CS2107 Introduction to Information Security
AY24/25, Y2S1
Notes

Sim Ray En Ryan

December 3, 2025

Contents

1 Introduction **4**
- 1.1 Assessment . . . 4
- 1.2 Security Fundamentals . . . 4
  - 1.2.1 CIA Triad . . . 4
  - 1.2.2 Trade-off in Security . . . 4
  - 1.2.3 Adversarial Thinking . . . 4

2 Encryption **4**
- 2.1 Symmetric Key Encryption . . . 4
  - 2.1.1 Characters in Cryptography . . . 5
  - 2.1.2 Attack Models . . . 5
  - 2.1.3 Attacker Capabilities . . . 5
- 2.2 Classical Ciphers . . . 5
  - 2.2.1 Substitution Cipher . . . 5
  - 2.2.2 Permutation/Transposition Cipher . . . 6
  - 2.2.3 One Time Pad (OTP) . . . 6
- 2.3 Modern Ciphers (Block Ciphers) . . . 6
  - 2.3.1 Triple DES . . . 6
  - 2.3.2 Block Cipher Modes of Operation . . . 6
  - 2.3.3 Stream Ciphers . . . 6
  - 2.3.4 Cryptography Pitfalls and Principles . . . 7

3 Authentication Credential **7**
- 3.1 Authentication and Credential . . . 7
- 3.2 Password Attacks . . . 7
  - 3.2.1 Cracking Attacks . . . 7
  - 3.2.2 Stealing Attacks . . . 7
  - 3.2.3 Password Strength and Protection . . . 8
- 3.3 Other Authentication Methods . . . 8
  - 3.3.1 ATM Skimming . . . 8
  - 3.3.2 Biometrics . . . 8
  - 3.3.3 Multi-Factor Authentication (MFA) . . . 8

4 Authenticity (Data Origin) **8**
- 4.1 Public Key Cryptography (PKC) . . . 8
- 4.2 Hashing Algorithms . . . 8
- 4.3 Authenticity Mechanisms . . . 9

5 Channel Security (Authentication Protocol + TLS) **9**

5.1 Public Key Infrastructure (PKI) 9
5.2 Authentication Protocols 9
5.3 TLS (Transport Layer Security) 9

6 Network Security **10**
6.1 MITM and DoS Attacks 10
6.2 Network Defense 10

7 Access Control **10**
7.1 Access Control Concepts 10
7.2 Access Control Models for Information Flow 10
7.3 Access Control Structures and UNIX 10

8 Secure Programming **11**
8.1 Vulnerabilities 11
8.2 Defense 11

9 Web Security **11**

1 Introduction

1.1 Assessment

- 2 CTF (20 + 1%)
- 4 ICQ (10 + 1%)
- Mid Terms (15 + 1%)
- Tutorial Attendance (7%)
- Case Study (3%)
- Finals (45%)

1.2 Security Fundamentals

1.2.1 CIA Triad

- **Confidentiality**
- **Integrity**
- **Availability**

1.2.2 Trade-off in Security

Security is often traded off against:

- Ease-of-use
- Performance
- Cost

1.2.3 Adversarial Thinking

- A **threat** is blocked by a **control** of a **vulnerability**.
- **Common Vulnerabilities and Exposures (CVE)**.
- **Zero Day Vulnerabilities**.

2 Encryption

2.1 Symmetric Key Encryption

- Process: Plaintext $x \rightarrow$ Encryption $E_k() \rightarrow$ Ciphertext $c \rightarrow$ Decryption $D_k() \rightarrow$ Plaintext x.

- **Correctness property**: $D_k(E_k(x)) = x$.
- **Secure**: Difficult to derive useful information of key k and plaintext x.

2.1.1 Characters in Cryptography

- **Alice**: Originator
- **Bob**: Recipient
- **Eve**: Eavesdropper
- **Mallory**: Malicious Actor

2.1.2 Attack Models

The goal is to design the strongest system to prevent an attacker's weakest goal.

- **Total Break**: Find the private key.
- **Partial Break**: Decrypt a ciphertext (without the secret key) or determine the type of plaintext.
- **Indistinguishability**: Attacker is able to distinguish different ciphertexts.

2.1.3 Attacker Capabilities

- **Ciphertext only attack**: Attacker has many ciphertexts c, may discover properties of plaintext.
- **Known plaintext attack**: Attacker is given plaintext x and the corresponding ciphertext c.
- **Chosen plaintext attack (CPA2)**: Attacker can choose plaintext x, place into a black box (encryption oracle), and receive corresponding ciphertext c.
- **Chosen ciphertext attack (CCA2)**: Attacker chooses ciphertext c, places into a black box (decryption oracle), and receives plaintext x.

