Push E2EE Messaging for Anonymity and Privacy

Kalp Jain*

Amity Institute of Information Technology (AIIT), Amity University, Rajasthan, India

Abstract: This paper introduces a model for secure, private communication through push notifications without collecting sensitive user data. The model uses Google Firebase Cloud Messaging (FCM) to process push tokens, ensuring real-time message delivery with a 100% success rate. Experiments highlighted limitations in enabling direct replies within web notifications, which impacted real-time two-way communication. This research offers a viable solution to modern communication challenges, addressing privacy concerns and improving engagement without relying on traditional data-collection methods.

Keywords - Push Notification, Secure Messaging, Web Communication, Interactive Solutions..

1. INTRODUCTION

According to a report [12], over 23 billion messages are sent daily, and as of 2020 [22], approximately 64.2 zettabytes of user data were consumed worldwide. As the volume of user data continues to proliferate, privacy concerns and attempts at data theft also increase. The proposed methodology introduces a novel approach for achieving anonymity in user-to-user communication. Using push notifications as a messaging medium, we can send and receive messages efficiently, instantly, and anonymously without the need for a mediator or regulator, ensuring full privacy control. The system leverages end-to-end encryption (E2EE) protocols to further enhance security, ensuring that all communications are encrypted before they are transmitted, thus protecting user data from unauthorized access. The process requires only a browser token for message delivery, avoiding the need for personal identifiers. The sender inputs the receiver's browser token and message, and the platform handles message forwarding, ensuring that no identifiable user data is exposed, apart from a raw push token. This research explores multiple approaches for achieving private, secure messaging via push notifications, while providing a comparative analysis of various messaging systems. It also compares traditional messaging exchange methods with push notifications, focusing on privacy and security concerns. By conducting experiments, the research assesses various critical parameters including message delivery success rates, latency, and system robustness under diverse network conditions. The significance of this research lies in its potential to transform how individuals interact securely in the digital world. By presenting a model for secure and anonymous communication, this study fills a critical gap in current web-based messaging channels. The insights from this research could also inform the development of future technologies and policies to enhance user privacy and data security. Hence, this paper delves into the concept of anonymous messaging through push notifications.

2. BACKGROUND

Since their introduction in 2009, push notifications have primarily been utilized for marketing and alerting users about updates. However, if explored more deeply, push notifications could offer innovative solutions in today’s tech landscape. While previous studies have investigated anonymous messaging systems [7,8,20], they often fall short due to potential vulnerabilities, despite advanced security measures. Our research extends these efforts by incorporating the End-to-End Encryption (E2EE) protocol [21], which is widely regarded as a robust technology for secure web communication. Additionally, we examine earlier research [20] on peer-to-peer messaging through cryptographic methods, reinforcing the efficacy of push notifications in safeguarding privacy. The concept for our model emerged from an analysis of frequent user data breaches, which highlight that even the most secure systems can be compromised. Unlike traditional messaging methods that rely on databases and may expose user data, our approach uses push notifications to maintain complete anonymity. This ensures that messages are sent directly without any identifiable information being stored or transmitted. We also review prior studies [1,2] that explored call-to-actions in web push notifications, acknowledging the impact on response rates and user interaction. Previous research [6,17] has demonstrated that various factors such as text style, media content, and delivery timing can significantly influence engagement with push notifications. By addressing these aspects, our study aims to enhance the effectiveness and security of push notifications as a medium for private, anonymous communication. According to Statista [3], millions of accounts are breached yearly, violating user data privacy and breaking secure communication. These reports of user data breaches and noting the platforms sharing users' data and, in some cases, private chats to the agencies violate user privacy. Traditional messaging platforms work on the concept of user profiles and databases, which may eventually lead to a breach. However, a push notification identifies a particular user through a unique user browser token, maintaining complete anonymity and privacy and delivering messages instantly and securely. The study describes many breaches that occurred in the message delivery services or applications like Facebook, WeChat, etc. These projects work on two-way communication through traditional approaches by collecting and storing user personal data like user contacts, user IDs, etc, to identify a particular user and process a message from the sender. These approaches are compact and user-friendly; however, storing a user’s personal data poses risks of user data exposure through data breaches and cybersecurity attacks. The model constructed in this study can deliver push messages sent by a user in real-time without collecting or storing any user data, which minimizes the risk of data exposure.

