SecureTrack: Personalized Security through Behavioral Insights

Ujjwal Baranwal
ASU–Tempe

WeiSheng Chiu
ASU–Tempe

Geeth Nischal Gottimukkala
ASU–Tempe

Devansh Tomar
ASU–Tempe

Dipanshu Singh
ASU–Tempe

Aashritha Machiraju
ASU–Tempe

ABSTRACT

Mobile app security strives to strike a balance between strong protection and user comfort. Our concept of SecureTrack addresses this by using behavioral analytics and machine learning to create individualized security profiles, improve health data management, and ensure compliance with privacy requirements such as GDPR, HIPAA, and CCPA, aiming to outperform traditional static security measures.

1 INTRODUCTION

The incremental usage of mobile health applications has created an unprecedented need for robust security measures that protect sensitive personal health information while maintaining usability. As users rely more on mobile devices to track, store, and share health data, traditional static security approaches have proven inadequate in addressing the complex challenges of modern mobile computing environments. Current security solutions often fail to account for the dynamic nature of user behavior, leading to either security vulnerabilities or excessive authentication requirements that frustrate users.

1.1 *Question: Why do we need to solve this problem?*

Traditional security measures, such as fixed passwords and standard multi-factor authentication, fail to protect sensitive health data, leaving gaps against advanced cyberattacks and causing user frustration. Privacy concerns over cloud-stored health information further highlight the need for stronger defenses. SecureTrack addresses these issues with a dynamic, behavioral analytics-based security framework that adapts to user patterns, offering a robust alternative to static models.

Healthcare data breaches have grown exponentially, with 133 million records compromised in 2023 alone—nearly triple 2022's 51.9 million. Hacking incidents increased by 239% between 2018–2023, now accounting for 79.7% of all breaches. The largest single breach in 2023 affected 11.27 million individuals. [3]

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 *Question: Why is it related to Mobile Computing?*

SecureTrack is fundamentally a mobile computing solution, utilizing device sensors (GPS, accelerometers, touchscreen) for behavioral data collection while integrating with mobile OS security frameworks. The system addresses mobile-specific constraints like battery life and processing power through optimized algorithms for real-time behavioral analysis.

1.3 *Question: Why is the solution new?*

SecureTrack advances mobile security by integrating real-time behavioral analytics and adaptive authentication. It generates dynamic security profiles based on location, timing, and interaction behaviors. The system improves security while lowering user friction by utilizing mobile computing infrastructure through continuous monitoring, sensor integration, and distributed processing.

SecureTrack addresses critical gaps identified across 1,800+ research papers in mHealth security. While 63% of existing solutions lack security guidelines, our framework proposes comprehensive HIPAA/GDPR protocols, expert oversight, and ML-based security testing, targeting 90% threat detection accuracy while maintaining continuous behavioral monitoring and compliance. [1]

Adding onto that, SecureTrack combines behavioral analytics and adaptive authentication, using sensors like location and gestures to create dynamic security profiles. False positives are minimized during data collection with a majority-vote mechanism, filtering outliers, and periodic re-authentication. Clustering techniques further distinguish overlapping user behaviors, ensuring a reliable and adaptive solution over traditional methods.

2 BACKGROUND AND RELATED WORK

As mobile applications increasingly handle sensitive data, robust security mechanisms are crucial. Traditional approaches like passwords and PINs are often insufficient against sophisticated threats such as phishing or credential theft. Multi-factor authentication (MFA), as discussed by Henricks and Kettani [2], addresses these vulnerabilities by requiring multiple forms of verification. While effective, MFA often introduces usability challenges, such as repeated re-authentication during legitimate user activities like travel or device changes.

Recent developments in behavioral analytics, including insights from "Behavioral Biometrics for Mobile User Authentication" [4], highlight the potential to enhance security through continuous monitoring of user behavior. Patterns such as login times, session durations, and geolocation can be analyzed to identify anomalies, enabling dynamic security adjustments. This approach minimizes disruptions for legitimate users while maintaining robust protection against unauthorized access.

