Final Security Mechanisms Report

Mobile Platform	Hybrid Application ; IoT System
Application domain type	Smart Air Quality
Authentication	Yes
Authentication schemes	Biometric-based authentication ; Channel-based authentication ; ID-based authentication ; Channel-based authentication
Has DB	Yes
Type of database	SQL (Relational Database)
Which DB	MySQL
Type of information handled	Personal Information ; Confidential Data
Storage Location	Remote Storage (Cloud Database)
User Registration	Yes
Type of Registration	The users will register themselves
Programming Languages	Dart
Input Forms	Yes
Upload Files	No
The system has logs	Yes
The system has regular updates	Yes
The system has third-party	Yes
System Cloud Environments	Public Cloud
Hardware Specification	Yes
HW Authentication	Basic Authentication (user/pass)
HW Wireless Tech	3G ; 4G/LTE ; 5G ; Bluetooth ; Wi-Fi ; Bluetooth ; GPS ; LoRa
Device or Data Center Physical Access	Yes

Security Backup Mechanisms

Security Backup Mechanisms for cloud-based mobile apps are procedures to keep data safe and secure in the event of an emergency, such as a computer crash, a user error, or a malicious attack. These mechanisms can include:

â€¢ Access Control: Access control restricts the access of certain parts of the application, such as confidential data or the applicationâ€™s backend, in order to limit the potential damage caused by malicious activities.

â€¢ Data Encryption: Data Encryption scrambles application data into an unreadable format, making it impossible to access without the decryption key.

â€¢ Password Hashes: Password Hashes are securely stored versions of the usersâ€™ passwords to prevent malicious activities such as credentials theft.

â€¢ Tokenization: Tokenization is a mechanism that replaces sensitive data with a token to reduce the risk of data theft.

â€¢ Backup System: A backup system can be used to store application data in separate, secure locations. This data can be used to restore the application to its former state in the event of a disruption.

Backup Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Backup	iOS	iTunes Backup	Syncs with iTunes for off-site backup	7 - Application
Backup	Android	Google Drive	Google's cloud solution for data storage and backup	7 - Application
Backup	Android	Third-party cloud solutions	Solutions such as Dropbox, OneDrive and iCloud Drive	7 - Application
Backup	All	Local Backup	On-site backups saved on the device's internal storage	1 - Physical
Backup	All	External Storage Backup	Off-site backups saved to external devices such as external hard drives and USB drives	1 - Physical

Security Audit Mechanisms

A Security Audit Mechanism is an automated or manual process which evaluates cloud-based mobile apps for security issues. It may include verifying the integrity of the code, inspecting system configurations, testing user authentication and authorization controls, and ensuring that the system is following best practices such as encryption, patching, and regular system updates. A Security Audit Mechanism can also identify potential security weaknesses and provide recommendations for mitigating these. Furthermore, a Security Audit Mechanism can perform performance and reliability checks, as well as other security checks such as penetration testing, infrastructure testing, and security vulnerability scanning. By utilizing these security audit mechanisms, organizations can ensure their cloud-based mobile apps are safe and secure.

Audit Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Authentication	iOS	Appleâ€™s App-ID and two factor authentication	A two-factor authentication and App-ID system used by Apple to verify and authenticate applications running on its iOS mobile platform	Application
Authorization	iOS	Access control list (ACL)	A tool used to manage user access to various parts of a mobile application, such as data or services	Application
Data Protection	Android	Google Play Store	Googleâ€™s Play Store protects uploaded applications from malicious code before it is distributed on the platform	Presentation
Auditing	iOS	App Store	The App Store provides an audit trail of all applications downloaded, to ensure proper users have the correct permissions to access applications	Application
Data Validation	Android	Android Content Providers	Android content providers are used to securely store data and detect malicious code before it is passed to applications running on the platform	Application

Cryptographic Algorithms Mechanisms

Cryptographic algorithms are used to ensure data confidentiality, authenticity, integrity and non-repudiation in cloud-based mobile apps. To achieve these goals, cryptographic algorithms are often used in combination with mechanisms, such as Digital Signatures, Secret Key Cryptography and Public Key Cryptography.