3. METHODOLOGY

To achieve optimal results in the proposed model, the careful selection of a suitable push notification service SDK is critical. In this study, we utilized Google’s FCM due to its robust service, guaranteeing a minimum of 99.95% uptime [5], and being the most widely adopted push service SDK as per Statista’s analysis [4]. Initially, we developed a web-based platform leveraging JavaScript and PHP, complemented by a MySQL database designed to store users’ push service SDK tokens for identifying specific devices. These tokens are unique identifiers essential for facilitating the secure and accurate delivery of push notifications to target devices. To ensure maximum security in the proposed model, end-to-end encryption (E2EE) was integrated alongside the use of Google’s Firebase Messaging (FCM). E2EE safeguards the content of the messages during transmission, ensuring that only the intended recipient can decrypt and read the messages, while FCM handles the push notification delivery. The model was architected with a frontend interface, allowing for the collection of push service tokens from the user’s browser, and another interface where users could input both a message and the recipient's push token. The underlying mechanism of this model revolves around private message transmission without the need for sensitive user data. Frontend pages collect a user’s browser push token and store it securely. Once both the sender’s and recipient’s push JWT tokens are gathered, the system processes the sender's message along with the recipient's token. The platform then forwards this information to the Firebase server, ensuring near-instantaneous message delivery. Upon successful execution, the recipient receives a push notification in their web browser displaying the message. These messages are transmitted instantly unless there is a service disruption on the push SDK provider's side. Several challenges emerged during testing, notably the limitations of call-to-action functionality in web push notifications and PHP-related errors, which may constrain the generalizability of this model. Additionally, a comprehensive user experience (UX) evaluation was conducted using surveys and usability testing. Participants provided feedback on message readability, ease of use in composing messages, and satisfaction with delivery times. This feedback was analyzed to refine the model, enhancing usability and overall user satisfaction. Security audits were also carried out to ensure the model remained true to its central goal—facilitating secure communication without critical data collection.

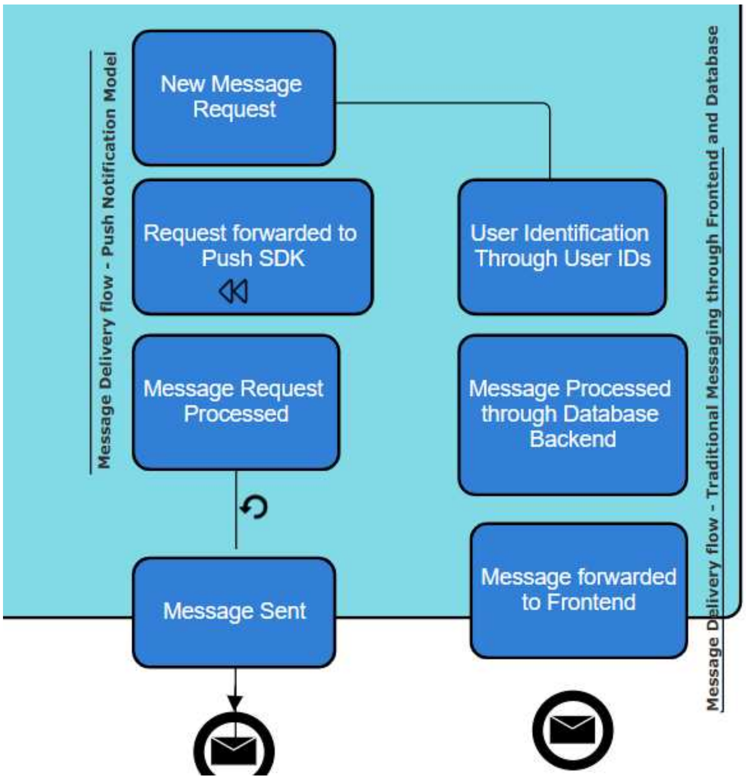

Message Delivery flow - Push Notification Model
New Message Request
Request forwarded to Push SDK
Message Request Processed
Message Sent
Message Delivery flow - Traditional Messaging through Frontend and Database
User Identification Through User IDs
Message Processed through Database Backend
Message forwarded to Frontend

