Android-based Patient Medical Record Data Security Application using AES and RSA Method Cryptography

Rudhi Wahyudi
University of Technology Yogyakarta
Yogyakarta, Indonesia

Moh. Ali Romli
University of Technology Yogyakarta
Yogyakarta, Indonesia

ABSTRACT
Electronic Medical Records (EMR) have become an important part of modern health systems, as they store information such as demographics, medical history, and laboratory test results. The confidentiality and security of ERM data is essential to prevent misuse. In the context of data exchange between health services, the development of Android applications based on Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) is a solution to existing problems. This research focuses on the implementation of AES and RSA encryption to maintain the confidentiality and integrity of ERM data. The app uses AES to protect patients' medical record data on Android devices, with encryption keys secured using the RSA algorithm. This approach provides additional security by protecting encryption keys from unauthorized access. The use of this application effectively maintains the confidentiality of ERM data, reduces the risk of data misuse, and provides a sense of security to patients. Strong safety encourages wider use of EMR and improves overall healthcare quality.

General Terms
Information technology, healthcare system, cryptography

Keywords
Andorid, RME, AES, RSA, security.

1. INTRODUCTION
Electronic Medical Record Data (ERM) of patients is a digital system that is used as an electronic information storage containing health certificates and health services obtained by patients [1]. Keeping patient data secure and confidential helps foster trust between patients and healthcare as a whole. Electronic Medical Record (ERM) is regulated in PERM No. 269 of 2008 Articles 10 and 12 [2], and ITE Law No. 11 of 2008 Articles 30 and 32 [3]. ERM written in PERMENKES It is still limited to legal aspects and has not explicitly regulated the data privacy of ERM itself. Patient medical records have switched to electronic based on the regulation of the Minister of Health of FMD number 24 of 2022 concerning medical records. Through this policy, health care facilities are required to carry out an electronic FMD patient medical history recording system in Makasud to become a regulatory framework supporting the implementation of health technology transformation which is part of the 6th pillar of health transformation [4]. Hospitals and clinics have used medical information management systems, to ensure the security of patient medical record data. Namun, penggunaan sistem ini It is still not possible to guarantee the full security of patient medical record data because often the information can be accessed from outside the network or through unprotected mobile devices [5]. Development of Android-based patient medical record data security applications by applying hybrid cryptography where AES (Advanced Encryption Standard) as symmetric cryptography and RSA (Rivest-Shamir-Adleman) as asymmetric cryptography. AES is used to protect the patient's electronic medical record data on the android device, while RSA is used to secure the "session key" (AES encryption), and then the session key is encrypted using the user's public key [6]. With this app, the patient's electronic medical record data is encrypted using AES and RSA so that it can only be accessed by authorized parties and has the corresponding encryption key. In addition, the use of android applications also allows medical personnel to access medical record data safely and efficiently from electronic devices. The application developed is expected to provide andorid-based electronic medical record data security using AES-256 and RSA can provide better patient data protection.

2. LITERATUR REVIEW
The first step in the development of electronic medical record data security applications is to understand the underlying architecture of the application, especially in the context of encryption and data security [7]. Existing systems should be evaluated to understand key concepts in electronic medical data protection, including encryption and access control. Standard measures should be identified to initiate the development of effective electronic medical record data security applications.

One of the most crucial aspects of electronic medical record data security is ensuring that any data collected must undergo a pre-processing process to eliminate sensitive information and minimize the risk of unauthorized access [8]. Once the pre-processing process is complete, the data can be transferred to the appropriate security algorithm, where the data can be securely encrypted and appropriate access controls can be implemented to protect the confidentiality and integrity of the data. In order to find the best methods and tactics for the building of this information system, literature studies are undertaken to investigate methods and strategies that have been used in the past. In this study, a variety of subheadings and keywords were used in the literature review. To identify each of them, see the list below.

2.1 Android Apps
An application is software that operates independently of the technical capabilities of the operating system [9]. Basically, applications can be run on various types of operating systems without having to adjust code or special configurations. This allows the application to be more flexible and can be used on a variety of devices and platforms. Android is a mobile operating system developed by Google based on modified versions of the Linux kernel and designed specifically for mobile devices such as smartphones, tablets, and smartwatches [10]. Android provides an open platform for software developers to create diverse and innovative applications, Android applications can be used for various purposes, such as communication, entertainment, education, health, and so on. In Android application development, there are various technologies and programming languages that can be used, such as Java, Kotlin, and C ++.