Digital Signatures validate the identity and authenticity of communications, while Secret Key Cryptography algorithms like AES, DES and 3DES protect transmitted data through the use of encryption. Public Key Cryptography algorithms like RSA, ECDSA and Diffie-Hellman can also be used to authenticate, encrypt and exchange secret keys between the mobile device and the cloud provider. In addition, protocols such as SSL / TLS can add an extra layer of security while protecting and verifying the communication and providing message integrity.

Cryptographic Algorithms Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer	Use for coding	Use for runtime
Integrity	Android	HMAC-SHA256	A cryptographic hash function based on SHA256 that combines a shared secret and the message	7	Yes	Yes
Confidentiality	iOS	AES-128	AES with 128 bit key size that supports authenticated encryption	6	Yes	Yes
Authentication	iOS	ECDSA	Elliptic Curve Digital Signature Algorithm that provides digital signatures	7	Yes	Yes

Biometric authentication mechanisms in cloud-based mobile apps are methods of authentication relying on the physiological characteristics of a user as a method of accessing the device or application. Examples of popular biometric authentication technologies available for cloud-based mobile devices are fingerprint scanning, facial recognition, and voice recognition. These technologies use advanced algorithms to validate a userâ€™s identity based on the physiological traits unique to each individual. By using these methods, companies and app developers can increase the security of their cloud services while preventing unauthorized access.

Biometric Authentication Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer

Authentication & Access Control	Android	Facial Recognition	Hardware based biometric authentication that uses the device front facing camera to snap a picture of the user's face and match it against stored images	Application
Authentication & Access Control	iOS	Voice Recognition	Software based biometric authentication that uses the device microphone and internal software to capture the user's voice and match it against stored audio	Application
Encryption & Decryption	Android	2-Factor Authentication with PIN & Pattern	Combined hardware and software based authentication that requires the user to enter a PIN and draw a pattern on a defined pattern grid.	Presentation
Encryption & Decryption	iOS	Retina Recognition	Hardware based biometric authentication that uses the device front facing camera to obtain a high-resolution picture of the user's eye and matches it against stored images	Application
IDS & IPS	Android	Fingerprint Scan	Hardware based biometric authentication that uses the device built-in fingerprint scanner to scan the user's fingerprint and match it against stored images	Application
IDS & IPS	iOS	3-Factor Authentication with PIN, Pattern & Password	Combined hardware and software-based authentication that requires the user to enter a PIN, draw a pattern on a defined pattern grid, and enter a password	Presentation

Channel-based authentication mechanisms in cloud-based mobile apps refer to a set of security protocols that validate users and authorize access to specific resources in a cloud mobile application. This authentication is done through a set of channels, such as biometrics, passwords, OTPs, or mobile phone numbers, each with its own level of security and authentication request. This type of authentication is used to ensure access to sensitive data and improve the overall security of the application.

Channel-based Authentication Mechanisms Examples:

Security Requirement	Mobile Plataform	Mechanism	Description	OSI Layer
Authentication	Android	HMAC-SHA256	Mobile application uses a pre-shared HMAC-SHA256 token to authenticate with the cloud server and establish a secure channel.	Application
Authorization	iOS	OAuth-2	Mobile application uses an OAuth-2 access token to authorize requests made to the cloud server and establish a secure channel.	Application
Identity Management	Cross-platform	OpenID Connect	Mobile application uses OpenID Connect to authenticate with the cloud server and establish a secure channel.	Application
Data Encryption	Cross-platform	TLS/SSL	Mobile application uses TLS/SSL to encrypt data transmitted between the mobile device and the cloud server.	Transport

ID-based authentication mechanisms are used to authenticate users in cloud-based mobile applications. This type of authentication typically involves the use of an identifier such as an email address or phone number, as well as a password or some other form of proof of identity. ID-based authentication may also involve the use of biometric markers like fingerprints or facial recognition to verify the user's identity. By using ID-based authentication, mobile applications can ensure that only authorized users are granted access, thereby protecting the data stored and exchanged on the application.