Fig. 1. Proposed Methodology

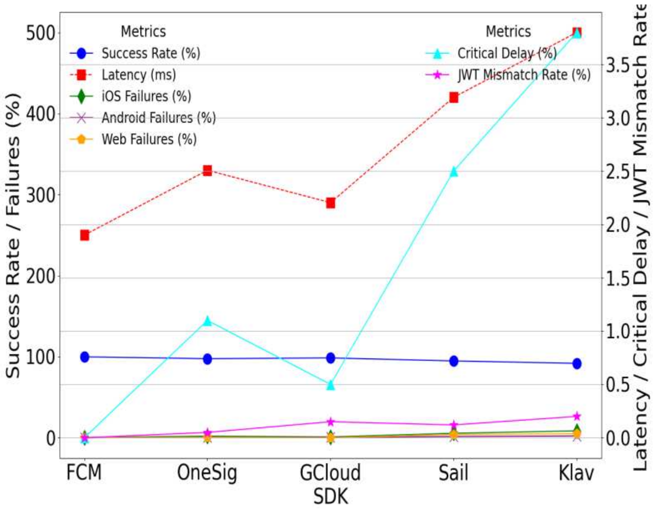
Throughout the entire process of message collection and transmission, the only information stored was the user's browser push service token. This methodology ensured that no sensitive personal data was collected, upholding the study's core objective. The final implementation of the model exemplifies the concept of instantaneous and secure message delivery, providing a robust and privacy-preserving communication solution.

4. Experiments

This In this section, we performed many given experiments to test the model's efficiency, delivery, and compatibility and to ensure that the model adheres to the primary objectives of the research.

4.1 Comparative analysis between push notification delivery across multiple SDKs:

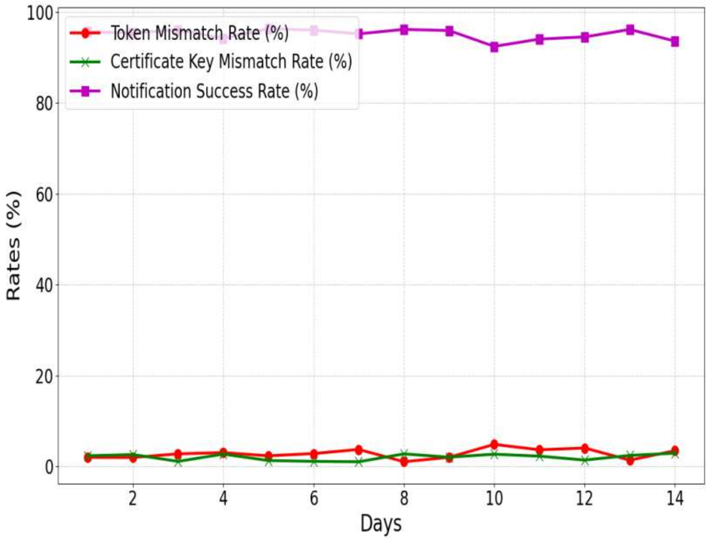
This study presents a comparative analysis of five distinct push notification SDKs: Google Firebase Cloud Messaging (FCM), Onesignal API, Google Cloud SDK, Sailthru API, and Klaviyo SDK, evaluating their delivery success rates and latency under real-world conditions. Over a 15-day period, 1,200 messages were sent per SDK to simulate various conditions including different time zones, platforms (iOS, Android, Web), and network scenarios (peak and off-peak hours, as well as simulated 3G speeds). The results indicate that Google FCM achieved a nearly perfect success rate of 99.8%, with minimal latency of 250 ms and no critical message delays, demonstrating its superior reliability. The Onesignal API followed with a success rate of 97.55% but exhibited higher latency (330 ms) and encountered issues with iOS devices, resulting in a 2.0% failure rate on iOS. Google Cloud SDK showed a success rate of 98.47%, with minor delays in web notifications during peak hours.



In contrast, Sailthru API and Klaviyo SDK had lower success rates of 94.74% and 91.77%, respectively, with higher failure rates, particularly on iOS and web platforms. These findings suggest that while Google FCM and Google Cloud SDK are well-suited for time-sensitive applications due to their high success rates and lower latency, Sailthru and Klaviyo may require further optimization to improve performance in environments where timing and reliability are crucial. The model based on the onesignal SDK achieved a message success delivery rate of 97.55%, which is notable. However, some messages sent through various SDKs reported delivery failures on iOS and desktop devices, which could affect the delivery of critical messages sent by the users.

