Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

Adaptive Differential Privacy for Protecting User Confidential Information on Android Devices

Manish Verma*

*Corresponding author, PhD Scholar, Department of CSE, SSET, Sharda University, Greater Noida, U.P. India. E-mail: manishverma649@gmail.com

Parma Nand

Prof., Department of CSE, SSET, Sharda University, Greater Noida, U.P., India. E-mail: parma.nand@sharda.ac.in

Journal of Information Technology Management, 2025, Vol. 17, Special Issue, pp.155-167.
Published by the University of Tehran, College of Management
doi: https://doi.org/10.22059/jitm.2025.102933
Article Type: Research Paper

Received: January 17, 2025
Received in revised form: March 03, 2025
Accepted: June 13, 2025
Published online: August 01, 2025

Abstract

The widespread adoption of Android phones has heightened concerns about user privacy. This research presents an Adaptive Privacy Management System (APMS) that integrates Machine Learning (ML) models with Differential Privacy techniques to enhance privacy protection. The APMS monitors application behavior and employs ML algorithms to detect anomalies and enable context-aware privacy enforcement. Differential Privacy ensures that sensitive data remains protected through the addition of noise and privacy-preserving computations. Experimental results demonstrate that the APMS achieves a 92.5% accuracy rate in detecting the privacy leakage. The anomaly detection model, using Random Forest, shows high accuracy (92.5%), recall (89.5%), and precision (73.9%), effectively identifying both normal and anomalous behaviors. Additionally, the impact of noise on data utility, controlled by the privacy budget (ε), is manageable. The results show that APMS is a robust system for safeguarding user confidential information, contributing to a more secure and privacy-centric Android ecosystem.

Keywords: Android Security, Data Protection, Confidential Information, Data Leakage, App Vul-nerabilities

Introduction

The exponential growth of Android devices has fundamentally reshaped the digital landscape, offering unprecedented connectivity and convenience to billions of users worldwide. According to a Statista report (2024), Android dominates the mobile operating system market, powering over 70% of all smartphones globally. This widespread adoption underscores the critical importance of securing user confidential information against the backdrop of escalating cyber threats. The open-source nature of the Android ecosystem and diverse application environment present security challenges, making it a prime target for malicious actors (Enck et al., 2014).

Data breaches and unauthorized access to sensitive information are frequent. Personal data, financial records, and proprietary information are targeted, which leads to severe financial losses, reputational damage, and privacy violations (Zhang et al., 2022). High-profile incidents that exposed millions of users' personal information highlight the urgent need for robust security measures.

To protect against privacy leakage, several advanced techniques were proposed. These techniques are broadly classified into static analysis, dynamic analysis, hybrid analysis, and machine learning-based approaches. Static analysis involves reviewing an application's source code to identify potential privacy risks without executing the application (Verma & Nand, 2022). This approach enables developers to detect vulnerabilities early in the development cycle, reducing the likelihood of privacy breaches before deployment (Arzt et al., 2014). In contrast, dynamic analysis monitors an application's behavior during runtime to uncover privacy violations in real-world scenarios, making it particularly effective in identifying issues that may not be visible in static code (Enck et al., 2014). Hybrid techniques are an amalgamation of both static and dynamic techniques. It takes advantage of both techniques. Machine learning models identify the patterns of suspicious behavior of an app (Demontis et al., 2019). While these techniques offer significant advancements in detecting privacy leaks, they also face challenges such as computational overhead, scalability, and adaptability to the constantly evolving landscape of cyber threats (Xu et al., 2021).

Further, emerging technologies offer promising avenues to enhance Android security further. For example, Machine learning can be leveraged to detect anomalous behaviors indicative of potential security breaches, while blockchain technology can provide immutable records of data transactions, enhancing transparency and traceability (Cholevas et al., 2024). Advanced encryption techniques ensure that even if data is intercepted, it remains unintelligible to unauthorized users.

This research introduces an Adaptive Privacy Management System (APMS) that synergizes Machine Learning (ML) models with Differential Privacy techniques to enhance the protection of sensitive data on Android devices by monitoring application behavior. ML

algorithms were employed to detect anomalies. Context awareness further strengthens our approach by incorporating contextual information such as location, time, and user activity to make more accurate and privacy-preserving predictions. Differential Privacy adds a layer of security by incorporating noise into data queries, thereby preserving the privacy of individual data points (Heinrich et al., 2024). This technique ensures that sensitive information is not compromised, even when aggregated data is analyzed. The APMS dynamically adjusts privacy settings based on user context, historical usage patterns, and identified risks, thus permitting data access only under secure conditions.