ID-based Authentication Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Authentication	iOS	FaceID	User authenticates with their face	Layer 7
Authentication	iOS	Touch ID	User authenticates with their thumbprint	Layer 7
Authorization	iOS	Apple App Tracking Transparency (ATT)	Authorizes a userâ€™s usage data to be tracked by a third-party for targeted advertising	Layer 7
Authentication	Android	Fingerprint Authenticator	User authenticates with their fingerpring	Layer 7
Authentication	Android	Face Unlock	User authenticates with their face	Layer 7
Authorization	Android	Google Play Billing Library	User authorizes payment for in-app billing	Layer 7

Cryptographic Protocols Authentication Mechanisms

Cryptographic protocols mechanisms for cloud-based mobile apps refer to the cryptographic techniques used to protect data and communications between user devices and cloud-services. The protocols involve the encryption of data and messages with symmetric and asymmetric algorithms, the digital signing of messages, the authentication of users, the establishment of secure tunnels, and the use of secure hashing and salting. The goal is to ensure that, if a malicious person attempts to intercept the headers or payload of a cloud-based mobile app, they will be unable to access valuable information.

Cryptographic Protocols Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Authentication	iOS	OAuth	OAuth is an open-standard authorization protocol for allowing access to a protected resource	Application layer
Encryption	Android	TLS	Transport Layer Security (TLS) is a cryptographic protocol used to provide secure communications over a computer network	Transport Layer
Integrity	iOS	SHA-1	Secure Hash Algorithm (SHA-1) is a cryptographic hash function used to generate a 160-bit hash value	Application layer
Non-repudiation	Android	HMAC	HMAC is a cryptographic mechanism used to verify the integrity of a message by using a secret key	Application layer

Access Control Mechanisms

Security Access Control Mechanisms (SACMs) are the technical and administrative strategies and tools used to protect cloud-based mobile apps from unauthorized access to confidential data and systems. These mechanisms are designed to restrict access to certain users, manage user privileges, authenticate user accounts, and authorize access requests. Examples of SACMs include multi-factor authentication (MFA), biometric authentication, single-sign-on (SSO), role-based access control (RBAC), application-level encryption, and least privilege access. SACMs allow organizations to properly control who has access to what resources and strictly enforce principles of confidentiality, privacy, and data security.

Access Control Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Data confidentiality	Android	RSA Encryption	Encryption of data with public and private keys	Application
Data integrity	Android	Hashing	Use of a hash algorithm such as SHA-2 to ensure that data is not tampered with	Transport
Account Management	iOS	Two-Factor Authentication	Use of two-factor authentication to verify user access	Presentation
Data access control	iOS	Role-Based Access Control (RBAC)	Defines levels of access based on user roles	Application
Resource authorization	iOS	Authorization Token	Generates a token at the end of a successful authorization process which is used to grant permission	Application

Inspection Mechanisms

An inspection mechanism is a process or tool used to ensure that cloud-based mobile apps meet certain quality and security requirements. Inspection mechanisms involve thoroughly evaluating the source code, architecture, and security of the app to ensure it meets the desired standard. Examples of inspection mechanisms include static code analysis, application security testing, architectural design reviews, and penetration testing. These inspection mechanisms help identify any weaknesses, vulnerabilities, or security issues in the app before it is deployed in the cloud.

Inspection Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Integrity	Android	ProGuard	Code obfuscation	8
Confidentiality	iOS	Secure store	Keychain security	7
Authentication	Android	SafetyNet API	Attest the device integrity	7
	Android	Android Keystore	Keystore security	7
	iOS	Apple push notification service (APNS)	Authentication message	7
Data Validation	Android	DX Guardrail	Verification of data model	7
	iOS	SwiftLint	Static analysis	7# Logging Mechanisms

An inspection mechanism is a process or tool used to ensure that cloud-based mobile apps meet certain quality and security requirements. Inspection mechanisms involve thoroughly evaluating the source code, architecture, and security of the app to ensure it meets the desired standard. Examples of inspection mechanisms include static code analysis, application security testing, architectural design reviews, and penetration testing. These inspection mechanisms help identify any weaknesses, vulnerabilities, or security issues in the app before it is deployed in the cloud.

