UDC 004.056.5

Data security in Android applications for the financial sector

E. Ponomarev

*National Research Lobachevsky State University of Nizhny Novgorod,
Nizhny Novgorod, Russia*

This article examines the features of financial applications on the Android platform and analyzes the main types of data used directly in such digital products. The focus is on security methods and technologies, including encryption, authentication, and authorization. Recommendations for secure development are provided to prevent potential cyber-attacks and ensure the protection of users' confidential information. The importance of regular security audits and updates is emphasized to address emerging threats in the mobile financial sector.

Keywords: data security, Android, financial sector, mobile applications, encryption, vulnerabilities, multi-factor authentication.

Безопасность данных в Android-приложениях для финансового сектора

Е. Пономарёв

*Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского, Нижний Новгород, Россия*

В данной статье рассматриваются особенности финансовых приложений на платформе Android, а также анализируются основные типы данных, которые используются в таких цифровых продуктах. Основное внимание уделяется методам и технологиям обеспечения защиты, включая шифрование, аутентификацию и авторизацию. Представлены рекомендации по безопасной разработке для предотвращения потенциальных кибератак и обеспечения сохранности конфиденциальной информации пользователей. Подчеркивается важность регулярных проверок безопасности и обновлений для устранения возникающих угроз в сфере мобильных финансов.

Ключевые слова: безопасность данных, Android, финансовый сектор, мобильные приложения, шифрование, уязвимости, многофакторная аутентификация.

Introduction

The proliferation of mobile financial applications has transformed the way individuals and businesses handle their monetary activities, making services such as banking, investments, and payments more accessible and efficient. A widely used mobile operating system (OS) Android plays one of the central roles in this digital transformation. But with the increasing reliance on apps for financial transactions comes a heightened risk of data breaches, cyber-attacks, and unauthorized access to sensitive information.

These types of apps process vast amounts of sensitive data, ranging from personal identification details to payment credentials. This makes them prime targets for cybercriminals who exploit vulnerabilities in both the software and the underlying platform. Security breaches can lead to severe outcomes, including monetary losses, identity theft, and a significant erosion of trust in digital financial services.

Given the nature of the data processed in these applications, it is imperative to implement robust safeguards measures that can protect user information at every stage – both during storage and transmission. Developers and institutions face a complex landscape where ensuring security is not only a technical challenge but also a regulatory requirement. The purpose of this paper – to analyze effective data security features and protection methods in financial applications on the Android platform.

Main part. Features of financial applications on the Android platform and their characteristics

The development of Android apps for the monetary sector has seen significant growth due to the platform's widespread adoption and flexibility. It makes an attractive choice for developers creating financial solutions such as banking apps, investment platforms, and payment systems. As a result, Android retains its position as the world's leading mobile OS in the second quarter of 2024 (fig.) [1].

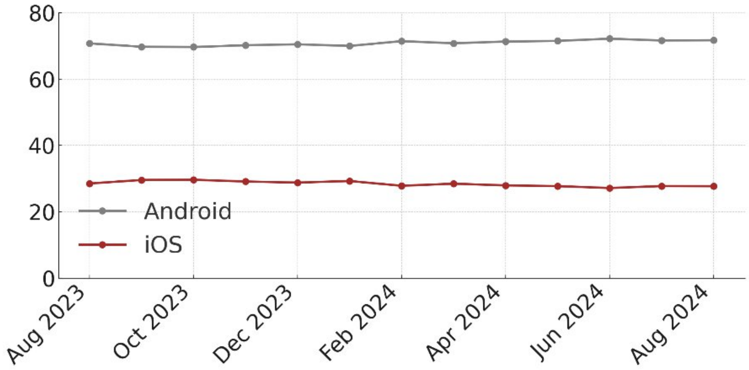

Figure. Mobile OS market share worldwide, %

This popularity among developers is driven not only by the platform's accessibility but also by the increasing demand for mobile payment services, making Android a dominant force in the mobile app market. These digital products handle a wide range of sensitive data that requires robust protection. It includes personal identification information such as names, addresses, phone numbers, and Social Security numbers, as well as financial details like bank account numbers, credit and debit card information, transaction histories, and account balances. These applications also process authentication credentials, including usernames, passwords, PINs, and biometric data such as fingerprints or facial recognition.

Due to the nature of the data processed, any security breach can lead to severe consequences, including identity theft, unauthorized access to funds, and financial fraud. According to statistics [2], in 2024, the average global cost of a data breach was $4,88 million. As such, these applications are prime targets for cybercriminals looking to exploit vulnerabilities in both the software and the underlying platform.

The Android platform provides both opportunities and challenges for financial app developers. One of the key characteristics is its open-source architecture that allows developers to modify and optimize applications according to their specific needs. This also makes Android more susceptible to security vulnerabilities compared to more closed systems like iOS.