2.2 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is an encryption algorithm that can be used to protect data. Information can be encrypted and decrypted using the symmetric ciphertext block of the AES method. When data is encrypted, it becomes unreadable and is changed to ciphertext; when data is decrypted, it returns to its plaintext state. [11]. The Advanced Encryption Standard (AES) algorithm is a symmetric block cipher algorithm that takes plain text in 128-bit blocks and converts it into ciphertext using 128, 192, and 256-bit keys [12]. The AES algorithm uses a substitution-permutation network, with multiple spins to generate ciphertext. The number of revolutions depends on the key used. A 128-bit key requires ten turns, a 192-bit key requires 12 turns, and a 256-bit key requires 14 turns. The AES structure scheme is shown below.

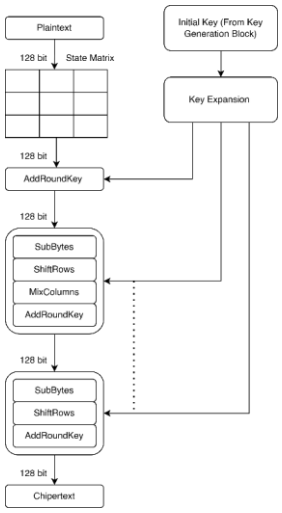

Fig 1: AES Structure Scheme

The encryption process on the AES algorithm is carried out through 4 stages that are carried out repeatedly. These stages are:

1. Byte turnover, the block text bytes are replaced based on the rules defined by the predefined S-box.
2. Panning rows, all rows except the first one are shifted by one block.
3. Mixing columns, randomize more messages by mixing column blocks.
4. Adding roundd keys, message in XOR with each roundd key.

When perfoERMd repeatedly (according to the key used), these steps ensure that the final ciphertext is secure. The AES ciphertext decryption process is similar to The encryption process is in reverse order. It can be concluded that Advanced Encryption Standard (AES) is a symmetric chipertext block that can encrypt (encipher) and decrypt (decipher) information [13].

2.3 Rivest Shamir Adleman (RSA)

The RSA algorithm is one of the most popular public key algorithms and is still used today. The advantage of this algorithm lies in its exponential process, which is to decompose a number into 2 prime numbers, and these two prime numbers take a long time to factor [14]. The RSA scheme itself uses a block cipher scheme. Where before encryption, the existing plaintext was divided into blocks of equal length, where the plaintext and ciphertext were integers between 1 and n, where n was usually 1024 bits, and the length of the block itself was less than or equal to log(n)+1 with base 2 [15]. The functions of encryption and decryption are described in the following functions.

C = Me mod
M = Cd mod n (function for decryption)
C = Ciphertext
M = Message / Plaintext
e = public key
d = private key
n = splitter modulus.

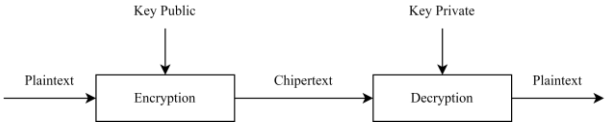

Fig 2: RSA Structure Scheme

One cryptographic method, RSA, uses a public key for encryption and a private key for decryption. The public key for encryption is known as the "public key," and the private key for decryption is known as the "private key" [16]. Anyone with access to the public key can encrypt data, but only the owner of the private key is able to decrypt it. For the encryption algorithm, Dekripi uses RSA with the formula, where the secret key is possessed by only a select few persons while the public key can be owned by anybody.

Encryption : $Ee\ (m) = c = m\ e\ mod\ n$

Decryption : $Dd(c) = m = c\ d\ mod\ n$

For the generation of RSA key pairs, an algorithm is used.

1. The huge prime numbers p and q were randomly selected. The values of p and q must remain a secret.
2. determined that $n\ =\ p\ x\ q$. It is not necessary to conceal the size of n.
3. $m\ =\ (p\ -\ 1)(q\ -\ 1)$ was calculated.
4. The public key, designated by the number e, is primed with respect to m. Due to the fact that e is prime with respect to m, the second-largest divisor is 1, $\gcd(e, m) = 1$. You can use Euclidean algorithms to discover it.
5. The private key, designated d, is calculated so that $(d\ \times\ e)\ mod\ m\ =\ 1$. The Extended Euclid algorithm can also be used to determine the equivalent d value.