The remaining section of this paper is as follows. Section II reports a literature review. Section III focuses on the proposed methodology. Section IV discusses the obtained results, and Section V concludes the paper.

Literature Review

The protection of user privacy within the Android ecosystem has become an increasingly critical area of research, driven by the widespread use of mobile applications and the sensitivity of data they access. The literature presents a variety of frameworks and methodologies aimed at enhancing Android privacy protection through improved permission handling, adaptive mechanisms, and user-centred designs. Android's permission-based security model allows users to control application access to sensitive resources such as location, contacts, and media files. Despite its importance, researchers have identified critical shortcomings in this model. Many applications request permissions unrelated to their core functions, leading to unnecessary data exposure and potential privacy violations (Harikrishnan & Periyasamy, 2024; Zhang et al., 2023). One of the major concerns raised in the literature is over-privileging, where applications request excessive permissions beyond their functional needs. Studies indicate that apps, especially in the health sector, frequently collect more information than required (Davis, 2023; Zhang et al., 2023). To address this, a framework was proposed to restrict extraneous permissions and limit unauthorized data access (Mishra et al. 2022). In response to permission misuse, fine-grained permission control frameworks were introduced. These systems offer users the ability to grant permissions at a more detailed level, improving autonomy and aligning permission grants with specific app functionalities. For instance, UIPDroid enables widget-level permission settings (Duan et al., 2022), while runtime monitoring introduces the mechanisms to ensure permission usage adheres to declared privacy policies (Wu et al., 2018).

Extending the concept of granularity, context-aware permission management dynamically adjusts permission grants based on contextual factors such as user location, behaviour, or time of access. This adaptive approach ensures that sensitive resources are only accessed when contextually appropriate (Abdella et al., 2016). The SmartGuard framework exemplifies this model by personalizing location privacy controls according to user preferences and device

context (Niu et al., 2021). Despite advancements in permission frameworks, low user awareness remains a significant obstacle to effective privacy management. Studies reveal that most users rarely review or modify app permissions (Alsoubai et al., 2022; Scoccia et al., 2018). To bridge this awareness gap, DopCheck, a framework was proposed that monitors changes in privacy policies and alerts users to data-sharing practices, thereby promoting informed decision-making (Yan et al., 2024).

Developers often request extensive permissions due to unclear guidelines or reliance on third-party libraries. Research underscores the importance of educating developers and enforcing best practices (Tahaei et al., 2023; Alkindi et al., 2021). Frameworks such as LP-Guardian demonstrate that comprehensive privacy can be achieved without degrading user experience (Fawaz & Shin, 2024).

Recent efforts in privacy protection emphasize user-centric approaches, where users are empowered to manage, restrict, or revoke permissions based on their preferences. Frameworks like CUPA and AFP offer personalized privacy management without disrupting app functionality (Scoccia et al., 2018; Alkindi et al., 2020). Moreover, intent-aware permission architectures have been developed to enhance user understanding by clearly communicating the purpose behind data access (Rahman et al., 2022).

In summary, the existing body of literature highlights the transition from traditional, static permission models to context-sensitive and user-oriented frameworks. These innovations reflect a growing emphasis on empowering users, reducing unnecessary data access, and maintaining usability. all of which are essential in addressing contemporary privacy challenges within the Android environment.

Methodology

This section outlines the methodology adopted for creating the APMS framework, detailing the data collection process, Machine Learning (ML) model development, Differential Privacy (DP) mechanism implementation, privacy enforcement, and system evaluation.

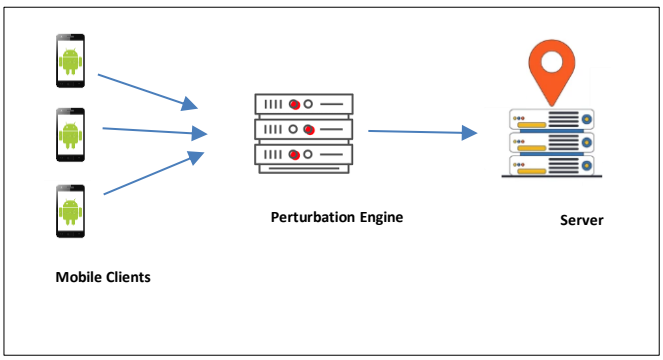

Figure 1. System overview

Figure 1 shows the flow of data and interactions in an Adaptive Privacy Management System (APMS). It starts with the Client, represented by a user device, which sends raw data (e.g., precise location information) to the Perturbation Engine. The Perturbation Engine applies Differential Privacy to obfuscate the sensitive data, ensuring user anonymity and privacy. The perturbed data is then forwarded to the Server, which processes the obfuscated information to provide location-based services. Finally, the results are sent back to the Client, maintaining a balance between data utility and user privacy throughout the process.