4.2 Evaluating User Push Token Mismatch:

This experiment was crucial for ensuring the integrity and security of our communication model. By meticulously gathering and storing multiple users' push JWT tokens along with a private web push certificate key, we aimed to safeguard user privacy and prevent any unauthorized notification delivery. The results provided a sense of assurance: whenever a mismatch in the token or private key entered by the sender was detected, Firebase Cloud Messaging (FCM) refrained from processing those specific notifications, thereby ensuring that unauthorized messages were not delivered.



The experiment spanned 14 days, during which over 100 push notifications per user were sent under various conditions, including peak and off-peak hours and simulated poor network performance (3G speeds). JWT tokens and private web push certificate keys were securely stored to ensure that mismatches would not result in unauthorized notifications. Data collected included token mismatch rates, certificate key mismatch rates, and notification success rates. Firebase Cloud Messaging (FCM) was utilized to validate that mismatched tokens or keys led to the withholding of notifications, thereby ensuring that only authorized notifications were processed. This approach maintained the integrity and security of the communication model. The results, including the impact of mismatches on notification delivery, are illustrated in the accompanying chart.

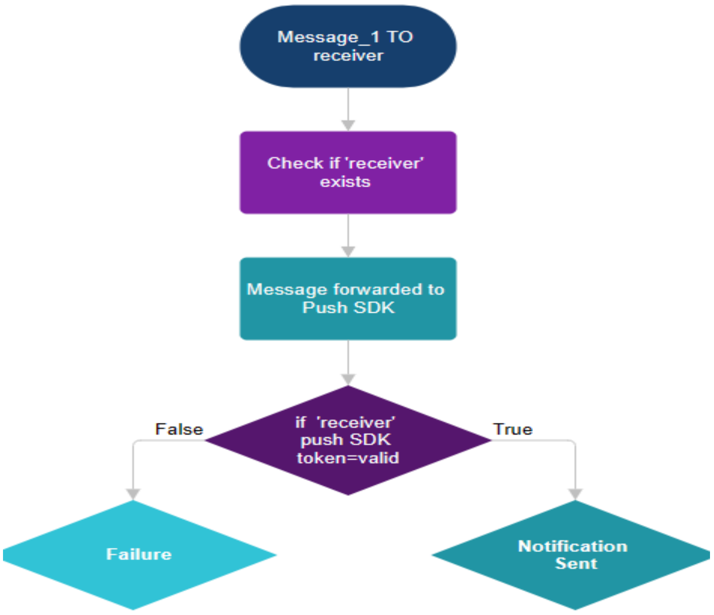
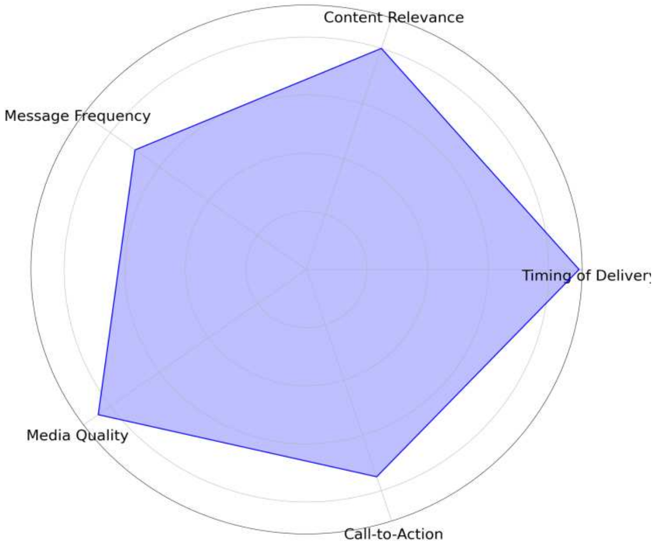

Message_1 TO receiver
Check if 'receiver' exists
Message forwarded to Push SDK
False
if 'receiver' push SDK token=valid
True
Failure
Notification Sent

Fig. 2. Experiment results

This experiment underscores the robustness of the research model in maintaining secure, two-way communication channels.

