Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Implementation of a Hybrid Cryptosystem Using ChaCha20 and ECC for Image Encryption in an Android Application

Samuel Anaya Putra Zai[1)*], Debi Yandra Niska[2)], Zulfahmi Indra[3)], Kana Saputra[4)], Adidtya Perdana[5)]
[1,2,3,4,5)] Ilmu Komputer, Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia
[1)]samuelanayaputra.zai@gmail.com, [2)]debiyandraniska@unimed.ac.id, [3)] zulfahmi.indra@unimed.ac.id, [4)] kanasaputras@unimed.ac.id
[5)] adidtya@unimed.ac.id

Submitted :Aug 25, 2025 | **Accepted** : Sep 18, 2025 | **Published** : Oct 2, 2025

Abstract: This study aims to develop an *Android* application capable of securely encrypting and decrypting images using a hybrid cryptographic method. The system combines the ChaCha20 algorithm as symmetric cryptography to encrypt image files, and Elliptic Curve Cryptography (ECC) as asymmetric cryptography to encrypt the ChaCha20 key. The key used is temporary (ephemeral), ensuring that only the intended recipient who possesses the appropriate ECC private key can decrypt the file. The application was developed using the Kotlin programming language in *Android* Studio, with a PHP-based backend and MySQL database. Testing was conducted using the black-box method and involved 15 beta testers to evaluate functionality, security, and usability aspects. The results show that all features of the application run properly, and the encryption and decryption processes can be performed efficiently and securely. Beta testers gave an average rating of 4.6 out of 5 and stated that the application is easy to use and provides sufficient protection for personal data. Therefore, the developed application successfully meets the objectives of the study and offers an alternative solution for securing image file transfers between users via *Android* devices**.**

Keywords: Android, ChaCha20, ECC, Hybrid Cryptography, Image encryption.

INTRODUCTION

The rapid development of information and communication technology, particularly on Android-based devices, has intensified the need for effective data security. Messaging applications and data exchange platforms are increasingly vulnerable to threats such as eavesdropping and unauthorized access, making reliable protection methods essential. Several cases in recent years highlight the urgency of this issue. In 2024, a case in Banyuwangi revealed that a teacher illegally accessed and sold government personnel data, gaining significant financial profit and exposing weaknesses in internal data management (Safitri & Rastika, 2024).

In 2023, nearly 35 million Indonesian passport records were allegedly leaked and found circulating online, raising serious concerns about the protection of citizen information (Putranto Saptohutomo, 2023). Likewise, in 2024, the leakage of voter data from the national election system resulted in sanctions for the electoral commission due to negligence in safeguarding personal information (Nur Alfiyan, 2024). These incidents underscore the pressing need to secure data transmission and storage, especially when handling personal and sensitive information.

To address such vulnerabilities, cryptography has become a fundamental tool in ensuring confidentiality, integrity, and authenticity of data. Symmetric cryptographic algorithms are valued for their efficiency in processing large volumes of data, while asymmetric algorithms provide secure key distribution and authentication mechanisms. However, each method carries its own limitations: symmetric methods rely heavily on secure key sharing, whereas asymmetric methods often demand higher computational resources.

Hybrid cryptography emerges as a promising solution by combining the strengths of both approaches. The ChaCha20 stream cipher offers lightweight and high-speed encryption, making it suitable for handling multimedia data such as images (Putu et al., 2025). On the other hand, Elliptic Curve Cryptography (ECC) provides robust security with relatively smaller key sizes, enabling secure key management without imposing excessive

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

computational overhead (Perdana et al., 2022). When integrated, these algorithms can deliver both efficiency and security, addressing the shortcomings of standalone methods.

The implementation of cryptography in android applications has also been carried out in various studies Setyawan (Bagoes Setyawan et al., 2024) designed a blueprint image encryption and decryption application using the android-based RSA algorithm to improve data security at PT Patco Elektronik Teknologi. According to Wahyudi, (2022), Android applications are effective in accommodating user needs due to their modular and integrative development capabilities, supported by a broad and diverse Android ecosystem. Similarly, Prayitno et al. (2025) emphasize that Android-based applications can be further optimized in specific scenarios such as user data security, including message and file encryption, making them a suitable platform for developing mobile-based security systems.