Data Collection and Preprocessing: Data collection is the foundation of the proposed system, focusing on gathering user behavior, app access patterns, and contextual information (e.g., location, time, and network state). User behavior data is essential to detect anomalies and develop privacy models tailored to individual users.

Collected Features:

1. User behaviour: app access frequency, app usage behaviour, access to sensitive data
2. Context-sensitive: Location, time of access, Device status, Sensitivity level of data accessed by the app:
3. App-specific behaviour: permission, api calls, third-party library

These features were collected by using Android's API's such as UsageStatsManager, ConnectivityManager, LocationManager, and PackageManager. Data, preprocessing techniques such as feature extraction, normalization, and noise reduction were applied to remove the irrelevant or noisy data and make it suitable for machine learning models.

Let $X = \{ u_{\mathrm{i}}, c_{\mathrm{i}} , a_i\}$, represent the set of data points collected from Android applications, where:

u_{i} : user behavior (e.g., frequency of access, data type accessed)

c_{i} : contextual information (e.g., location, time, network condition, battery information)

a_i : app-specific parameters (e.g., permissions, app metadata)

The data points are preprocessed, and key features F= {f1, f2, …, fm} are extracted from X, where each feature represents a relevant factor for privacy management, such as app access frequency, user context, and app behavior.

Machine Learning Model Development

According to Ashisha et al. (2023), ML techniques are very popular in medical applications for predicting different disorders. Machine Learning (ML) enhances Adaptive Privacy

Management Systems (APMS) by enabling dynamic, context-aware privacy protection. It analyzes user behavior to predict privacy preferences, allowing the system to personalize privacy settings in real-time. ML also supports the detection of anomalies and privacy risks, ensuring that unusual patterns are flagged and privacy violations are prevented. "Additionally, machine learning optimizes the balance between privacy and utility, ensuring data remains useful while safeguarding sensitive information. Thangamayan et al. (2024) show that Random Forest and Gradient Boosting achieved the highest accuracy, while Logistic Regression performed the worst.

In this step, two key ML models are developed:

The model analyzes the permissions requested by an app and its network traffic patterns, comparing them against a baseline to predict suspicious behavior and help prevent user data leakage. Fuzzy logic & machine learning algorithms are combined for risk prediction (Pattun et al., 2023).

Anomaly Detection Model: designed to identify unusual patterns in data that deviate from the norm. In the context of privacy leakage, this model helps detect activities that are out of the ordinary or deviate from expected user behaviors. The Isolation Forest efficiently detects outliers in a high-dimensional app dataset. The model detects deviations from expected patterns (anomalies) that may indicate privacy risks. The anomaly detection model techniques used to flag unusual access patterns, as demonstrated by Kyritsis et al. (2018). Mathematically, anomaly detection can be shown as:

Let f(x) represent the function that maps feature vectors to a normal behavior profile. The model is trained on the historical data Xtrain to learn normal behavior N(x):

$$N(x) = \mathrm{E}\,[\, f(x) \mid x \epsilon\, X_{train}] \tag{1}$$

For a new data point x_t , the anomaly score $A(x_t)$ is computed as the deviation from normal behavior:

$$A(x_t) = [\, |F(x_t) - N(x)|\,]^2 \tag{2}$$

If $A(x_t) > \delta$, where δ is a predefined threshold, the behaviour is flagged as anomalous, indicating a potential privacy violation.

Context-Aware Privacy Prediction: The model predicts optimal privacy settings based on user-specific behavior and contextual information. For instance, if a user accesses sensitive information from a trusted network, the privacy settings may differ from when they access it from an unknown public network. By incorporating context, the model ensures that the privacy settings adapt dynamically, a technique supported by the recent work by Bou C. et al. (2021) on context-aware systems.