The security of the RSA algorithm lies in the level of difficulty in factoring non-prime factors into its prime factor, which in this case $n\ =\ p\ \times\ q$. If n is successfully factored into p and q, then $m\ =\ (p-1)(q\ -\ 1)$ can be calculated. And since the encryption key e has been announced (not kept secret- will), then the decryption key d can be calculated through the equation $(d\ \times\ e)\ mod\ n\ =\ 1$ [17]. As long as there is no way to factor large numbers into prime factors, then as long as the security of the RSA algorithm is guaranteed.

3. RESEARCH METHOD

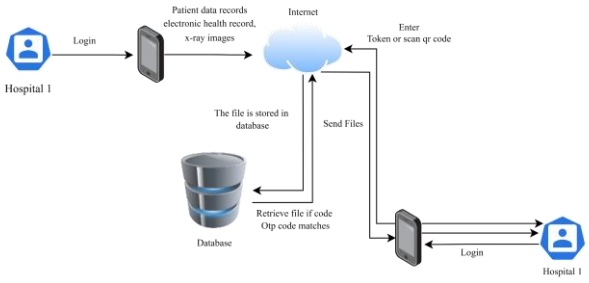

Fig 3: System Architecture

Fig 3, is the proposed architectural design of the system. This system involves health care workers as the main users. The primary task of healthcare workers is to manage healthcare data, including patient referral data, electronic health records, and transcription of health reports. Healthcare workers can create Electronic Medical Record (ERM) data and encrypt patient ERM data before exchanging data between health services. The destination hospital receives the patient's encrypted ERM data and decrypted using a random string token or QR code from the referral healthcare service. By using this approach, the patient's ERM data is kept secure during data exchange.

3.1 System Design Logic

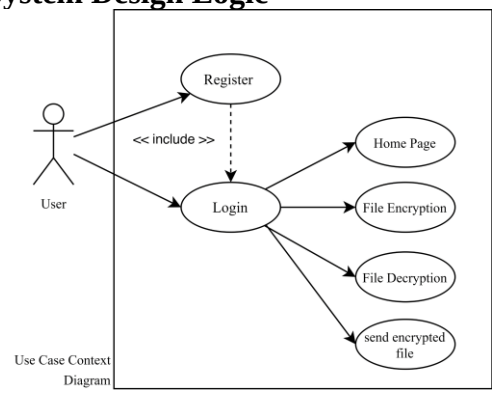

Fig 4: Use Case Diagram

A use case diagram is a modeling used to identify the functional needs of a system [18]. Use case diagrams are the first step in modeling a system and are used as tools in the system analysis and design process to identify and gather the functional requirements of the system to be developed. Fig 4 shows the use case of Android-based medical record data security application using AES-256 and RSA.

3.2 Analysis of information technology systems and requirements

Technology architecture that adheres to the concepts of service orientation is known as service oriented architecture (SOA) . This idea of "service orientation" adopts the strategy of breaking down a huge problem into a number of smaller services, each of which aims to address a particular issue. Once all issues can be divided into separate services, the issue must be resolved by enabling participation in the orchestration from all services. Technology architecture that adheres to the concepts of service orientation is known as service oriented architecture (SOA). This idea of "service orientation" adopts the strategy of breaking down a huge problem into a number of smaller services, each of which aims to address a particular issue. Once all issues can be divided into separate services, the issue must be resolved by enabling participation in the orchestration from all services.

4. RESULT AND DISCUSSION

4.1 Implementations AES

AES implementations in ERM data security applications encrypt and decrypt files before they are exchanged between health services. For applications that require secure file transfers, such as web servers and Android-based applications, AES was chosen because it can encrypt data with various key lengths, including 128-bit, 192-bit, and 256-bit. The following is the implementation of AES in an ERM data security application. Before encrypting and decrypting, key expansion is carried out to obtain the round key. With the implementation of AES 128 bits, the maximum key length is 16 bits and 10 key rounddes are required from the expansion process. If the key used is “REKAMMEDISPASIEN”, key converted into DataReader form into 4x4 state then perfoERMd RotWord function, converting every bit in column 4 up 1 time with 0th RounddKey, which results in 1st RounddKey. Then replace the RotWord result with the existing value in Rijndael's S-Box table (SubBytes). To get the first column of the 1st RounddKey, it is necessary to XOR between the first column of the 0th RounddKey and the result of the SubBytes being XOR -kan with Rcon (Roundd Constanta). Then, XOR between the first column (Wi) and the second column of the 0th RounddKey. The expansion results for encryption and decryption are used to generate the 1st RounddKey of the entire process. This process is repeated 10 times.