The objective of this research is to design and implement an Android-based application for image encryption and decryption using a ChaCha20–ECC hybrid model. By applying this combination, the system is expected to offer fast encryption performance while ensuring strong protection of cryptographic keys. This approach is not only intended to safeguard sensitive images against unauthorized access but also to demonstrate a practical cryptographic model suited for the mobile environment, where both security and efficiency are critical. This work hopefully can contribute to the growing need for secure mobile applications by proposing an encryption solution tailored to the challenges of modern cybersecurity. It aims to enhance data confidentiality, reduce risks of leakage during transmission or storage, and strengthen public trust in mobile-based information systems.

LITERATURE REVIEW

Research in cryptography for mobile applications has evolved toward balancing efficiency and security. Traditional symmetric algorithms, such as AES and RC4, have been widely used due to their speed and effectiveness in securing large volumes of data. Rahman et al., (2024) demonstrated that these algorithms provide an additional security layer for both stored and transmitted information, enhancing resistance against unauthorized access. However, symmetric encryption faces challenges in key distribution, which can compromise security in real-world applications.

Stream ciphers have gained increasing attention for image encryption because of their performance advantages. ChaCha20, as highlighted by Rahim & Ramadhan Nasution (2024), efficiently encrypts bitmap images while maintaining fidelity after decryption. This illustrates the suitability of ChaCha20 for applications requiring fast encryption, such as real-time communication or image sharing on mobile devices. Yet, the reliance on secret key distribution remains a limitation if ChaCha20 is deployed in isolation.

To address such limitations, asymmetric algorithms like Elliptic Curve Cryptography (ECC) have been widely examined. ECC provides robust security with smaller key sizes compared to RSA, making it computationally attractive for resource-constrained devices such as Android smartphones (Rilo Taqwa & Haryo Sulaksono, 2020). Its main strength lies in secure key exchange and digital signatures, which enhance trustworthiness in data transmission. However, ECC alone is less efficient for bulk data encryption, leading to performance bottlenecks in image encryption scenarios.

Consequently, several researchers have explored hybrid cryptosystems that combine the strengths of symmetric and asymmetric techniques. Widarma (2023) proposed integrating AES with ECC to leverage both speed and secure key management, while Wahyudi et al. (2024) introduced a layered approach using Vigenere Autokey and Caesar Cipher to increase entropy and enhance resistance against cryptanalysis. Similarly, Teguh et al. (2021) applied Rijndael to Android-based messaging, proving the practicality of applying cryptography within mobile environments. These works indicate a strong research trend toward hybridization as a strategy to overcome the trade-offs of individual algorithms.

Despite these advancements, a critical research gap remains in the integration of ChaCha20 and ECC. While ChaCha20 offers fast and lightweight encryption, its lack of secure key management limits independent use. Conversely, ECC provides highly secure key exchange but lacks efficiency for image-level encryption. No significant studies to date have combined ChaCha20 and ECC in Android-based image encryption, leaving unexplored potential for a hybrid cryptosystem that unites both speed and secure key handling.

Therefore, this study seeks to fill that gap by implementing a ChaCha20–ECC hybrid cryptosystem for image encryption in Android applications. The approach is expected to achieve a balance between computational efficiency and robust security, addressing both the practical constraints of mobile platforms and the increasing demands for safeguarding sensitive image data against cyber threats.

The comparison between the ChaCha20 + ECC algorithm and traditional algorithms such as AES + ECC can be seen in Table 1 below.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 1. Comparasion ChaCha20+ECC and AES+ECC

Aspect	ChaCha20 + ECC	AES + ECC
Type	Stream cipher + ECC	Block cipher + ECC
Speed	Fast in software, ideal for mobile/IoT	Fast with hardware (AES-NI)
Energy Efficiency	Low power on small processors	Low power with AES-NI support
Security	Strong, flexible, no padding needed	Strong, industry standard, needs block modes
Implementation	Simple in software	More complex without acceleration
Advantages	Stable across platforms, safe from cache attacks	Widely supported, NIST standard
Disadvantages	Limited adoption, not NIST-standardized	Slower in software-only setups

METHOD

Research Stage

The research stages follow a System Development Life Cycle (SDLC) approach that is customized to build android applications from scratch. Software Development Life Cycle refers to the steps in which system analysts and programmers work while developing information systems (Desyani et al., 2022). This approach is suitable because the developed application, which is a security system based on ChaCha20 and Elliptic Curve Cryptography (ECC) algorithms, is designed completely from the zero stage without any previous development base. The SDLC stages include planning, needs analysis, design, implementation, testing, and maintenance, ensuring the application meets the data security standards required by users. Figure 1 below illustrate how SDLC works