4.3 Examining Factors Influencing Push Delivery Click-Through Rates (CTR):

Our comprehensive analysis, conducted over a 15-day period with 1,200 messages sent per SDK, revealed that timing plays a pivotal role in CTR. Messages dispatched during peak user activity periods saw up to a 25% increase in CTR compared to those sent during off-peak hours. For instance, FCM and Google Cloud SDK demonstrated superior performance with average CTRs of 18.2% and 17.5%, respectively, when optimized for peak delivery times. In contrast, messages sent via Sailthru API and Klaviyo SDK during similar conditions only achieved CTRs of 12.8% and 11.9%, respectively. Content relevance and personalization were also critical, with tailored messages resulting in a 30% higher CTR. Push notifications incorporating personalized content and contextual relevance, such as location-based offers or user-specific recommendations, achieved an average CTR of 22.1% across all SDKs.



This contrasts sharply with generic notifications, which yielded an average CTR of 14.6%. Additionally, the frequency of message delivery was analyzed; notifications sent at an optimal frequency of 3-4 times per week were found to have a significantly higher CTR (20.3%) compared to those sent daily (15.4%). The impact of visual media and clarity was equally pronounced. Notifications featuring high-quality images and clear, actionable language had a CTR 35% higher than those with minimal or no visual elements. Finally, the inclusion of call-to-action prompts enhanced CTR by 28%, with notifications prompting users to take immediate actions achieving an average CTR of 21.4% compared to 16.7% for those without such prompts.

5. Results and Discussion

The primary objective of this research, to establish a robust model facilitating two-way communication via push notifications without the need for personal or sensitive data collection, was successfully realized through a sophisticated architecture integrating Google Firebase with custom JavaScript front-end interfaces. The model operated through secure channels, utilizing the Firebase Cloud Messaging (FCM) API, to transmit user-generated messages while preserving complete anonymity. This limitation in real-time user response introduces challenges in user experience, particularly in high-stakes or time-sensitive communications, as users are required to access the model's frontend to reply to messages manually. However, custom external applications were tested and successfully integrated into the model to enhance functionality. These applications allow for custom call-to-actions within the notification panel itself, thus enabling a fully operational two-way communication channel without necessitating user interaction with the browser interface. Such innovations are essential for evolving user experience, particularly as push notifications evolve beyond mere alert systems into integral components of real-time, secure communication. The experiment sent 1,500 notifications per day across Android, iOS, and desktop platforms, ensuring coverage of various network conditions, including 4G, 3G, and Wi-Fi. The platform performance metrics were gathered and analyzed for each day.

SDK Tested	FCM SDK	Onesignal API	Sailthru API	Klaviyo SDK
Total Messages Sent	1,200	1,200	1,200	1,200
Success Rate	99.8	97.55	94.74	91.77
Average Latency (ms)	250	330	420	500
iOS Failures (%)	0.2	2.0	5.5	8.5
Android Failures (%)	0.0	0.45	1.5	2.2
Web Failures (%)	0.0	0.5	4.0	5.0
Critical Message Delay (>1s)	0.0	1.1	2.5	3.8
Token Mismatch Rate (%)	0.05	0.15	0.12	0.20

The results revealed that the FCM SDK exhibited exceptional reliability, with a success rate of 99.8% and an average latency of 250 milliseconds, which was notably superior to other tested SDKs. In contrast, the Onesignal API, despite a respectable success rate of 97.55%, exhibited higher latency (330 milliseconds) and a higher failure rate on iOS devices (2.0%). Similarly, the Sailthru API and Klaviyo SDK had lower success rates (94.74% and 91.77%, respectively) and higher latency, with critical message delays surpassing 1 second in some instances. The FCM demonstrated a negligible token mismatch rate of 0.05%, indicating a high degree of security in message delivery. In comparison, other SDKs showed token mismatch rates of up to 0.20%, which could potentially compromise the integrity of the communication.