Let C (u,c) be a context-aware function that predicts the optimal privacy settings for a user u based on their context c. This function is represented as:

$$P(x_t) = C(u_t, c_t) \quad (3)$$

The function $P(x_t)$ is trained on historical data to predict privacy settings, where u_t represents user-specific factors and c_t represents contextual factors such as location, time, and network conditions.

The ML model for privacy prediction is trained to minimize the error in predicting privacy settings. Given a set of training data x_{train} , the objective is to minimize the prediction error:

$$\min \sum_{x_t \in x_{train}} [|P(x_t) - y_t|]^2 \quad (4)$$

where y_t is the ground truth for privacy settings.

Noise Impact or Differential Privacy Mechanism Implementation

To ensure Differential Privacy, we apply noise η to the data query results. Let Q(x) be the query function for the data access request, and let ΔQ represent the global sensitivity of the query:

$$\Delta Q = \max |Q(x) - Q(x')| \quad (5)$$

For each query, noise from a Laplace distribution $Lap(b)$ where $b = \frac{\Delta Q}{\epsilon}$, is added to preserve privacy:

$$Q(x)_{DP} = Q(x) + \eta \quad (6)$$

$$\eta \sim Lap(\frac{\Delta Q}{\epsilon}) \quad (7)$$

Here, ϵ represents the privacy budget. The privacy loss $\mathcal{L}$ for a user u is defined as:

$$\mathcal{L}(\mathrm{u}) = \sum_{i=1}^{n} \epsilon_i \quad (8)$$

where ϵ_i is the privacy budget used for each query i. The system tracks $\mathcal{L}(\mathrm{u})$ and adjusts noise levels dynamically to maintain privacy.

The privacy enforcement mechanism is based on the results of the anomaly detection and context-aware privacy prediction models.

- **Anomaly-Based Enforcement**: If $A(x_t) > \delta$, the system enforces stricter privacy settings by increasing noise levels in the DP mechanism or denying the app access request.

- **Context-Aware Enforcement**: The system dynamically adjusts privacy settings based on the output $P(x_t)$ of the context-aware model. If the predicted privacy setting is stricter than the current setting, the system enforces the stricter setting.

System Performance Evaluation

This multi-faceted validation approach, including simulations and user studies, aligns with current trends in privacy management research.

The performance of the ML models is evaluated using metrics such as accuracy A, precision P, recall R, and F1-score F1, defined as:

$$A = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \quad (9)$$

$$P = \frac{TP}{TP+FP} \quad (10)$$

$$R = \frac{TP}{TP+FN} \quad (11)$$

$$F1 = 2.\frac{P.R}{P+R} \quad (12)$$

where:

- TP: true positives (correctly detected anomalies or privacy violations)
- TN: true negatives (correctly identified normal behaviors)
- FP: false positives (incorrectly flagged normal behaviors)
- FN: false negatives (missed anomalies or privacy violations)

The goal is to maximize A, P, R, and F1 to ensure high accuracy and effectiveness in managing privacy.

Results

The accuracy of the anomaly detection model

- Dataset: Simulated user behaviour, app access patterns, and contextual information (e.g., location, network state).
- Threshold for Anomalies: Set to detect 10% contamination (i.e., 10% of the data are anomalous).

Table 1. Confusion matrix for anomaly detection

	Predicted Positive	Predicted Negative
Actual Positive	85	30
Actual Negative	10	200

Table 2. Performance of anomaly detection

Metric	Value
Accuracy	92.50%
Precision	73.90%
Recall	89.50%
F1-Score	80.90%

In Table 2, the performance of anomaly detection has been calculated from Table 1.

- **Accuracy** is high at 92.5%, meaning that the anomaly detection system can correctly identify normal and anomalous behaviors most of the time.
- **Recall** is 89.5%, indicating that the system correctly identifies actual anomalies.
- **Precision** is lower (73.9%) due to a moderate number of false positives, but still reflects reliable detection.

Privacy Breach Reduction

The effectiveness of APMS privacy has been measured using the breach metric. Unauthorized app requests for sensitive information (e.g., contacts, location) are simulated. Data breaches are measured with and without the APMS being enabled.

Table 3. Privacy Breach Metric

Privacy Breach Metric	Without APMS	With APMS	Reduction (%)
Number of Breaches	45	8	82%
Unauthorized Access	1000	175	82.50%

Table 3 shows the reduction in privacy breaches by 82%, demonstrating that it successfully denies suspicious app access and adjusts privacy settings based on real-time anomaly detection and context-aware models.