Fig. 1 System Development Life Cycle (SDLC)

a. Planning

In the planning phase, this research began by identifying the primary objective: to design an Android application for securing photos using ChaCha20 and ECC hybrid encryption. The ChaCha20 algorithm was selected for the data encryption process due to its speed and efficiency, while ECC was implemented to maintain the confidentiality of the symmetric key. Through this combination, the application is expected to effectively protect user data and ensure the security of the keys used.

The requirements analysis covered technical, functional, and user experience aspects. On the technical side, the project required cryptography libraries that support the implementation of both ChaCha20 and ECC on the Android platform. Functionally, the application must allow users to upload photos, perform encryption, and subsequently decrypt the protected data. Feedback from potential users was also considered to ensure the application is user-friendly and meets security expectations. Key features include photo encryption and decryption, key encryption with ECC, and notifications for decryption failures or invalid files.

Furthermore, this stage also accounted for necessary resources, scheduling, and potential risks. Development required specific hardware and software, such as Android Studio, alongside a budget allocation for licenses and testing. The development timeline was structured to include planning, implementation, beta testing, and revision phases. Potential risks—such as feature misalignment with user needs, technical issues with the algorithms, or development delays—were anticipated and managed through routine reviews, the use of trusted cryptographic libraries, extensive testing, and updates based on user feedback.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

b. Requirement analysis
The functional requirements primarily cover the core encryption and decryption mechanisms. The non-functional requirements focus on application reliability, efficient performance, and a user-friendly interface. Additionally, research was conducted to identify the most suitable technologies for development.
c. System Design
During the design phase, the architecture of the Android application is meticulously planned. This includes the database structure, the encryption and decryption workflows, and the user interface (UI/UX). This stage also involves the careful selection of libraries and APIs to be integrated, as well as the creation of flow diagrams and data models for the application.
d. Implementation
The application was developed using the Kotlin programming language through Android Studio with the support of Java libraries as the core cryptographic algorithm. The user interface (UI) is designed using XML and connected with View Binding to support encryption, decryption, and file management features, while dependencies such as cryptography library and Room database are added to fulfill the functional needs of the application. The ChaCha20 algorithm is used to encrypt image files, while Elliptic Curve Cryptography (ECC) serves to encrypt the ChaCha20 symmetric key so that data security is maintained. The encryption process produces an encrypted output file that is ready to be sent through the application, while decryption is done on the receiving side by utilizing the ECC private key to unlock the symmetric key and ChaCha20 to restore the file to its original form. This mechanism enables the achievement of end-to-end security in data transfer between users.
e. Testing
The application is tested to ensure all features work properly, including testing encryption-decryption functionality, data integrity with ChaCha20 and ECC, interface ease of use, and system performance and security.

f. Deployment
The app is expected to be uploaded to an android app distribution platform such as Google Play Store (if accepted), or can be shared directly with users through other means.
g. Maintenance
bugs or issues reported by users will be fixed, and updates to the app will be made as needed or feature enhancements are required. Maintenance also includes monitoring the performance of the application as well as adjusting to android operating system updates.

Beta Testing

Beta testing is testing done from the user's perspective. This test is carried out on the basis of wanting to know how much the level of user acceptance before the application is actually released. The results of the calculation of the level of user acceptance will later be used as input to make improvements to the application in the future (Menora et al., 2023). Its primary function is twofold: first, as an application form, it helps the development team identify and select suitable testers by collecting information on their devices, technical expertise, and relevant experience; second, as a feedback mechanism, it provides a structured way for testers to report bugs, evaluate usability, and suggest improvements. In this study, the beta tester form was created and distributed using Google Forms to facilitate easy access and efficient data collection. The participants were undergraduate students from Universitas Negeri Medan, who were selected as testers to represent the academic user environment