SecureTrack builds on these concepts by combining behavioral insights with adaptive security measures, offering a solution tailored to mobile applications that balances usability and protection. Additionally, SecureTrack is evaluated against traditional security measures and existing behavioral-based systems. While traditional methods like passwords and PINs are foundational [2], they lack adaptability. Existing behavioral systems [4] provide limited improvements, but SecureTrack stands out by integrating multi-sensor data and real-time adaptability, offering advanced false positive minimization and enhanced usability.

Feature	Traditional Security	Existing Systems	SecureTrack
Real-time Adaptivity	None	Limited [4]	Full
Multi-sensor Integration	None	Partial [4]	Full
False Positive Minimization	Minimal [2]	Basic [4]	Advanced
Usability	High [2]	Moderate [4]	Low
Compliance with GDPR/HIPAA	Partial [2]	Partial [4]	Full

Figure 1: Table: Comparative Analysis of SecureTrack

3 OVERVIEW OF THE DESIGN

The target of the *SecureTrack* is to ensure the user is the owner of the application or phone. If *SecureTrack* senses that the user is not the owner, it will ask for further authentication, like biometric authentication. This validation procedure is like the immune system in our body. The immune cells migrate everywhere in our body to detect substances in our body. If the substance does not belong to the body, it will destroy it. [6]. Therefore, in our context, our first mission is to determine self-actions and non-self-actions. To target non-self-actions, the *SecureTrack* first collects the user behavior and the context of the environment as the database. Then, the *SecureTrack* determines the other user and the owner based on this database. The flow of the procedure is shown in Fig. 2. As a result, there are three phases in *SecureTrack*: data collection, determination, and authentication.

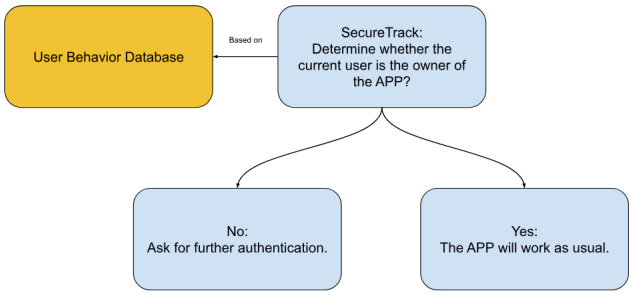

Figure 2: Shows the working procedure of *SecureTrack*. The *SecureTrack* based on the user behavior database to determine whether the current user is the owner of the phone.

3.1 Data collection

In the *A Sense of Self for UNIX Processes* [5], it collects the short sequence of the privileged system call as the database. However, in our project, we plan to implement *SecureTrack* on the app level, so it is hard to access system calls. Therefore, since the phone embeds with many sensors, we are going to leverage this advantage. We analyzed several ways to track user behavior and build the database so we could decide which to use in the implementation phase.

3.1.1 Camera. We could capture the current user every second and do facial recognition to determine whether the current user is the owner of the phone. However, the camera is a battery-consuming APP, and most users may not accept monitoring every second. Therefore, this may not be the top choice. Nevertheless, it could be used to validate our decision due to its high accuracy. To be more specific, if *SecureTrack* detects that the user is not the owner based on other sensors, it could open the camera and determine whether this is a false positive. Since we only open the camera when needed, it would not consume too much battery. In addition, the camera is opened in a very short time, 1 second, so *SecureTrack* could promise privacy.

3.1.2 Finger size. The finger size is unique to everybody. In addition, with different angles when pressing the touch screen, the size of the area may vary from person to person. Therefore, we could store the owner's touching area. If the size of the touching area changes rapidly, the *SecureTrack* raises a different user alarm. However, this may be easily spoofed.

3.1.3 The path of interacting with the touch screen. The gesture of doing a thing may vary from person to person. For example, if users are asked to drag from top to down to display the notification center in an Android phone, people with small hands will slide the screen further to the right compared to people with large hands. Therefore, we could record the owner's gesture to achieve a specific feature. If the *SecureTrack* detects different gestures, it could raise the alarm.