Logging Mechanisms Examples:

Security Requirement	Mobile Plataform	Mechanism	Description	OSI Layer
Authentication	iOS	DeviceCheck	DeviceCheck enables customers to securely store small bits of data on Apple devices during the coding and runtime phases	Application
Access Control	iOS	KeyChain	Appleâ€™s Keychain, is a encrypted storage system that primarily stores passwords, certificates, and encryptionkeys	Application
Auditing	Android	Syslog	System logging mechanism for capturing and persistently logging system and audit-specific events in the Android OS	Transport
Logging	Android	LumberJack	Logging mechanism for logging the events for mobile applications	Application

Device Detection Mechanisms

Security Device Detection Mechanisms in Cloud-based mobile apps are technologies responsible for detecting the mobile device that is used to access the application. The mechanisms can vary from OS-level or device-level properties and can include biometrics such as facial recognition, fingerprint scanning, and voice recognition. These mechanisms allow cloud-based mobile apps to detect the device used and ensure that only authorized devices are able to access the app, providing an extra level of security against potential malicious activity.

Device Detection Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Coding Phase	iOS	Mobile App Wrapping	A tool used to secure enterprise apps	Application
Coding Phase	Android	App Reverse Engineering Protection	A technique used to protect code from reverse engineering	Application
Runtime	iOS	Jailbreak Detection	Detects if the device is jailbroken or not	Application
Runtime	Android	Root Detection	Detection of rooted devices	Application

Physical Location Mechanisms

Security physical location mechanisms are applied to cloud-based mobile apps to ensure that user data is not accessed or stored from locations outside of an approved geographic region. These mechanisms include technologies such as geofencing and IP address tracking. Geofencing verifies that user data is being

accessed and stored within a predetermined geographic area by creating a virtual fence around the area. IP address tracking allows mobile apps to identify the geographical location associated with a particular IP address in order to verify that a user is located in the approved geographic area. These security location mechanisms are essential for cloud-based mobile apps, as they help prevent unauthorized access to user data from malicious actors located in remote locations.

Physical Location Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Authenticated Access	iOS	Biometric Scanner	Uses user's fingerprints as part of the authentication process	Physical
Data Integrity	Android	Transparent Encryption	Files are encrypted transparently and automatically	Network
Data Availability	Both	Secure Boot & Root	Ensures that all parts of system are authenticated and verified	Physical
Data Confidentiality	iOS	App sandboxes	Prevents unauthorized access to specific files	Application
Data Security	Android	Full Disk Encryption	Encrypts all data on device	Network

Confinement Mechanisms

Security Confinement Mechanisms in Cloud-based mobile apps refer to the various measures put in place by app developers to help ensure the security and integrity of data within the app. These mechanisms might include measures like authentication requirements, security protocols, encryption, tokenization, application sandboxing, and isolated virtual machines. These measures help limit the risk of data theft or compromise within a cloud-based mobile application.

Confinement Mechanisms Examples:

Security Requirement	Mobile Platform	Mechanism	Description	OSI Layer
Vulnerability Protection	Android	Flask	Flask is a Python web development framework used to protect against malicious code injections	Application Layer
Isolation of Data	iOS	Security-Enhanced Linux (SELinux)	SELinux is a Linux kernel security module used to isolate code from its data	Network Layer
Security of Data	Blackberry	BitLocker	BitLocker is a Windows data encryption system meant to protect data while it is stored	Data Link Layer
Secure Communications	Symbian	IPsec	IPsec is a protocol suite used in secure communication by authenticating and encrypting data	Presentation Layer
Secure Data Transfer	Palm	DM-Crypt	DM-Crypt is a drive encryption system meant to protect data while it is transferred	Session Layer

IoT and LoRa Security Mechanisms

IoT ecosystem represents a very flexible way of organizing smart applications and building consumer-oriented infrastructure. However, there are a number of issues that affects the security and privacy of the involved parties when it comes to low-power IoT devices. Often, there is a trade-off between implementing cybersecurity measures and maintaining operations within given tolerances. As a result, implementation of data protection and cyber defence mechanisms comes as the last priority.

Seguem, abaixo, o resumo dos mecanismos de seguranÃ§a a implementar, para o ecosistema IoT e LoRa.