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

System Design

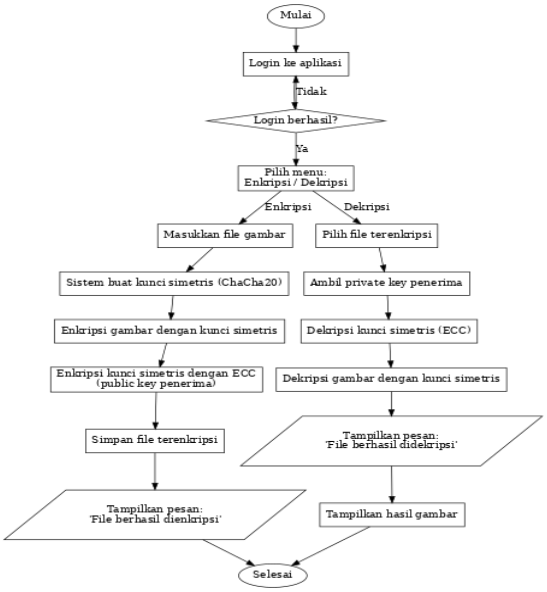

Fig. 2 System Design

The flowchart on figure 2 above illustrates the process of image file encryption and decryption within the application. The process begins when the user logs into the system. If the login is successful, the user can select whether to perform encryption or decryption. For encryption, the user inputs an image file, and the system generates a symmetric key using ChaCha20. The image is then encrypted with this symmetric key, followed by encrypting the symmetric key itself using the recipient's ECC (Elliptic Curve Cryptography) public key. The resulting encrypted file is stored and a confirmation message is displayed.

On the other hand, for decryption, the user selects an encrypted file and provides the recipient's private key. The system first decrypts the symmetric key using ECC, then uses this key to decrypt the image. Finally, the application displays the decrypted image along with a success message. This structured flow ensures that the application securely handles the entire encryption and decryption process, maintaining data confidentiality and integrity.

Application Interface

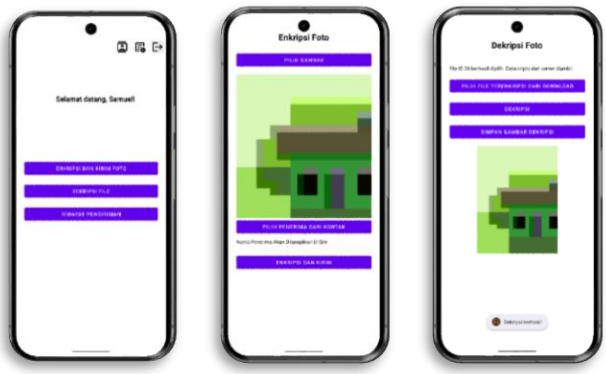

Fig. 3 Application Interface

The System implementation is presented as figure 3 above through the display of the developed mobile application interface. The main menu provides users with several options, including photo encryption, photo decryption, and settings management. The encryption page allows users to select an image and perform the encryption process, which is then visually represented within the application. Similarly, the decryption page

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

enables users to select an encrypted image file and execute the decryption process to restore the original image. These results demonstrate that the system is capable of performing both encryption and decryption functions through an intuitive interface, with images serving as the primary data to validate the functionality of the application.

System Testing

The testing phase was carried out to evaluate the functionality of the developed system. The first scenario tested the encryption process, where users were able to encrypt an image, fill out the submission form, and send the encrypted file. The system successfully sent the encrypted file and redirected the user back to the main menu, indicating that the encryption and submission process worked as expected. The second scenario tested the decryption process by selecting an encrypted file and performing decryption. The system successfully decrypted the file and displayed the original image to the user. These results confirm that both the encryption and decryption features function properly in accordance with the system design. Table 2 below are the system testing for this application development.

Table 2. System Testing

No.	Testing Scenario	Expected Result	Result
1.	encrypt and fill out the submission form and send the data.	successfully send encrypted files, then return to the main menu when successfully sending data	Success
	 		
2.	decrypt and select the encrypted file to be decrypted	decrypt and perform the decryption process then the decrypted image is displayed.	Success
	 		

The testing phase was carried out to evaluate the functionality of the developed system. The first scenario tested the encryption process, where users were able to encrypt an image, fill out the submission form, and send the encrypted file. The system successfully sent the encrypted file and redirected the user back to the main menu, indicating that the encryption and submission process worked as expected. The second scenario tested the decryption process by selecting an encrypted file and performing decryption. The system successfully decrypted

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

the file and displayed the original image to the user. These results confirm that both the encryption and decryption features function properly in accordance with the system design.