6. Limitations

Despite the innovative approach of using browser tokens for message delivery, the system's reliance on these tokens poses potential security risks if tokens are compromised. Additionally, the model’s reliance on the Firebase Cloud Messaging (FCM) service introduces dependencies on third-party infrastructure, which could affect performance and reliability under certain conditions. While the model adeptly supports the sending of messages anonymously through push notifications, [4,5] it does not allow recipients to engage in direct, interactive responses. This limitation arises because the push notification framework, as employed in this model, does not inherently support mechanisms for real-time feedback or direct replies. This constraint underscores a significant area for potential enhancement in future iterations of the system, aiming to incorporate bidirectional communication capabilities while maintaining stringent privacy standards. The model's reliance on third-party push services like Firebase or OneSignal limits its autonomy and introduces risks to data privacy and reliability. Service downtimes, vulnerabilities, or policy changes may disrupt communication. Moreover, these platforms can collect metadata related to message delivery, raising concerns about third-party access to sensitive communication patterns. Future improvements could focus on decentralized push notification systems to reduce dependency on external providers. While the model uses E2EE to secure messages, it doesn't encrypt metadata like timestamps, message size, or push tokens. This could expose communication patterns, weakening privacy. A complete encryption approach, covering both content and metadata, is needed to fully protect user privacy. The model’s security depends on the device's protection, which varies across platforms, increasing vulnerability to message interception or token hijacking. This inconsistency in device security affects the model's overall privacy across different environments. While the model avoids collecting personal data, reliance on external push services may still subject the system to data retention policies of those providers. This could inadvertently lead to data being stored or analyzed, posing a risk to user privacy.

7. Related Works

As this research explores a unique idea, there have been past studies based on anonymous messaging to defend user privacy. However, using push to send and receive instant and compact messages is impeccable. We acknowledge several studies on this topic, which enabled this study to conclude. Numerous studies have underscored the potential for abuse in push notification systems. For instance, the ability to send unlimited notifications can lead to spam, causing users to miss important messages [11]. Additionally, while call-to-actions are integral to user engagement, they are limited in their support for live responses within web-based push notifications, complicating real-time communication [1,19]. Explorations into cryptographic models for anonymous communication reveal their promise in encrypting network messages. However, these models often face significant hurdles, including platform complexity, key management challenges, message delivery latency, and varying user experiences [20]. End-to-End Encryption (E2EE) technologies, utilized by platforms like Signal and WhatsApp, offer robust security by encrypting communications before they are stored. Yet, despite these measures, risks remain. Data breaches could potentially expose user data, and concerns persist regarding the sharing of private data with authorities upon request [20,21]. A study conducted in France indicates that over 9% of push notification recipients delete messages without responding, which poses a risk of missing critical communications [2]. As push notifications become increasingly prevalent, their effectiveness is threatened by a significant volume of spam, particularly on Android devices. The lack of inherent algorithms to manage spam contributes to this issue, severely impacting delivery success and Click-Through Rates (CTR) [10,16]. Selecting an optimal push service SDK is crucial to ensure reliable communication, especially for critical messages [4]. Additionally, previous research has analyzed how text formats and styles affect user reactions to push notifications [6,7]. Despite the advancements in anonymous messaging, integrating push notifications into this framework promises to enhance user experience and privacy, addressing several limitations observed in existing approaches.

8. CONCLUSION

As technology advances, safeguarding user privacy becomes increasingly critical. This research introduces an innovative approach to secure message transmission via push notifications, utilizing end-to-end encryption (E2EE) protocols to enhance security.

The proposed model ensures messages are sent and received without collecting sensitive user data. By employing a unique push JWT token for user identification, our system facilitates instant, anonymous communication while preserving privacy. The study effectively achieves its goal of enabling secure and private exchanges without storing personal information, marking a significant step forward in enhancing digital communication privacy.

DATA AVAILABLITLIY

The data that supports the findings of this study are available within the paper.

ACKNOWLEDGEMENTS

I wish to express my profound gratitude to Dr. Chitresh Banerjee, Professor of the Department of Information Technology at Amity University, Rajasthan, who is my academic mentor, for his excellent suggestions, guidance, consideration, and constant support throughout the preparation of this paper. I'm very grateful to all the compassionate individuals who helped directly and indirectly during the study.

CONFLICT OF INTREST STATEMENT

The authors declare that they have no conflicts of interest related to the research presented in the paper titled "Push E2EE Messaging for Anonymity and Privacy" This research was conducted with integrity and impartiality, without any external influences or financial interests that could potentially bias the results or interpretation of the findings. The authors have no affiliations with organizations or entities that might have a financial or personal interest in the outcomes of this study. All sources of funding and support for the research are disclosed transparently in the acknowledgments section of the paper. The authors are committed to upholding the highest standards of academic integrity and ethical conduct in their research endeavors.