3.1.4 Location. Mostly, people go to the same places every day. For example, as an ASU student, he/she may go to school, dorm, library, and gym. In addition, the order of places may be similar day by day. For example, the student may start their day at the dorm, go to school, go to the library after class, and then go to the gym to call the day. Therefore, we could trace the user's location; if the user is at a place where he/she has not been before, the alarm would work.

During the implementation phase, we could leverage the above sensors to record user behavior based on the sensors' advantages and disadvantages. Specifically, we could separate it into layers and use different sensors during our decision procedure. For example, we could do the majority vote in finger size, the path of interacting with the touch screen, and the location. If they raise the alarm, *SecureTrack* would activate the camera to do a double-check. However, if the *SecureTrack* detects the location where the owner has never been, it will directly activate classical authentication.

3.2 Determination

The *SecureTrack* needs time to collect the owner's behavior, so it will ask for authentication in the first few days after the user gets the new phone. This may decrease the user experiment. To maximize the user experience, we can restrict to only trigger authentication when the user starts to use the phone. Before the user closes the screen, it will not ask for authentication again. However, it may cause false negatives if the owner passes his/her phone to his/her

friend. To avoid this, we can use the accelerometer to detect the movement. If the value of the accelerometer changes strongly, since this motion may indicate the owner passed his/her phone to his friend, it will trigger the authentication.

To improve the correctness of the modal, we need to filter out the extreme value during the data collection phase since it may come from another user when the owner passes his/her phone to his/her friend. Another way to improve it is that we could validate the collected value with the user. We could raise the authentication required sometimes. If the authentication modal returns that it is the owner, we can mark the collected value as the owner and store it in our database. During detection, the sensors collect the current values and compare them with the behavior database. If the value difference is over the threshold, the alarm will be raised. Since we only do the value comparison, it does not cost many computation resources.

3.3 Authentication

If the *SecureTrack* senses that the user is not the owner, it will ask the user to prove that he/she is the owner. The user would be asked to log in to the phone again using facial recognition or fingerprints provided by the phone's OS. If the user cannot provide that he/she is the owner, the *SecureTrack* will lock the phone.

3.4 *Question: Is the solution described technically sound?*

Our opinion, it is technically sound. The core feature of *SecureTrack* is to record user behavior. We propose using several sensors that are equipped in smartphones nowadays to record user behavior so our proposal is workable based on the techniques now.

4 DESIGN AND METHODOLOGY

This section outlines the design principles and methodology behind SecureTrack, focusing on data security, regulatory compliance, and resource efficiency.

4.1 Design Considerations

The system design aims to balance technical feasibility, user experience, and robust security measures. The architecture consists of multiple layers, each addressing a specific function.

The **data collection layer** securely captures user interaction data through APIs and SDKs integrated into the application. AES-256 encryption is used for data storage, and TLS 1.3 is used for secure transmission. These mechanisms ensure privacy and security compliance [5]. The **behavioral analytics engine** processes the collected data to create and update unique behavioral profiles. This engine utilizes scalable cloud computing resources for efficient data processing, employing machine learning models to detect patterns and establish behavioral baselines [6].

The **anomaly detection module** operates in real-time, continuously monitoring incoming data against stored behavioral profiles. Distributed computing techniques ensure system responsiveness under large-scale data processing demands. Detected anomalies trigger the **security enforcement layer**, which integrates seamlessly with authentication systems, including biometric verification and token-based mechanisms. Dynamic security measures are tailored based on anomaly severity, reducing user friction while maintaining robust protection [1].

To optimize battery life, intermittent monitoring strategies are implemented. Sensors activate only during high-risk periods, reducing energy consumption while preserving system performance. Lightweight machine learning models and edge computing further enhance efficiency [7].

Battery optimization techniques, including intermittent monitoring and on-device processing, address potential resource constraints.

4.2 Methodology

To achieve a robust and adaptive security system, SecureTrack employs a methodology integrating behavioral analytics with real-time anomaly detection.