The wide variety of Android devices, OS versions, and customizations present challenges in maintaining uniform security standards across all devices. This fragmentation increases the complexity of securing financial applications, as they must be optimized and secured for a wide range of environments. Android’s flexibility allows users to download applications from various sources, not just the official Google Play Store. While this increases accessibility, it also introduces significant security risks as apps from unverified sources might contain malware or lack necessary security updates.

Security measures and methods for Android-based financial applications

Due to the critical sensitivity of the data, maintaining strong security measures is a significant task for various Android banking applications. It is imperative that developers comprehend and implement the fundamental security strategies and practices in order to safeguard user information from unauthorized access, cyber threats, and data breaches.

In this regard, **encryption** is the foundation of data security for financial applications. It ensures that sensitive information is inaccessible to unauthorized users, even if it is intercepted [3]. There are two key types of encryptions that can be employed:

– **Encryption at rest**: all sensitive data stored on the user's device, such as account details, payment information, and user credentials, should be encrypted. Android provides tools such as the **Android Keystore System** and **EncryptedSharedPreferences** for secure local data storage. These systems allow for the storage of cryptographic keys in hardware-backed keystores, making it difficult for attackers to extract keys from the device.

– **Encryption in transit**: data transmitted between the app and the backend servers must be encrypted using secure protocols such as **TLS (Transport Layer Security)**. This ensures that financial data, including login credentials and transaction details, are protected from man-in-the-middle (MITM) attacks.

Developers must ensure that cryptographic credentials are managed securely. Hardcoding these secrets within the application or using weak algorithms for securing data can expose information to attackers. Strong algorithms like **AES-256** (Advanced Encryption Standard) should be employed to guarantee the highest levels of security. In addition to AES-256, there are a number of other approaches that have their own advantages and disadvantages (tabl.) [4, 5].

Table

Comparison of encryption methods

Encryption method	Key length	Strengths	Weaknesses	Common implementations
AES	128, 192, 256 bits	High level of security, efficient for both hardware and software, widely supported.	Requires secure key management, vulnerable to side-channel attacks if improperly implemented.	Used in TLS, encrypted databases, mobile apps.
RSA (Rivest-Shamir-Adleman)	1024-4096 bits	Strong encryption for key exchange, widely used for securing internet communications.	Slow performance for large datasets, vulnerable to quantum computing attacks.	Employed in SSL/TLS protocols, digital certificates.
ECC (Elliptic Curve Cryptography)	160-521 bits	Provides high security with smaller key sizes, efficient for mobile and embedded systems.	More complex to implement, requires advanced expertise.	Used in SSL/TLS, blockchain, modern cryptography

In financial applications, securely storing user data on the device is serious. While Android offers several options for data storage, not all are suitable for sensitive banking information. Sensitive data should always be stored in internal storage rather than external storage, as internal storage is more secure and private to the apps.

For managing cryptographic keys, the **Android KeyStore API** allows for the generation and storage of keys in hardware-protected environments. This ensures that even if a device is compromised, the keys remain secure. Financial applications should use encrypted databases such as **SQLCipher** for storing sensitive user information. This ensures that even if the database is accessed by an attacker, the information remains unreadable. It is also important to minimize data retention on the device. Data that is no longer needed should be securely deleted, and personal or financial information should not be unnecessarily cached.

A key component of financial application security is ensuring that only authorized users can access sensitive data and functionality. To achieve this, several authentication mechanisms should be implemented. One of them is **multi-factor authentication (MFA)**, which adds an extra layer of security [6]. This approach generally requires the use of two or more distinct authentication factors, such as knowledge-based credentials (e.g., passwords), possession-based tokens (e.g., one-time passwords or physical devices), and biometric verification methods (e.g., fingerprint or facial recognition).

Another protective method is biometric authentication, such as the **BiometricPrompt API**, which offers a secure and user-friendly way to integrate biometric verification into financial applications. This approach enhances security by reducing reliance on traditional password-based systems, which can be vulnerable to brute-force or credential-stuffing attacks. For mobile monetary software that integrate with third-party services, using secure authentication frameworks such as **OAuth 2.0** and **OpenID Connect** ensures that user credentials are not exposed or misused. These protocols enable secure token-based authentication that enhances user privacy and security.

Proper **session management** is important in preventing session hijacking and unauthorized access. Sessions should be short-lived, and tokens should be refreshed periodically using secure methods. Token-based authentication with **JWT (JSON Web Tokens)** should be implemented with proper expiration times and token invalidation mechanisms.

Mobile applications, particularly on Android, are susceptible to reverse engineering, where attackers can decompile the app to extract sensitive information such as cryptographic keys, application programming interfaces (API) endpoints, and business logic. To reduce this risk, developers should use tools like **ProGuard or R8 for code obfuscation**, which transforms readable code into an unreadable format, making it harder for attackers to reverse-engineer the application and access sensitive information. **App hardening** techniques should be applied to add extra layers of protection against tampering, reverse engineering, and malware injection. These techniques include anti-debugging mechanisms, integrity checks, and rooting or jailbreaking detection, ensuring that the application does not run in compromised environments.