In the AES algorithm, the encryption process consists of four types of byte transformations: SubBytes, ShiftRows, MixColumns, and Add RounddKey. This process will be perfoERMd on the file before it is stored in the ERP data security application database. "DATAMEDISPATIENTA" is plain text that will be encrypted. SubBytes, Shiftrows, MixColumns, and AddRounddeKey are the four transformations that will be used to insert input on the 1st roundde. This process still functions as a pre-roundde. The result of SubBytes substitution is done by shifting cyclically in rotation on the last three lines of state (first line r = 0, not shifted). Line two is shifted once to the left; row three is shifted twice; and row four is shifted three times. MixColumns changes the state of each column by using polynomials for each column. To get the first roundd of encryption, the final step is to XOR between the result of the MixColumns and the 1st RounddKey, this process is called AddRounddKey. Do this up to 10 times. The resulting ciphertext is the numbers 45, 71, 93, 45, 91, 141, 31, 15, 34, 113, 96, 247, 208, 158, 37, 19.

The decryption process on AES uses the same key for the encryption process. After that, enter the first 16 bytes of the converted ciphertext into hexadecimal format into a 4x4 state. This process is called AddInvRounddKey if there is an XOR between Ciphertext and RounddKey 10. The AddInvRounddkey process still works as an Initial-roundde and will be the input for the first roundde to be processed with four transformations: InvShiftrows, InvSubBytes, AddInvRounddKey, and InvMixColumns. InvShiftrows on the result of Initial-Roundde, which is executed by rotating a cyclic. Line two is shifted three times to the left, row three twice, and row four once. The values in the S-Box-1 table (InvSubBytes) are used to substitute the results of InvShiftrows. This process is called AddInvRounddKey, which is the XOR of InvSubBytes with the 9th RounddKey. InvMixColumns changes the result of AddInvRounddKey by operating the state of each column. Converting each column to a polynomial is a way of doing this operation. InvMixColumns transformation is perfoermd until the 10th roundd; InvShiftrows, InvSubBytes, and AddInvRounddKey transformations are not perfoermd on the 10th roundd.

4.2 Implementations RSA

RSA methods in ERM data security applications include the management of private and public key pairs used in RSA-OAEP encryption schemes. RSA public keys are used for AES symmetric key encryption, RSA private keys are used to access data that has been encrypted. The key generation process of the RSA algorithm determines two prime numbers ranging from 1 to 1000, which are then randomized and start with p and q. Both values must be kept using prime values $p = 23$ and $q = 9$, and multiplication must be done to give n values of both primes.

$$n = p \times q$$
$$n = 23 \times 9$$
$$n = 207$$

hitung nilai totien (island) dengan cara
$\emptyset\,(n) = (p-1)(q-1)$
$\emptyset\,(n) = (23-1)(9-1)$
$\emptyset\,(n) = (22)(8)\ \emptyset\,(n) = 176$

Euclid's algorithm can be used to systematically find $\gcd(e, \emptyset(n)) = 1$. Using the formula $\gcd(e, \emptyset(n)) = 1\ e\ \emptyset(n) = 1\ mod\ n$ or $e\ \emptyset\,(n)\ mod\ n = 1$, $e = 21$ so $gcd(21{,}176) = 1$. To generate the secret key d, such that $e \times d = 1\ mod\ (\emptyset(n))$ or $d = (1 + k\ \emptyset(n))$. Here, the value of k is the free amount used to produce the value of an interger or integer. If we take the assumption that $k = 17$, then $d = \frac{(1+k\ \emptyset(n))}{e}$, and $d = \frac{(1+17.176)}{21} = 181$, we will get the following key pair.
public $key(e, n) = (21{,}207)$
private $key(d, n) = (181, 207)$
In the encryption process using the public key which is $K = (e, n) = (21{,}207)$ and using the formula $C = pe\ mod\ n$. Before encrypting, all you have to do is change the password to ASCII encoding.

