Lecture 1:

Security:
① Regulating assets — information / functionality
② Manage risks
③ Secondary concern

trade off: Security vs functionality

* Security is about imposing countermeasures to reduce risks to assets to acceptable level
* Security policy is a specification of what security goal the countermeasures are intended to achieve. [what & from whom]
* Security mechanism to enforce the policy. [how]
* Security objectives: CIA
① confidentiality (secrecy)
② Integrity
③ Availability
④ Non-repudiation for accountability

* Any discussion of security should start with an inventory of:
① Stakeholders ③ Threats
② Assets ④ Attackers

* How to realise security objectives
① Authentication — prevention
② Authorization — detection
③ Auditing — Reaction
④ Action

* Threats vs Security requirement
information disclosure — confidentiality
tampering with info. — integrity
denial of service — availability
Spoofing — authentication
unauthorized access — authorization

* Cryptography:
The science & art of keeping messages secured [plaintext → ciphertext]
* Cryptanalysis:
The science & art of breaking cipher system
* Strong cipher: Best attack should be brute force key search
* Encryption: message + key → cryptogram
Decryption: cryptogram + key → message
* Asymmetrical (public-key): different keys
* Symmetric (secret-key): same key
* passive attack: content disclosure
* Active attack: content modification

* Monoalphabetic substitution cipher: — total number of possible keys

① Simple shift cipher. key diversity = 25 $(p_i + k) \bmod 26$ or $(c_i - k) \bmod 26$

→ cipher letter is a shifted version of the plain letter with fixed shift value

② Random alphabet substitution cipher key diversity ≈ 26!

* cryptanalysis of simple shift cipher

① known plain text attack

② Try all shift values ⇝ exhaustive key search attack

③ statistical cryptoanalysis

↳ map the highest frequency letter in plain english to the highest frequency in ciphertext

* Polyalphabetic substitution cipher

* One-Time-key cipher [stream]

↳ $c_i = (p_i + k_i) \bmod 26$

- Large possible key diversity
- unbreakable cipher
- True random

* Transposition cipher [block] key diversity = N!

43 1 2 7 6 5

3 4 2 1 7 6 5

* Data Encryption cipher: [DES]

① 64 key bits ⇒ PC-1 → 56 bit ⇝ the last bit in each of 64 bits is not used

(HEX → BINARY bits)

② Divide into L & R ⇒ 28 & 28 [4 blocks = 7 bits]

③ Rotate shift — rotate left / rotate right → 16 key

④ Concatenate L + R [56 bit]

⑤ key ⇒ 64 bits into PC-2 — 16 different keys

⇒ Brute force attack

* play fair cipher

pros:

Simple substitution

cons:

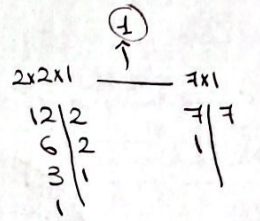
Can be easily cracked

RSA

* Symmetric key encryption

① key generation:

- choose 2 prime p & q ⇒ 3, 7
- compute $n = p * q = 3 \times 7 = 21$
- compute euler $\phi(n) = (p-1)(q-1) = 2 \times 6 = 12$
- choose e $\quad 1 < e < \phi(n)$ & $\gcd(e, \phi(n)) = 1$ ⇒ e=7 for example
  12|7 = ERROR
- key (e, n) = (7, 21)
  ↳ publickey



② Message encryption:

$c = m^e \bmod n \quad \Rightarrow \quad c = 2^7 \bmod 21 = 2$

(e — public)

cipher, plain

$\frac{128}{21} = 6.09$

$128 - 6 \times 21$

③ Message decryption:

$m = c^d \bmod n \quad \Rightarrow \quad e \cdot d = 1 + k \cdot \phi(n)$

$ed \bmod \phi(n) = 1$

$d = e^{-1} \bmod \phi(n)$

$e \neq d$

d → int

{ public → encrypt
{ private → decrypt

* Block cipher → DES [symmetric] — secret key [authentication]
* Stream cipher → RSA [asymmetric] — publickey
  - ① digital signature
  - ② Encryption / decryption
  - ③ key exchange

Lecture 4:

Software Vulnerabilities:

① Cross-Site Scripting (XSS):
→ Enables attackers to inject client-side scripts into web pages viewed by other users.

② OS command injection:
→ Enables attackers to execute OS commands on the server that is running an app

③ Reliance on untrusted inputs. (cookies)
→ An attacker can change inputs using customized clients or other attacks

④ Use of hard-coded credentials [javap -c]

⑤ Missing authentication for critical functions
→ Software doesn't perform any authentication for a functionality that requires a provable user identity

⑥ Missing encryption of sensitive data

Authentication : verify who someone is
Authorization : verify what the user has access to

* OTP: one time password
* hash is the same as mod → 1-way

* Soft delete → UI disappearence
  physical delete → DELETE

Lecture 5

* Secure code review checklist

(1) Design
- Security is layered - each layer assumes other layers may have been compromised
- CIA & AAA principles

(2) Authentication & user management
- Standard security framework are used
- Cookies are not persisted, but encrypted
- Handles suspicious events

(3) Authorization
- Re-authenticate for requests that have side-effects
- Authorization cannot bypassed by cookie manipulation

(4) Session management
- Expire in a reasonably short time
- Avoid excessive cookie use
- Session ID is complex

(5) Input validation
- Data should be checked for special characters

(6) Cryptography
- Restricted areas require Secure Socket Layer [SSL]

(7) Exception handling
- Error messages do not reveal sensitive information
- System errors are never shown to users

(8) Auditing & logging
- unusual activity
- Logs have enough detail.

Lecture 6

* Prevention techniques

1) Encoding: which escapes the user input so that we can interpret it only as data, not as code.

2) Validation: which filters the user input so that we can interpret it as code without malicious commands

* Encoding examples:

- SQLI : ' to ''
- XSS : < to < > to &gt
- use SafeSqlLiteral(_)
- use encodeHTML
- use htmlspecialchars($str) --> php
- use document.write(escape(userInput)) --> JS

* Validation

1) classification strategy:
   - blacklisting or whitelisting
2) validation outcome:
   - rejected or sanitized