System Evaluation

To evaluate the efficiency of the proposed hybrid cryptosystem, a quantitative performance analysis was conducted by measuring the encryption time, decryption time, memory usage, and CPU usage for images of different sizes. The tests were carried out to determine how the system scales with increasing data size and to assess its suitability for resource-constrained environments such as mobile devices. The measurement process was repeated 10 times, and the average values were taken. The results of the performance evaluation are summarized in Table 3.

Image Size (KB)	Encryption Time (ms)	Decryption Time (ms)	Memory Usage (MB)
100 KB	136.92	80.85 ms	41.95 MB
500 KB	367.99	114.27	187.92
1 MB	829.83	209.30	250.64

Beta Testing

The beta testing questionnaire was designed to evaluate the functionality, usability, and security of the application from the user's perspective. In this system beta testing consists of eight questions that cover key aspects of system performance, including whether the application can run smoothly without issues, the clarity and ease of its user interface, and the attractiveness of its design in relation to its intended purpose. Additionally, the questionnaire assesses the clarity of instructions provided during use, the effectiveness of the data encryption and decryption processes, as well as the ability of the system to verify file integrity and securely store encrypted files on the user's device. By addressing these areas, the questionnaire serves as an essential tool to gather user feedback, ensuring that the application is both functional and user-friendly before its official release. This Table 4 below are the questionnaire that used for evaluation

Table 4. Beta Testing Questionnaire

No	Question
1	Can the application run smoothly without issues?
2	Is the application's user interface easy to understand?
3	Is the application design attractive and aligned with its purpose?
4	Does the application provide clear instructions when used?
5	Does the data encryption work properly?
6	Are there any issues when verifying the integrity of the encrypted file?
7	Can the encrypted file be stored securely on your device?
8	Does the data decryption work properly?

The results from that questionnaire are shown in figure 4. the bar chart below, which illustrates the percentage of positive responses for each question. As seen in the chart, several aspects of the application received very high ratings, with Questions 1, 5, 7, and 8 achieving a perfect score of 100%, indicating that the application runs smoothly, encryption and decryption processes work properly, and encrypted files can be securely stored. Questions 2 and 4 also scored highly at 87.5%, suggesting that most users found the interface easy to understand and the instructions clear. However, Question 3, regarding the attractiveness of the application's design, received a score of 50%, and Question 6, related to verifying the integrity of encrypted files, had a rating at 30% for "yes" which mean most people doesn't have an issue when verifying the integrity of encrypted file. These findings suggest that while the core functionality of the application is strong and reliable, improvements are needed in the design aspects and in strengthening the file integrity verification process.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15274
e-ISSN : 2541-2019
p-ISSN : 2541-044X

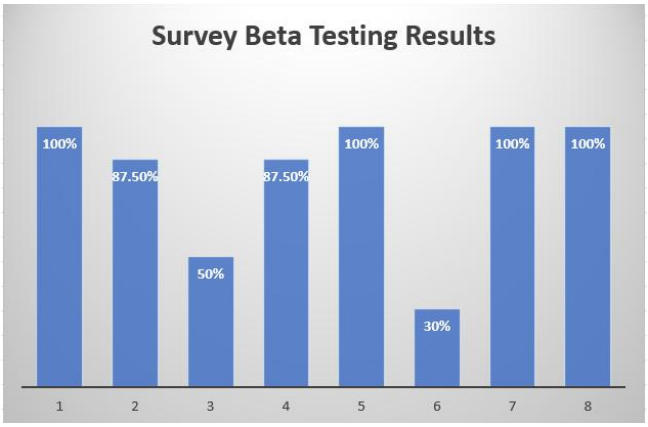

Fig. 4 Beta Testing Results

User Ratings

The results of the user ratings are shown in the figure 5 below, where respondents were asked to provide a star rating for the application. The application achieved an average rating of 4.63 out of 5, indicating a generally positive reception from users. As shown in the bar chart, the majority of respondents, 10 users (62.5%), gave the application the highest rating of 5 stars, while 6 users (37.5%) provided 4 stars. Notably, no respondents gave ratings of 3 stars or below, which reflects a strong level of user satisfaction overall. These results suggest that the application is well-received, with users finding it functional and reliable, though there may still be minor areas for improvement to reach a perfect satisfaction score.