these m values are still located inside the hose $[0.207 - 1]$, $K = (21{,}207)$, $C = pe\ mod\ n$
$C_1 = 04921\ mod\ 207 = 160$
$C_2 = 05721\ mod\ 207 = 132$
$C_3 = 05421\ mod\ 207 = 39$
$C_4 = 04821\ mod\ 207 = 60$
$C_5 = 04921\ mod\ 207 = 160$
$C_6 = 04821\ mod\ 207 = 60$
$C_7 = 04821\ mod\ 207 = 60$
$C_8 = 05121\ mod\ 207 = 153$
$C_9 = 04921\ mod\ 207 = 160$
$C_{10} = 05721\ mod\ 207 = 132$
$C_{11} = 05721\ mod\ 207 = 132$
$C_{12} = 04921\ mod\ 207 = 160$
$C_{13} = 04821\ mod\ 207 = 60$
$C_{14} = 05121\ mod\ 207 = 153$
$C_{15} = 04921\ mod\ 207 = 160$
$C_{16} = 04821\ mod\ 207 = 60$
$C_{17} = 04821\ mod\ 207 = 60$
$C_{18} = 05021\ mod\ 207 = 53$

ciphertext yang dihasilkan.

$$C = \begin{matrix} 160 & 132 & 39 & 60 & 160 & 60 \\ 60 & 153 & 160 & 132 & 132 & 160 \\ 60 & 153 & 160 & 60 & 60 & 53 \end{matrix}$$

The decryption process uses the private key, which is $K = (d, n) = (181{,}207)$, and uses the formula $P = Cd\ mod\ n$. In other words, this process returns the cipher text to the form of plaintext. $K = (181{,}207)\ P = Cd\ mod\ n$.
$m_1 = 160181\ mod\ 207 = 170$
$m_2 = 132181\ mod\ 207 = 115$
$m_3 = 39181\ mod\ 207 = 058$
$m_4 = 60181\ mod\ 207 = 151$
$m_5 = 160181\ mod\ 207 = 170$
$m_6 = 60181\ mod\ 207 = 151$
$m_7 = 60181\ mod\ 207 = 151$
$m_8 = 153181\ mod\ 207 = 001$
$m_9 = 160181\ mod\ 207 = 170$
$m_{10} = 132181\ mod\ 207 = 115$
$m_{11} = 132181\ mod\ 207 = 115$
$m_{12} = 160181\ mod\ 207 = 170$
$m_{13} = 60181\ mod\ 207 = 151$
$m_{14} = 153181\ mod\ 207 = 001$
$m_{15} = 160181\ mod\ 207 = 170$
$m_{16} = 60181\ mod\ 207 = 151$
$m_{17} = 60181\ mod\ 207 = 151$
$m_{18} = 53181\ mod\ 207 = 189$

4.3 Interface Design

By implementing cryptographic hybrid encryption and decryption measures (AES and RSA) and based on established business flows, it produces an andorid-based application to keep ERM data secure. Here's what the application interface looks like after the system development stage.

4.3.1 Login and Register Page View

The login page displays a page to verify the user whether the email and password match or not, if appropriate the system will forward on the home page if not will be asked to enter the appropriate email and password again. The register page works if the user does not have an account yet or wants to create a new account.

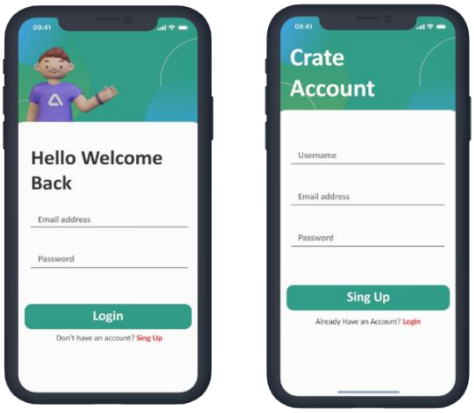

Fig 8: Login and Register Page View

4.3.2 Home Page View

The home page contains the history of encryption files by extension, the history of the latest encryption files, and can decrypt files with the tokens that have been obtained.

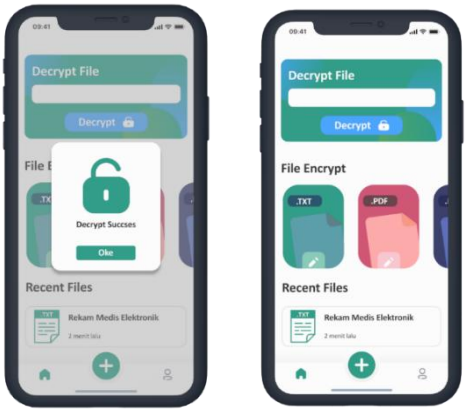

Fig 9: Home Page View

4.3.3 Page View File Encryption

This page serves to encrypt patient medical record data, where users will be asked to enter a file name, file description, and upload the file then the encryption process is carried out.