Data Collection and Behavioral Profiling (AD-4-2) User interaction data such as login times, geolocations, device types, and session durations are continuously collected. Privacy is ensured through data minimization and anonymization techniques, which limit the exposure of personally identifiable information (PII). Encryption using AES-256 for storage and TLS 1.3 for transmission secures data throughout its lifecycle [5]. This addresses privacy concerns by providing detailed strategies for safeguarding sensitive data during continuous monitoring.

Regulatory Compliance Implementation (IS-4-4) SecureTrack ensures compliance with GDPR, HIPAA, and CCPA through a robust consent management framework, enabling users to control data sharing and access. All data access is logged and periodically audited to ensure accountability. Privacy notices within the app inform users about data usage, reinforcing transparency and trust [7].Detailed mechanisms for enforcing compliance, including audits and user consent management, have been clarified.

Real-Time Anomaly Detection and Security Responses Anomalies are detected by comparing ongoing interactions against behavioral profiles using advanced machine learning techniques, such as decision trees and k-means clustering. Detected anomalies trigger dynamic security responses, including re-authentication or session termination, based on their severity [1]. The system integrates privacy safeguards, regulatory compliance, and battery-efficient monitoring to ensure a secure and user-friendly mobile solution.

5 EVALUATION

SecureTrack leverages behavioral analytics to enhance app security, addressing static vulnerabilities by adapting protocols to user behavior. Our 5-year phased strategy focuses on innovation, compliance, and usability through research, development, testing, and scaling.

5.1 5-Year Plan

SecureTrack's implementation follows a phased approach to ensure robust deployment and widespread adoption.

(1) **Year 1: Research, Feasibility, and Prototype Development**
Objective: Establish SecureTrack with foundational research, feasibility studies, and a working prototype.

- **Market Research & Needs Assessment:** Identify user pain points and analyze existing solutions. Conduct 50 user interviews and analyze 5 existing security solutions.
- **Privacy and Regulatory Compliance:** Ensure compliance with data privacy laws and secure data handling protocols.
- **Behavioral Analytics Framework:** Develop and test anomaly detection algorithms in a prototype.
- **Early Prototyping and Feedback:** Build a prototype to gather user feedback for iterative improvements. Test with 20 users and iterate based on feedback.
- **Security Framework Design:** Implement security measures based on behavioral deviations.

(2) **Year 2: MVP Development and Testing**
Objective: Create a Minimum Viable Product (MVP) with core features and real-time anomaly detection.
- **MVP Features:** Develop behavior tracking, anomaly detection, and dynamic security features.
- **Security Tuning:** Refine models to reduce false positives.
- **User Testing and Feedback:** Conduct pilot testing to refine usability and security.
- **Scalability Testing:** Ensure performance under increasing user loads. Validate handling of 10,000 concurrent users.

(3) **Year 3: Full App Development and Integration**
Objective: Launch the app with advanced analytics and third-party integration.
- **Advanced Behavioral Analytics:** Enhance machine learning models for better accuracy. Achieve 90% detection accuracy with 20% fewer false positives.
- **Cloud Integration and Security:** Ensure secure cloud data storage and consent-based data sharing.
- **Beta Testing:** Expand user testing with real-world conditions. Test with 500 beta users across regions.
- **UI Enhancements:** Improve the interface and add personalized feedback for security actions.

(4) **Year 4: Public Launch and Marketing**
Objective: Release SecureTrack with a focus on adoption and performance.
- **Optimization:** Fix bugs, optimize performance, and conduct penetration testing.
- **Marketing Campaign:** Highlight behavioral security features in a broad launch campaign. Achieve 100,000 downloads in 6 months.
- **Post-launch Support:** Provide updates and responsive user support.

(5) **Year 5: Expansion and Advanced Features**
Objective: Broaden integration, geographic reach, and introduce advanced security.
- **Health Platform Integration:** Link with more health platforms for richer profiles.
- **Global Expansion:** Enable multi-language support and comply with regional regulations.
- **Advanced Security:** Add biometric and risk-based authentication. Use blockchain for secure user interaction audit trails.