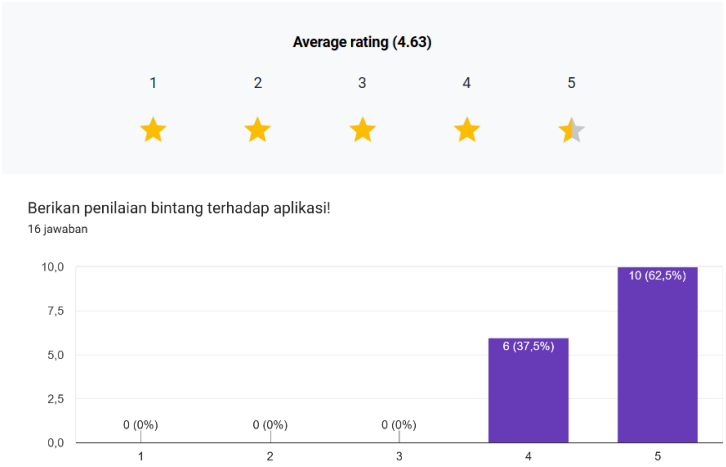

Fig. 5 User Ratings

DISCUSSIONS

The research successfully implemented a hybrid cryptographic system using ChaCha20 for symmetric encryption and Elliptic Curve Cryptography (ECC) for key encryption, addressing the need for secure image transmission on Android devices. The results of the black-box testing confirmed that all core functionalities, including user registration, login, encryption, file transmission, and decryption, performed as expected. This demonstrates that the system is reliable and functional under normal operating conditions. By comparing the measured outcomes with the modeled expectations during the design phase, it is evident that the hybrid approach effectively balances speed, security, and usability. The integration of ChaCha20 ensures fast encryption suitable for mobile devices, while ECC provides secure key exchange without excessive computational overhead.

A significant finding from this research is the successful use of ephemeral public keys during encryption, which adds an extra layer of security by ensuring that each encryption session is unique. This mitigates the risk of replay attacks and enhances the overall robustness of the system. Additionally, the feedback from beta-testers highlights the importance of balancing security with usability. Testers reported smooth performance, clear instructions, and a perception of adequate data protection, indicating that the system meets user expectations for both functionality

*name of corresponding author

and trustworthiness. These findings align with the research goal of creating a user-friendly application that also adheres to modern cryptographic standards.

When compared to traditional systems that rely solely on symmetric or asymmetric encryption, the hybrid approach in this research offers clear advantages. ChaCha20's lightweight nature makes it ideal for mobile environments, while ECC's efficiency addresses the limitations of RSA or other asymmetric algorithms. This combination not only solves the specific engineering problem of securing image data transmission but also sets a foundation for future applications requiring similar security mechanisms. Additionally, while no critical bugs or technical issues were reported during beta-testing, further stress testing and independent security audits are recommended to validate the system's resilience against real-world attacks. Addressing these limitations will prepare the application for wider distribution through platforms like the Google Play Store.

CONCLUSION

The results from testing and evaluation indicate that the Android application implementing a hybrid cryptographic system based on ChaCha20 and Elliptic Curve Cryptography (ECC) has been successfully developed, enabling secure and efficient encryption and decryption of image files. The ChaCha20 algorithm demonstrated efficiency and lightweight performance in handling the encryption process, while ECC provided security for the symmetric key, ensuring that only recipients with the appropriate private key could decrypt and access the data. Black-box testing confirmed that all application features functioned as intended, and user evaluation involving 15 beta testers produced positive results, with an average rating of 4.6 out of 5. The majority of participants agreed that the application is user-friendly and capable of safeguarding personal data effectively.

REFERENCES

Bagoes Setyawan, A., Khaerudin, M., & Setiawati, S. (2024). Perancangan Aplikasi Enkripsi Dan Dekripsi Gambar Cetak Biru Pada PT. Patco Elektronik Teknologi Menggunakan Algoritma RSA Berbasis Android. *NUANSA INFORMATIKA*, *18*(2), 19–25.

Desyani, T., Saifudin, A., Altamerano, H. R., Gunantio, S., Al-Rasyid, S., & Saputra, Y. E. (2022). Pengembangan Sistem Pemantauan Cryptocurrency Berbasis Web Menggunakan Metode Waterfall. *Jurnal Informatika Universitas Pamulang*, *7*(1), 195–200. https://doi.org/10.32493/informatika.v7i1.19724

Menora, T., Primasari, C. H., Wibisono, Y. P., Sidhi, T. A. P., Setyohadi, D. B., & Cininta, M. (2023). Implementasi Pengujian Alpha dan Beta Testing pada Aplikasi Gamelan Virtual Reality. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, *3*(1), 48–60.