Fig 10: Enkripisi File Page View

4.3.4 File Details and Encryption History Page

On this page contains details of files that have been encrypted ranging from file names, file descriptions, to file extensions, on the file details page there is a get token button to decrypt files contained on the home page. This page contains a history of files that have been encrypted, where each file contains the file name, extension, and encryption time of the file.

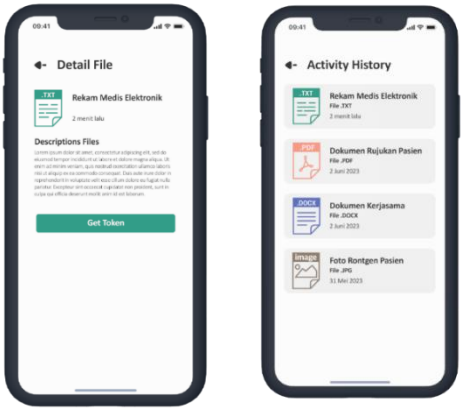

Fig 11: File Details and Encryption History Page

4.3.5 Profile and Setting Pages

This page contains user profiles ranging from profile photos, usernames, emails used, on the profile menu users can change user data, view file encryption history, change passwords, and log out of the account being used.

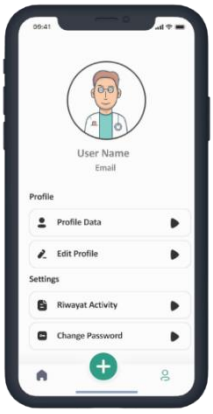

Fig 12: Profile and Settings Page

4.4 Discussion

Testing is an important step in ensuring that the system that has been developed meets user expectations and provides a satisfactory user experience. Questionnaires (UEQs) are conducted using the User Experience Questionnaire (UEQ), to measure various aspects of user experience, such as effectiveness, efficiency, satisfaction, and overall impression. The test was carried out by providing questionnaires to 22 health service staff. The results of the test in Table 1, where the attractiveness with a value of 1,518 shows that the system has a fairly good level of attractiveness, able to lure users to interact with the application or platform. Clarity with a value of 1,553 indicates that the system interface is considered self-explanatory and easy to understand by users. Efficiency with a value of 1,263 shows the potential to increase user efficiency in using the system. Accuracy with a value of 1,211 indicates the level of reliability and accuracy of the system in performing the tasks desired by the user. Stimulation with a value of 1.434 indicates that the system provides a sufficiently good level of stimulation in providing interesting stimuli and experiences to users, and novelty with a value of 0.605 indicates that there is potential to increase the level of novelty and innovation of the system. The results of the calculations are depicted on a scale, Fig 13 with values collected in benchmarks to give conclusions about the quality of the system being evaluated. The results of these tests will help in the improvements and adjustments needed to maximize the quality of the user experience of the system that has been designed.

Table 1 UEQ Results

UEQ Scales (Mean and Variance)		
Attractiviness	1.518	1.37
Clarity	1.553	1.59
Efficiency	1.263	1.95
Accuracy	1.211	1.54
Stimulation	1.434	1.25
Novelty	0.605	0.77

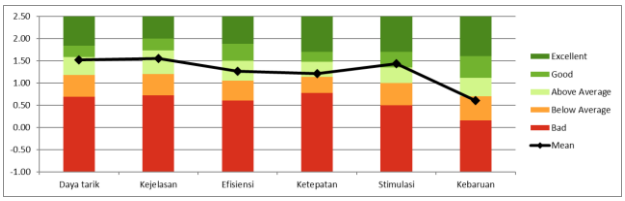

Fig 13: UEQ Benchmark

5. CONCLUSION

This application aims to protect patient medical record data to keep it safe and encrypted when stored and transferred. The AES-256 algorithm is used to encrypt medical record data. AES-256 is one type of modern encryption algorithm and is widely used because the key used has a length of 256 bits. The RSA algorithm is used to secure the AES encryption keys used in the application. RSA is an asymmetric cryptographic algorithm that allows the use of two keys, namely a public key for encryption and a private key for decryption. When a patient's medical record data is stored, the app uses AES-256 to encrypt that data. This prevents unauthorized access to sensitive information. Before the medical record data is transmitted, the AES encryption key will be encrypted using the RSA algorithm using the public key provided by the recipient. This ensures that only recipients who have the

corresponding private key can decrypt the AES encryption key. By combining AES-256 encryption and key security using RSA, this application provides a high level of security to protect patient medical record data in the Android environment. This gives users confidence that their sensitive information is well protected from unauthorized access or misuse.