Phase	Key Risks	Risk Impacts	Mitigation Strategy
Year 1	Incomplete feedback, regulatory issues	Delayed Prototype, regulatory delays	Collaborate with legal experts
Year 2	Inaccurate detection, MVP feature bloat	False positives, delayed release	Refine models, prioritize MVP features
Year 3	Cloud integration, scalability issues	Poor performance, reduced engagement	Partner with experts, diverse testing
Year 4	Poor launch marketing, security breach	Low adoption, data vulnerability	Targeted campaign, rigorous pentesting
Year 5	Compliance issues, engagement decline	Slow expansion, declining interest	Refine models, introduce gamification

Figure 3: Table: Risk Analysis and Mitigation

5.2 *Why will this take 5 years?*

The 5-year plan ensures robust machine learning models, GDPR/HIPAA compliance, extensive testing, and resolution of technical, legal, and scalability challenges.

6 CONCLUSION

SecureTrack represents a significant step forward in addressing the critical challenges of mobile security for health applications. By leveraging behavioral analytics and machine learning, we have developed a dynamic and adaptive security framework that not only enhances user protection but also maintains a seamless user experience. This approach bridges the gap between traditional static security measures and the evolving needs of mobile environments, ensuring compliance with privacy regulations such as GDPR, HIPAA, and CCPA.

The development of SecureTrack has been both an intellectually stimulating and rewarding endeavor. While the framework addresses many current vulnerabilities, we acknowledge opportunities for further refinement, particularly in areas like optimizing battery efficiency and ensuring user trust through enhanced privacy safeguards.

With a structured five-year plan, SecureTrack is poised to evolve into a scalable and versatile solution that adapts to future technological advancements and user expectations. This project underscores the importance of innovative, user-centric design in tackling modern security challenges and sets a strong foundation for future advancements in mobile health application security.

REFERENCES

[1] B. Aljedaani and M. A. Babar. 2021. Challenges With Developing Secure Mobile Health Applications: Systematic Review. *JMIR Mhealth Uhealth* 9, 6 (Jun 2021), e15654. https://doi.org/10.2196/15654

[2] Aaron Henricks and Houssain Kettani. 2020. On Data Protection Using Multi-Factor Authentication. *Proceedings of the 2019 International Conference on Information System and System Management (ISSM 2019)* 1 (2020), 1–4. https://doi.org/10.1145/3394788.3394789

[3] HIPAA Journal. 2024. Healthcare Data Breach Statistics. (2024). https://www.hipaajournal.com/healthcare-data-breach-statistics/#:~:text=2021%20was%20a%20bad%20year,stolen%2C%20or%20otherwise%20impermissibly%20disclosed Accessed: 2024-11-30.

[4] Maria Papaioannou, Georgios Mantas, Emmanouil Manos Panaousis, Aliyah Essop, Jonathan Rodriguez, and Victor Sucasas. 2023. Behavioral Biometrics for Mobile User Authentication: Benefits and Limitations. In *2023 IFIP Networking Conference (IFIP Networking)*. 1–6. https://doi.org/10.23919/IFIPNetworking57963.2023.10186419

[5] A. Somayaji T.A. Longstaff S. Forrest, S.A. Hofmeyr. 1996. A sense of self for Unix processes. https://ieeexplore.ieee.org/document/502675. (1996).

[6] Lauren Sompayrac. 2016. *How The Immune System Works*. Vol. 1-12. Wiley Blackwell.

[7] Varun Tiwari, Deepika Kirti, Ruchi Sawhney, Santosh Kumar Singh, Vikas Rao Vadi, Swati Sharma, Hari Mohan Jain, and Jatinder Kaur. 2024. Addressing the Major Challenges and Issues in Mobile Cloud Computing. *African Journal of Biological Sciences* 6, Si4 (2024), 1178–1186. https://doi.org/10.48047/AFJBS.6.Si4.2024.1178-1186