2.2 Classical Ciphers

2.2.1 Substitution Cipher

- **Key**: A substitution table S.
- **Key Space Size**: For the alphabet is 26!.
- **Attacks**: Totally broken under **known plaintext attack**. Not secure under **ciphertext-only attack** due to frequency analysis.

2.2.2 Permutation/Transposition Cipher

- Splits plaintext into blocks of t characters and permutes characters in blocks.
- Not secure on its own, but interlacing (like in AES) makes it secure.

2.2.3 One Time Pad (OTP)

- XORs n-bit plaintext/ciphertext and n-bit key. Key size must equal plaintext size.
- **Attacks**: Unbreakable, secure against exhaustive search.

2.3 Modern Ciphers (Block Ciphers)

Modern ciphers are designed such that a successful attack does not perform better than an exhaustive search.

- **Data Encryption Standard (DES)**: Key length 56 bits. Exhaustive search needs 2^{56} loops.
- **Advanced Encryption Standard (AES)**: Block length 128 bits, key length 128, 192, or 256 bits. No known attacks on AES itself.

2.3.1 Triple DES

- Two forms: $E_{k1}(E_{k2}(E_{k1}(x)))$ or $E_{k1}(D_{k2}(E_{k1}(x)))$. Both require 2^{112} operations.
- Vulnerable to **Meet in the Middle** attack (Known Plaintext Attack) in time and space $O(2^{k+1})$.

2.3.2 Block Cipher Modes of Operation

- **Electronic Code Book (ECB)**: Applies cipher to each block using the same key K. **Vulnerability**: Leaks information as identical plaintext blocks are encrypted to the same ciphertext.
- **Cipher Block Chaining (CBC)**: Has an **Initialization Vector (IV)** y_0. $y_i = E_k(x_i \oplus y_{i-1})$. **Limitation**: Slow due to lack of parallelism.
- **CounTeR Mode (CTR)**: Uses a **Nonce** and counter. Encrypts the counter value, then XORs the message. **Advantage**: Parallelizable.

2.3.3 Stream Ciphers

- Generates a cryptographically secure pseudorandom sequence using a short secret key and uses it as a "one time pad".
- **IVs** must be different to prevent revealing the XOR of two plaintexts.

2.3.4 Cryptography Pitfalls and Principles

- **IV Choice**: IV must be unpredictable.
- **Key Reuse**: OTP cannot be reused.
- **Randomness**: Use secure random sources.
- **Padding Oracle Attack**: An attack on modes like CBC that relies on the padding being correct.
- **Kerckhoff's Principle**: The system should be secure even if everything except the secret key is public knowledge.

3 Authentication Credential

3.1 Authentication and Credential

- **Authentication**: Origin of information is confirmed (implies Integrity).
- **Credential**: Information bound to owner.

3.2 Password Attacks

3.2.1 Cracking Attacks

- **Online Attack**: Interacts with the authentication system, limited by countermeasures (login delays, lockouts).
- **Offline Attack**: Attacker obtains a password hash and tests passwords offline.
- **Dictionary Attack**: Tests common passwords.
- **Guessing Attack**: Infers password from user's social information.

3.2.2 Stealing Attacks

- **Shoulder Surfing**.
- **Sniffing**: Networks, wireless keyboards.
- **Side-Channel Attack**: E.g., Keyboard Sounds.
- **Keyloggers**: Hardware or Software.
- **Phishing / Spearphishing**.
- **Stolen Password Files / Databases**.

3.2.3 Password Strength and Protection

- **True Random Password**: High **Entropy**.
- **Security Requirement**: Online ($> 2^9$ Bits), Offline (> 128 bits).
- **Protection**: Use **KDF / Cryptographic Hashes** to store $d = \text{Hash}(P)$. Apply **Salting**.

3.3 Other Authentication Methods

3.3.1 ATM Skimming

- Authentication by card (magnetic strip) and PIN.
- **Skimmer**: Captures magnetic strip data. PIN captured by camera or spoofed keypad.
- **Defense**: Anti Skimmer Devices, Keypad Shields, Unforgeable smartcards.

3.3.2 Biometrics

- Uses a stored **template** for comparison.
- **Errors**: False Match Rate (**FMR**), False Non-Match Rate (**FNMR**).
- **Attacks**: Spoofed/Fake fingerprints, defended by "liveness" detection.

3.3.3 Multi-Factor Authentication (MFA)

- Requires ≥ 2 **different factors**.
- **Factors**: Something you **know** (Passwords), **have** (Tokens), **are** (Biometrics).

4 Authenticity (Data Origin)

4.1 Public Key Cryptography (PKC)

- Uses different keys (Asymmetric). Slower but better for key management.
- **Textbook RSA**: Public key $\langle n, e \rangle$, Private key d. Security based on difficulty of **factorizing** n.