6. REFERENCES

[1] As'Ari, I., & Purwito, D. 2021, Penerapan Sistem Komunikasi Rujukan PSC 119 – Satria Oleh Fasilitas Pelayanan Kesehatan Di Kabupaten Banyumas. https://doi.org/10.37036/ahnj.v6i2.166

[2] Pemerintah Indonesia. 2008. PeERMnkes No. 269 Tahun 2008 tentang Praktik Kedokieran, pasal 10 dan 12 perlu mengatur kembali penyelenggaraan Rekam Medis dengan Peraturan Menteri Kesehatan. Jakarta.

[3] Pemerintah Indonesia. 2008. Undang-undang (UU) Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik. Jakarta.

[4] Pemerintah Indonesia. 2022. Peraturan Menteri Kesehatan (PMK) nomor 24 tahun 2022 tentang Rekam Medis. Jakarta.

[5] Hartono, Bambang 2013, Sistem Informasi Manajemen Berbasis Komputer, Rineka Cipta : Jakarta.

[6] J. Mirza and M. Sharma, "A Hybrid Cryptographic Technique for Secured Authentication in Cloud Computing", International Journal of Computer Applications, vol. 141, no. 13, pp. 51-56, 2016. Available: 10.5120/ijca2016909797

[7] Sinchana, M. K., and R. M. Savithramma. "Survey on Cloud Computing Security." In Innovations in Computer Science and Engineering, pp. 1-6. Springer, Singapore, 2020.

[8] Anand, N., 2020. Using AES Algorithm Encryption and Decryption of Text File , Image and Audio in Openssl and Time Calculation for Execution. 22(6), pp.39–44.

[9] Randi, A., Lazuardy, K., Chandra, S., & Dharma, A. (2020). Implementasi Algoritma Advanced Encryption Standard pada Aplikasi Chatting berbasis Android. JIKOMSI Jurnal Ilmu Komputer dan Sistem Informasi, 3(1), 2-4.

[10] L. Ma, X. Sun and W. Jin, "Symmetric–asymmetric hybrid encryption and decryption system based on chaotic iris phase mask and computer-generated holography", Optical Engineering, vol. 59, no. 08, 2020. Available: 10.1117/1.oe.59.8.083106.

[11] Sari, E. K., & Febryansah, R., 2022, Analisis Perancangan Aplikasi Rujukan Pasien Pada Rsud Dr. A. Dadi Tjorodipo.

[12] Gandhewar N, Sheikh R. Google Android: An Emerging Software Platform For Mobile devices. International Journal on Computer Science and Engineering. 2010; 1(1): 12-17.

[13] Ilham, L. I., & Widyassari, A. P. (2021). Pengembangan Aplikasi Pesan Instan Terenkripsi Menggunakan Algoritma Kriptografi AES (Advanced Encryption Standard). Jurnal Teknik Elektro Smart, 1(1), 1-4.

[14] Irawan, C., Winarno, A., Studi, P., Informasi, S., Komputer, F.I. and Nuswantoro, U.D., 2020. Kombinasi Algoritma Kriptogarafi Aes Dan Des Untuk Enkripsi. pp.978–979.

[15] Fauzan, D. A., & Fathurrozi, A. (2023). Perancangan Aplikasi Pengamanan Data Menggunakan Algoritma RSA (Rivest Shamir Adleman) dan AES (Advanced Encryption Standard) Berbasis Web. 4(1), 91–104.

[16] Murad, S. H., & Rahouma, K. H. (2021). Implementation and Performance Analysis of Hybrid Cryptographic Schemes applied in Cloud Computing Environment. Procedia Computer Science, 194, 165–172. https://doi.org/10.1016/j.procs.2021.10.070

[17] Putra, T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD.

[18] T. Erl, Service-Oriented Architecture Second Edition, Upper Saddle River, New Jersey: Prentice Hall, 2017. Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.