Financial applications rely heavily on backend API to interact with banking systems, process payments, and retrieve data, making API security crucial for the overall safety of the application. All API endpoints should be protected through secure authentication and authorization mechanisms, such as OAuth tokens, ensuring that sensitive operations are only accessible to authorized users. Implementing **rate limiting** and **throttling** can be essential to defend against denial-of-service (DoS) attacks, preventing malicious actors from overwhelming the system with excessive requests. Server-side input validation is also significant to safeguard against injection attacks, such as SQL or command injection, which could result in data theft or unauthorized access.

Securing Android-based financial apps requires a comprehensive approach that integrates encryption, secure storage, strong authentication, and continuous monitoring. By implementing these security measures, developers can significantly reduce the risk of data breaches and ensure the safety of sensitive financial information. Regular updates, code obfuscation, and secure API communication further reinforce the security posture of the application, making it more resilient against evolving cyber threats.

Recommendations for secure development of Android-based finance apps

To ensure the security of various banking apps on the Android platform, developers must adopt **secure coding practices** and integrate **vulnerability assessment tools** throughout the development lifecycle. Secure coding involves writing code that is not only functional but also resilient to attacks and potential exploits. This requires developers to follow established best practices, such as validating input, avoiding hardcoded credentials, properly managing sensitive data, and using encryption standards to safeguard user information [7].

One of the most effective ways to maintain security throughout the development process is by leveraging tools designed to identify and mitigate vulnerabilities in mobile applications. A key resource in this area is the **OWASP Mobile Security Testing Guide (MSTG)**, which provides comprehensive guidelines and methodologies for assessing the security of mobile applications, including those on the Android platform. The MSTG covers a wide range of security aspects, such as data storage, cryptography, network communications, and authentication mechanisms, offering actionable recommendations for addressing potential weaknesses.

By incorporating security testing early in the development cycle and continuously throughout the product's lifecycle, developers can proactively identify and address vulnerabilities before they are exploited [8]. This process can be further supported by automated tools that scan code for known issues, as well as manual penetration testing to simulate real-world attack scenarios. In combination, these practices help ensure that the final product not only meets functional requirements but also protects sensitive monetary data from unauthorized access and cyber threats.

Conclusion

As the use of Android financial applications continues to grow, ensuring robust data security has become an essential priority for developers and financial institutions. By implementing key security practices such as encryption, secure data storage, strong authentication mechanisms, and regular security audits, organizations can significantly reduce the risk of unauthorized access, cyber-attacks, and data breaches. As the threat landscape evolves, monetary digital products must continuously adapt their security measures to stay ahead of emerging risks and ensure that users' data remains safe while they provide seamless and reliable services.

REFERENCES

1. Mobile Operating System Market Share Worldwide // StatCounter Global Stats. – URL: https://gs.statcounter.com/os-market-share/mobile/worldwide [Accessed 10th September 2024].

2. Cost of a Data Breach Report 2024 // IBM. – URL: https://www.ibm.com/reports/data-breach [Accessed 13th September 2024].

3. Zhang Q. An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption / Q. Zhang // 2021 2nd International Conference on Computing and Data Science (CDS). – IEEE, 2021. – pp. 616-622.

4. Israfilov A. Cybersecurity in the automotive industry: vulnerabilities and protection / A. Israfilov // Sciences of Europe. – 2024. – No. 145 (145). – pp. 60-63.

5. Alexan W. IoMT Security: SHA3-512, AES-256, RSA and LSB Steganography / W. Alexan, A. Ashraf, E. Mamdouh, S. Mohamed, M. Moustafa // 2021 8th NAFOSTED Conference on Information and Computer Science (NICS). – IEEE, 2021. – pp. 177-181.

6. Bushuev S. Economic aspects of Big Data: analysis of data privacy protection methods / S. Bushuev // Cold Science. – 2024. – No. 7. – pp. 25-35.

7. Sharma A. Security of Android Banking Mobile Apps: Challenges and Opportunities / A. Sharma, S.K. Singh, S. Kumar, A. Chhabra, S. Gupta // International Conference on Cyber Security, Privacy and Networking (ICSPN 2022). – Cham: Springer, 2023. – pp. 406-416.

8. Bobunov A. Development of test automation methodologies in the financial sector: a comparative analysis of approaches in the USA, Europe, and Asia / A. Bobunov // Cold Science. – 2024. – No. 2. – pp. 61-70.

INFORMATION ABOUT THE AUTHOR

Ponomarev Evgenii, bachelor's degree, National Research Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia.
e-mail: ponomarev_evgenii@rambler.ru