Nur Alfiyan, M. (2024). *Kebocoran Data Pemilih 2024 Terkena Sanksi DKPP, KPU: Kita Terima*. Https://Www.Ftnews.Co.Id/Public/Kebocoran-Data-Pemilih-2024-Terkena-Sanksi-Dkpp-Kpu-Kita-Terima .

Perdana, D., Purwiko, P., Dewanta, F., & Afianti, F. (2022). Analisa Penggunaan Elliptic Curve Cryptography pada Sistem Autentikasi pada Internet of Things. *JURNAL MULTINETICS*, *8*(1), 42–49.

Putranto Saptohutomo, A. (2023, July 6). *Data 34 Juta Paspor WNI Bocor Dianggap Kesalahan Konyol*. Https://Nasional.Kompas.Com/Read/2023/07/06/18401011/Data-34-Juta-Paspor-Wni-Bocor-Dianggap-Kesalahan-Konyol?Page=all.

Putu, I., Brama, A., Negara, P. C., Naufal, M., Abror, N., & Tarigan, N. R. (2025). STUDI LITERATUR MENGENAI KINERJA BLOWFISH, AES, CHACHA20, DAN GCM DALAM SISTEM KEAMANAN DATA. *Jurnal Matematika Dan Ilmu Pengetahuan Alam*, *6*. https://doi.org/10.3483/trigonometri.v1i1.800

Rahim, F., & Ramadhan Nasution, Y. (2024). Implementasi Algoritma ChaCha20 Pada Pengamanan File Citra Bitmap. *JURNAL FASILKOM*, *14*(3), 615–624.

Rahman, R., Surya Pratama, N., Sains, J., Balaikota No, J., Harapan, B., Bacukiki Bar, K., Parepare, K., & Selatan, S. (2024). Peningkatan Keamanan Data dengan Kriptografi Modern pada Sistem Operasi. *Jurnal Sistem Informasi Dan Ilmu Komputer*, *4*. https://doi.org/10.59581/jusiik-widyakarya.v2i2.3995

Rilo Taqwa, A., & Haryo Sulaksono, D. (2020). IMPLEMENTASI KRIPTOGRAFI DENGAN METODE ELLIPTIC CURVE CRYPTOGRAPHY (ECC) UNTUK APLIKASI CHATTING BERBASIS ANDROID Article History ABSTRAK. In *Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika (KERNEL)* (Vol. 1, Issue 1).

Safitri, K., & Rastika, I. (2024, September 24). *Oknum Guru Honorer yang Jual Data BKN Raup Keuntungan Rp 121 Juta*. Https://Nasional.Kompas.Com/Read/2024/09/24/16093191/a-Guru-Honorer-Yang-Jual-Data-Bkn-Raup-Keuntungan-Rp-121-Juta.

Teguh, A., Alhamdi, F., & Siahaan, R. F. (2021). Penerapan Kriptografi Dalam Pengamanan Pesan Text Berbasis Android Dengan Menggunakan Metode Rijndael. *Jurnal Mahajana Informasi*, *6*(2).

Wahyudi, E. N., Ardhianto, E., Handoko, W. T., Murti, H., Supriyanto, E., Lestariningsih, E., & Redjeki, R. S. (2024). JIP (Jurnal Informatika Polinema) PENINGKATAN KEAMANAN DATA MELALUI TEKNIK

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

SUPER ENKRIPSI MENGGUNAKAN ALGORITMA VIGENERE DAN CAESAR. *JIP (Jurnal Informatika Polinema)*, *10*(5), 315–321.

Wahyudi, T. (2022). Pengembangan Aplikasi Berbasis Web dan Android Sebagai Penunjang Kerja di Indonesia: Systematic Literature Review. *Journal Computer Science*, *1*(2). https://scholar.google.co.id/.

Widarma, A. (2023). InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan Attribution-NonCommercial 4.0 International. Some rights reserved. *InfoTekJar : Jurnal Nasional Informatika Dan Teknologi Jaringan*, *8*(1), 1–5. http://bit.ly/InfoTekJar

*name of corresponding author