**: 600 Mbps, 2.4/5 GHz
- **802.11ac (Wi-Fi 5)**: 1.3 Gbps, 5 GHz
- **802.11ax (Wi-Fi 6)**: 10-12 Gbps, 2.4/5 GHz

WEP (Wired Equivalent Privacy)

BROKEN - NEVER USE

- Uses RC4 stream cipher with 40/104-bit key + 24-bit IV
- **Weaknesses**:
  - IV too short (24 bits) - repeats in ~5 hours
  - No message integrity protection
  - Vulnerable to bit-flipping attacks
  - Weak key generation

WPA (Wi-Fi Protected Access)

- **Transitional fix for WEP**

- Uses RC4 with TKIP (Temporal Key Integrity Protocol)
- Dynamic key generation (every 10,000 packets)
- Better integrity checking (MIC)
- Still based on RC4 (outdated)

WPA2 (Current Standard)

- **Full IEEE 802.11i implementation**
- Uses **AES-CCMP** (not RC4)
  - AES: Strong encryption
  - Counter Mode: Unique IV per packet
  - CCMP: Authentication + integrity
- **Two Modes**:
  - WPA2-Personal (PSK): Shared password
  - WPA2-Enterprise (802.1X): RADIUS authentication

802.1X Enterprise Authentication

Components:

- **Supplicant**: Client device
- **Authenticator**: Access point
- **Authentication Server**: RADIUS/TACACS+

EAP Methods:

- **EAP-TLS**: Mutual certificates (most secure)
- **EAP-TTLS**: Server cert + encrypted tunnel for credentials
- **PEAP**: Protected EAP, encrypted tunnel
- **LEAP**: Cisco proprietary (weak, deprecated)

Week 10: PGP & SSL/TLS Deep Dive

PGP (Pretty Good Privacy)

Hybrid Cryptosystem: Combines symmetric + asymmetric encryption

How it Works:

1. Generate random symmetric key
2. Encrypt message with symmetric key (fast)
3. Encrypt symmetric key with recipient's public key (secure)
4. Send both encrypted message and encrypted key

With Digital Signature:

1. Hash the message
2. Encrypt hash with sender's private key (signature)
3. Encrypt message + signature with symmetric key
4. Encrypt symmetric key with recipient's public key
5. Receiver verifies signature and decrypts message

Implementation: GnuPG (GPG) - open-source PGP

SSL/TLS Handshake (Detailed)

Prerequisites: TCP 3-way handshake first (SYN → SYN/ACK → ACK)

TLS Handshake Steps:

1. **Client Hello**: TLS versions, cipher suites, client random
2. **Server Hello**: Chosen TLS version, cipher suite, server random, certificate
3. **Server Key Exchange**: Public key parameters + digital signature
4. **Client Key Exchange**: Client's ECDHE public key
5. **Both compute shared secret** using Diffie-Hellman
6. **Change Cipher Spec**: "Switch to encryption"
7. **Encrypted Handshake (Finished)**: Verify handshake integrity

Common Cipher Suite Example: ECDHE-RSA-AES128-GCM-SHA256

- ECDHE: Key exchange (forward secrecy)
- RSA: Authentication
- AES-128-GCM: Symmetric encryption
- SHA-256: Hashing/signatures

Key Security Principles

Defense in Depth

- Layer multiple security mechanisms
- No single point of failure
- Combine encryption, authentication, integrity checks

Current Best Practices

- **Symmetric**: AES-256
- **Asymmetric**: RSA-2048+ or ECC-256+
- **Hashing**: SHA-256 or SHA-3
- **Passwords**: Salted + KDF (bcrypt, PBKDF2)
- **Wi-Fi**: WPA2-Enterprise or WPA3
- **Web**: TLS 1.2+ (prefer TLS 1.3)
- **Email**: PGP/GPG or S/MIME

Deprecated/Insecure (DO NOT USE)

- ❌ DES, RC4
- ❌ MD5, SHA-1
- ❌ SSL (any version)
- ❌ WEP, WPA (TKIP)
- ❌ RSA < 2048 bits
- ❌ TLS 1.0, TLS 1.1

Future Threats

- **Quantum Computing**: Threatens RSA, ECC, Diffie-Hellman
- **Solution**: Post-Quantum Cryptography (PQC) - Industry beginning integration (2025)

Important Acronyms

- **AES**: Advanced Encryption Standard

- **CA**: Certificate Authority
- **DES**: Data Encryption Standard
- **DH**: Diffie-Hellman
- **EAP**: Extensible Authentication Protocol
- **ECC**: Elliptic Curve Cryptography
- **HMAC**: Hash-based Message Authentication Code
- **IV**: Initialization Vector
- **KDF**: Key Derivation Function
- **MAC**: Message Authentication Code
- **MITM**: Man-in-the-Middle
- **PGP**: Pretty Good Privacy
- **PKI**: Public Key Infrastructure
- **PSK**: Pre-Shared Key
- **RADIUS**: Remote Authentication Dial-In User Service
- **RSA**: Rivest-Shamir-Adleman
- **TLS**: Transport Layer Security
- **WEP**: Wired Equivalent Privacy
- **WPA**: Wi-Fi Protected Access

Exam Preparation Tips

1. **Understand key differences**: WEP vs WPA vs WPA2, SSL vs TLS, symmetric vs asymmetric
2. **Know attack types**: Passive/Active, known-plaintext, chosen-plaintext, MITM
3. **Practice calculations**: Diffie-Hellman, RSA encryption/decryption, modular arithmetic
4. **Memorize key sizes**: DES (56), AES (128/192/256), RSA (2048+), ECC (256+)
5. **Understand certificate chains**: Root CA → Intermediate CA → End-entity
6. **Know when to use what**: Digital signatures vs encryption, hashing vs MAC
7. **Security timelines**: What's deprecated (WEP, SSL, MD5) vs current (WPA2, TLS 1.3, SHA-256)