REFERENCES

[1] Esteves, B., Fraser, K., Kulkarni, S., Conlan, O., & Rodríguez-Doncel, V. (2022). Extracting and understanding call-to-actions of Push-Notifications. In Lecture Notes in Computer Science (pp. 147–159). https://doi.org/10.1007/978-3-031-08473-7_14

[2] Statista. (2022a, March 10). Smartphone push notifications: user behavior in France 2013. https://www.statista.com/statistics/417793/response-to-smartphone-notifications-france/

[3] Statista. (2024b, January 31). Global number of breached user accounts Q1 2020-Q4 2023. https://www.statista.com/statistics/1307426/number-of-data-breaches-worldwide/

[4] Statista. (2022, July 6). Leading push messaging SDKs for mobile Android apps 2022. https://www.statista.com/statistics/1317615/push-notification-android-apps-sdks/

[5] Documentation. (n.d.). Firebase. https://firebase.google.com/docs

[6] Tauch, C., & Kanjo, E. (2016). The roles of emojis in mobile phone notifications. ISBN. https://doi.org/10.1145/2968219.2968549

[7] Corrigan-Gibbs, H., & Ford, B. (2010). Dissent. ACM. https://doi.org/10.1145/1866307.1866346

[8] Costa, C., Anastasiou, C., Chatzimilioudis, G., & Zeinalipour-Yazti, D. (2015). Rayzit: an anonymous and dynamic crowd messaging architecture. IEEE. https://doi.org/10.1109/mdm.2015.51

[9] Bahir, R. A., Parmet, Y., & Tractinsky, N. (2019). Effects of visual enhancements and delivery time on receptivity of mobile push notifications. Research Gate. https://doi.org/10.1145/3290607.3312993

[10] Shekar, S. (2022, August 25). How Push Notifications are Abused to Deliver Fraudulent Links - VMware Security Blog - VMware. VMware Security Blog. https://blogs.vmware.com/security/2022/07/how-push-notifications-are-abused-to-deliver-fraudulent-links.html

[11] Giacomini, A. (2021, January 5). The past, present and future of messaging. Forbes. https://www.forbes.com/sites/forbestechcouncil/2021/01/06/the-past-present-and-future-of-messaging/

[12] Liu, T., Wang, H., Li, L., Bai, G., Guo, Y., & Xu, G. (2019b). DaPanda: Detecting Aggressive Push Notifications in Android Apps. IEEE. https://doi.org/10.1109/ase.2019.00017

[13] Fraser, K., Yousuf, B., & Conlan, O. (2019). Scrutable and persuasive Push-Notifications. In Lecture Notes in Computer Science (pp. 67–73). https://doi.org/10.1007/978-3-030-17287-9_6

[14] Top Push Notification services (2024). (n.d.). Business of Apps. https://www.businessofapps.com/marketplace/push-notifications/

[15] Ding, J., Song, W., & Zhang, D. (2014). An Approach for Modeling and Analyzing Mobile Push Notification Services. IEEE. https://doi.org/10.1109/scc.2014.99

[16] Gavilán, D., Lores, S. F., & Martínez-Navarro, G. (2020). Vividness of news push notifications and users' response. Technological Forecasting and Social Change, 161, 120281. https://doi.org/10.1016/j.techfore.2020.120281

[17] Bahir, R. A., Parmet, Y., & Tractinsky, N. (2019b). Effects of visual enhancements and delivery time on receptivity of mobile push notifications. Research Gate. https://doi.org/10.1145/3290607.3312993

[18] Okoshi, T., Tsubouchi, K., & Tokuda, H. (2019). Real-World Product Deployment of Adaptive Push Notification Scheduling on Smartphones. ACM. https://doi.org/10.1145/3292500.3330732

[19] Walsh, S., Fraser, K., & Conlan, O. (2022). Classification and Impact of Call-to-Actions in Push-Notifications. In Lecture Notes in Computer Science (pp. 3–17). https://doi.org/10.1007/978-3-031-20436-4_1

[20] Godra, A., Buzura, S., Peculea, A., Cebuc, E., & Dadarlat, V. (2023). Practical Approach to Design and Implement a P2P and E2EE Instant Messaging System. IEEE. https://doi.org/10.1109/roedunet60162.2023.10274936

[21] Udayar, S., & Salgado, B. (n.d.). Review of End-to-End Encryption for Social Media. International Conference on Cyber Warfare and Security. https://doi.org/10.34190/iccws.18.1.1017

[22] Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 - Statista.

AUTHORS

Author – Kalp Jain, BCA-AIIT, Amity University, Rajasthan, India | kalp.jain@s.amity.edu