Noise Impact on Data Utility (Differential Privacy Evaluation)

Adding noise to data using differential privacy mechanisms helps protect user privacy by distorting app queries that request personal information. It ensures that individual data remains confidential while still allowing for useful insights to be derived from the data.

Overall System Performance

Table 4. Privacy Breach Metric

Metric	Value
Anomaly Detection Accuracy	92.50%
Privacy Breach Reduction	82%

- The APMS demonstrates strong performance in both anomaly detection and privacy breach reduction.
- The lower privacy budget ε value ensures more privacy but may degrade utility.

Conclusion

The Adaptive Privacy Management System (APMS) performs excellently in both anomaly detection and privacy breach reduction. Anomaly detection accuracy is 92.5%, the system efficiently distinguishes between normal and anomalous behaviors, while a recall rate of 89.5% ensures that most true anomalies are accurately identified. Although the precision, at 73.9%, reflects a moderate level of false positives, it still maintains a reliable level of detection.

APMS significantly reduces 82% privacy breaches, effectively curbing unauthorized app access to sensitive user information. Noise impact has been evaluated through differential privacy mechanisms. Additional noise is effectively controlled by the privacy budget (ε). At smaller values of ε (e.g., 0.1), it ensures more privacy, but noise heavily impacts data utility and vice versa.

In conclusion, APMS effectively balances the need for strong privacy protection with maintaining the usefulness of data. By offering reliable anomaly detection, significantly reducing privacy risks, and allowing flexible control over noise levels, the system demonstrates its ability to safeguard user privacy while minimizing any impact on overall performance.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Abdella, J., Ozuysal, M., & Tomur, E. (2016). CA-ARBAC: Privacy preserving using context-aware role based access control on Android permission system. *Networks*, *00*, 1–23. https://doi.org/10.1002/sec

Alkindi, Z. R., Sarrab, M., & Alzeidi, N. (2021). User privacy and data flow control for Android apps: Systematic literature review. *Journal of Cyber Security and Mobility*, *10*(1), 261–304. https://doi.org/10.13052/jcsm2245-1439.1019

Alsoubai, A., Ghaiumy Anaraky, R., Li, Y., Page, X., Knijnenburg, B., & Wisniewski, P. J. (2022, April 29). Permission vs. app limiters: Profiling smartphone users to understand differing strategies for mobile privacy management. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3491102.3517652

Arzt, S., Rasthofer, S., Fritz, C., Bodden, E., Bartel, A., Klein, J., Le Traon, Y., Octeau, D., & McDaniel, P. (2014). FLOWDROID: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. *ACM SIGPLAN Notices*, *49*(6), 259–269. https://doi.org/10.1145/2594291.2594299

Ashisha, G. R., Mary, A. X., George, T. S., Sagayam, M. K., Fernandez-Gamiz, U., Günerhan, H., Uddin, M. N., & Pramanik, S. (2023). Analysis of diabetes disease using machine learning techniques: A review. *Journal of Information Technology Management*, *15*(4), 139–159. University of Tehran. https://doi.org/10.22059/jitm.2023.94897

Bou Chaaya, K. (2021). *Privacy management in connected environments* (Doctoral dissertation, Université de Pau et des Pays de l'Adour).

Cholevas, C., Angeli, E., Sereti, Z., Mavrikos, E., & Tsekouras, G. E. (2024). Anomaly detection in blockchain networks using unsupervised learning: A survey. *Algorithms*, *17*(5). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/a17050201

Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., Corona, I., Giacinto, G., & Roli, F. (2017). Yes, machine learning can be more secure! A case study on Android malware detection. http://arxiv.org/abs/1704.08996

Duan, M., Jiang, L., Shar, L. K., & Gao, D. (2022). UIPDroid. 227–231. https://doi.org/10.1145/3510454.3516844

Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B. G., Cox, L. P., Jung, J., McDaniel, P., & Sheth, A. N. (2014). TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems*, *32*(2). https://doi.org/10.1145/2619091

Fawaz, K., & Shin, K. G. (2014). Location privacy protection for smartphone users. *Proceedings of the ACM Conference on Computer and Communications Security*, 239–250. https://doi.org/10.1145/2660267.2660270

Harikrishnan, P. R., & Periyasamy, P. (2024, August). A review on the analysis of the effectiveness of permission-based security models in Android apps. In *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)* (Vol. 1, pp. 1451–1459). IEEE.