Security Requirement	Mobile Platform	Mechanism Name	Description	Mechanism Example	OSI Model Layer
Data Confidentiality	Android, iOS	End-to-End Encryption	Ensures that only authorized parties can access data in transit.	TLS 1.3 for HTTPS communication between app and cloud	Transport, Application
	Android, iOS	AES Encryption	Encrypts sensitive data stored on devices and in the cloud.	AES-256 for local storage and cloud databases	Application, Presentation
Data Integrity	Android, iOS	Data Integrity Checks	Validates that data is not altered during transmission or storage.	HMAC-SHA256 for message authentication	Transport, Application
	Android, iOS	Digital Signatures	Ensures authenticity and integrity of data sent between devices and servers.	RSA/ECC signatures for sensitive data exchange	Application

User Authentication	Android	OAuth 2.0 / OpenID Connect	Secure user authentication to access cloud services.	Firebase Authentication	Application
	iOS	Biometric Authentication	Ensures only authorized users can access the app.	Face ID / Touch ID	Application, Presentation
	Android, iOS	Multi-Factor Authentication	Adds an extra layer of security by combining passwords and OTPs.	Google Authenticator	Application
Access Control	Android, iOS	Role-Based Access Control	Restricts access based on user roles to limit data exposure.	AWS IAM Policies	Application
	Android, iOS	Mobile Device Management	Enforces security policies on mobile devices, especially for lost/stolen cases.	Microsoft Intune	Application, Network
Data Privacy	Android, iOS	Data Anonymization	Protects user privacy by masking personal identifiers before analysis.	Pseudonymizing names and addresses	Application
	Android, iOS	Encrypted Identifiers	Uses temporary, encrypted IDs for user tracking to protect privacy.	Subscription Concealed Identifier (SUCI) in 5G	Network, Application
Secure Communication	Android, iOS	VPN / IPsec	Encrypts all traffic over untrusted networks, like public Wi-Fi.	OpenVPN or IPsec for Ethernet connections	Network, Data Link
	Android, iOS	LoRaWAN AES Encryption	Encrypts data transmitted over LoRa networks.	LoRaWAN AES-128 for IoT sensor data	Network, Data Link
Device Authentication	IoT Devices	Device Certificates	Authenticates IoT devices to prevent unauthorized access.	X.509 certificates for device authentication	Data Link, Network
	Android, iOS	Mutual Authentication	Ensures both server and device verify each other's identity.	TLS with mutual certificates	Transport, Application
Network Security	Android, iOS	Firewalls	Filters network traffic to block malicious connections.	Cloudflare WAF for cloud server protection	Network
	IoT Devices	Intrusion Detection Systems	Monitors traffic for suspicious activities and potential breaches.	Snort IDS for IoT and cloud networks	Network, Application
Data Availability	Android, iOS	DDoS Protection	Protects the cloud backend from Distributed Denial of Service attacks.	AWS Shield for cloud services	Network, Application
	Android, iOS	Load Balancers	Distributes traffic to prevent overload and ensure service uptime.	AWS Elastic Load Balancer	Transport
Firmware and Software Updates	IoT Devices	Secure Firmware Updates	Ensures devices receive authenticated and encrypted updates.	Over-the-air updates with digital signatures	Application, Data Link
	Android, iOS	App Store Verification	Ensures only approved and verified apps are installed on devices.	Google Play Protect / Apple App Store policies	Application
Compliance & Auditing	Android, iOS	Logging & Auditing	Tracks access and changes to sensitive data for compliance.	AWS CloudTrail for monitoring access logs	Application, Network
	Android, iOS	GDPR / CCPA Compliance	Ensures compliance with data protection regulations for user data.	User data access requests, consent management	Application

References

| 1. Bouzidi, M., Gupta, N., Cheikh, F. A., Shalaginov, A., & Derawi, M. (2022). A novel architectural framework on IoT ecosystem, security aspects and mechanisms: a comprehensive survey. IEEE Access, 10, 101362-101384. 2. Devalal, S., & Karthikeyan, A. (2018, March). LoRa technology-an overview. In 2018 second international conference on electronics, communication and aerospace technology (ICECA) (pp. 284-290). IEEE.