4.2 Hashing Algorithms

- Produce an n-bit Digest.
- **Properties**: **Collision Resistant**, **One Way**.

- **Birthday Attack**: Exploits the probability of finding a collision with a short digest.

4.3 Authenticity Mechanisms

- **Unkeyed Hash**: Checks integrity of downloaded file. Vulnerable to **2nd Pre-Image attack**.
- **Symmetric Keyed Hash (MAC)**: **HMAC** or **CBC-MAC**. Generates a tag t using a shared secret key K. Provides authenticity (integrity), not confidentiality.
- **Asymmetric Key (Digital Signature)**: Generates $s = \text{sign}_{k_{\text{private}}}(\text{hash}(F))$ using the sender's private key. Provides **Non-repudiation**.

5 Channel Security (Authentication Protocol + TLS)

5.1 Public Key Infrastructure (PKI)

- **CA (Certificate Authority)**: Issues and signs digital **Certificates**. Root CAs are pre-loaded in OS/browsers.
- **Certificate**: Binds a public key to an identity, with a time window of validity.
- **Chain of Trust**: Certificate verified up to a trusted Root CA.
- **Revocation**: Handled via **CRL** (Certificate Revocation List) or **OCSP** (Online Certificate Status Protocol).

5.2 Authentication Protocols

- **Challenge Response**: Uses a **nonce** (random number) for freshness to prevent replay attacks.
- **Authenticated Key Exchange (AKE)**: PKC-based or **Diffie Hellman** (Station To Station, $k = g^{ab} \bmod p$) to establish a shared session key.

5.3 TLS (Transport Layer Security)

- Uses unilateral authenticated key exchange to generate a session key.
- **Handshake**: Client Hello $\rightarrow$ Server Hello (sends cert) $\rightarrow$ Client Key Exchange (sends encrypted secret key) $\rightarrow$ Finished (exchange messages with secret key).

6 Network Security

6.1 MITM and DoS Attacks

- **MITM**: Can occur at any layer. Examples: **DNS Spoofing** (resolves name to attacker's IP), **ARP Poisoning** (malicious node claims to be another MAC address).
- **DoS (Availability)**: Flooding (HTTP, DNS requests). **DDoS** (Distributed DoS) uses a **BotNet**.
- **Reflection Attack**: Request spoofed from victim's address, response sent to victim.
- **Amplification Attack**: Small request yields a large response.

6.2 Network Defense

- **Securing Channel**: SSL/TLS, IPSec, WPA2.
- **VPN**: Tunnels data through a lower layer.
- **Principles**: **PLP** (Least Privilege), **Compartmentalization**, **Defense in Depth**.
- **Firewall**: Filters traffic (Ingress/Egress Filtering). Uses **DMZ** for external services.
- **IDS (Intrusion Detection System)**: Uses attack signature detection or anomaly detection.

7 Access Control

7.1 Access Control Concepts

- **Subjects** (Principals) perform **Operations** (r, w, x) on **Objects**.
- **DAC** (Discretionary, owner-decided) vs **MAC** (Mandatory, system-wide policies).

7.2 Access Control Models for Information Flow

- **Bell LaPadula Model** (Confidentiality): Cannot Read Up, Cannot Write Down.
- **Biba Model** (Integrity): Cannot Write Up, Cannot Read Down.

7.3 Access Control Structures and UNIX

- Structures: **ACL** (Object $\rightarrow$ Subject (Rights)) or **Capabilities** (Subject $\rightarrow$ Object (Rights)).
- **UNIX File System**: Access checked by Owner $\rightarrow$ Group $\rightarrow$ Other.

- **UID 0** (root): All security checks off.
- **Setuid (s bit)**: If set, process effective UID is the file owner's UID (used for **Controlled Invocation**).

8 Secure Programming

8.1 Vulnerabilities

- **Control Flow Integrity**: Vulnerabilities can lead to overwriting the **Return address** on the **Call Stack** (Stack Smashing).
- **Buffer Overflow**: Lack of bound checks in C/C++ (`strcpy`) overwrites memory.
- **Format String Vulnerability** (`printf`): Can lead to confidentiality leaks or integrity violations (`%n`).
- **Integer Overflow**: Input can manipulate integer values to control code flow.
- **Code Injection** (SQL Injection), **Race Condition**.

8.2 Defense

- **Input Validation**, **Filtering**, **Parameterized Queries**.
- **Safe Functions**, **Bound Checks**, **Type Safety**.
- **Canaries** and **Memory Protection**.

9 Web Security

- SSL and TLS, Address Bar Spoofing.
- Cookies, Same Origin Policy.
- XSS (Cross-Site Scripting), XSRF (Cross-Site Request Forgery).