Kyritsis, A. (2019). *Archive ouverte UNIGE: Enhancing wellbeing using artificial intelligence techniques*. https://doi.org/10.13097/archive-ouverte/unige:130751

Mishra, B., Agarwal, A., Goel, A., Ansari, A. A., Gaur, P., Singh, D., & Lee, H.-N. (2022). Privacy protection framework for Android. *IEEE Access*, *10*, 7973–7988. https://doi.org/10.1109/ACCESS.2022.3142345

Niu, B., Li, Q., Wang, H., Cao, G., Li, F., & Li, H. (2022). A framework for personalized location privacy. *IEEE Transactions on Mobile Computing*, *21*(9), 3071–3083. https://doi.org/10.1109/TMC.2021.3055865

Pattun, G., Afroaz, K., Siddiqui, A. T., & Ghazala, S. (2023). Prediction of type-I and type-II diabetes: A hybrid approach using fuzzy logic and machine learning algorithms. *Journal of Information Technology Management*, *15*, 35–56. https://doi.org/10.22059/jitm.2023.95244

Rahman, M. R., Miller, E., Hossain, M., & Ali-Gombe, A. (2022). Intent-aware permission architecture: A model for rethinking informed consent for Android apps. *arXiv preprint* arXiv:2202.06995.

Scoccia, G. L., Ruberto, S., Malavolta, I., Autili, M., & Inverardi, P. (2018). An investigation into Android run-time permissions from the end users' perspective. *Proceedings of the International Conference on Software Engineering*, 45–55. https://doi.org/10.1145/3197231.3197236

Tahaei, M., Abu-Salma, R., & Rashid, A. (2023, April 19). Stuck in the permissions with you: Developer & end-user perspectives on app permissions & their privacy ramifications. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3544548.3581060

Thangamayan, S., Sinha, A., Moyal, V., Maheswari, K., Harathi, N., & Utama, A. N. B. (2024). Comparative study on different machine learning algorithms for neonatal diabetes detection. *Journal of Information Technology Management*, *16*(1), 5–26. https://doi.org/10.22059/jitm.2024.96359

Verma, M., & Nand, P. (2023). Review on the static analysis techniques used for privacy leakage detection in Android apps. In B. Unhelkar, H. M. Pandey, A. P. Agrawal, & A. Choudhary (Eds.), *Advances and applications of artificial intelligence & machine learning. ICAAAIML 2022* (Lecture Notes in Electrical Engineering, Vol. 1078). Springer, Singapore. https://doi.org/10.1007/978-981-99-5974-7_28

Wu, F., Sun, R., Fan, W., Liu, Y., Liu, F., & Lu, H. (2018). A privacy protection approach based on Android application's runtime behavior monitor and control. *International Journal of Digital Crime and Forensics*, *10*(3), 95–113. https://doi.org/10.4018/IJDCF.2018070108

Xu, R., Baracaldo, N., & Joshi, J. (2021). Privacy-preserving machine learning: Methods, challenges and directions. http://arxiv.org/abs/2108.04417

Yan, C., Meng, M. H., Xie, F., & Bai, G. (2024). Investigating documented privacy changes in Android OS. *Proceedings of the ACM on Software Engineering*, *1*(FSE), 2701–2724. https://doi.org/10.1145/3660826

Zhang, S., Lei, H., Wang, Y., Li, D., Guo, Y., & Chen, X. (2023). How Android apps break the data minimization principle: An empirical study. *Proceedings of the 2023 38th*

IEEE/ACM International Conference on Automated Software Engineering (ASE 2023), 1238–1250. https://doi.org/10.1109/ASE56229.2023.00141

Zhang, X., Yadollahi, M. M., Dadkhah, S., Isah, H., Le, D. P., & Ghorbani, A. A. (2022). Data breach: Analysis, countermeasures and challenges. *International Journal of Information and Computer Security*, *19*(3–4), 402–442. https://doi.org/10.1504/ijics.2022.127169

Bibliographic information of this paper for citing:

Verma, Manish, & Nand, Parma (2025). Adaptive Differential Privacy for Protecting User Confidential Information on Android Devices. *Journal of Information Technology Management*, 17 (Special Issue), 155-167. https://doi.org/10.22059/jitm.2025.102933