DOI: 10.22146/jnteti.v13i1.4395

Implementation of QR Code Attendance Security System Using RSA and Hash Algorithms

Arif Indra Irawan[1], Iman Hedi Santoso[1], Istikmal[1], Maya Rahayu[2]

[1] Telecomunication Engineering Study Program, School of Electrical Engineering, Universitas Telkom, Bandung, Indonesia
[2] DIII - Telecommunication Engineering Study Program, Department of Electrical Engineering, Politeknik Negeri Bandung, Bandung, Indonesia

[Received: 3 August 2022, Revised: 5 November 2022, Accepted: 5 December 2023]

Corresponding Author: Arif Indra Irawan (arifirawan@telkomuniversity.ac.id)

ABSTRACT — The quick response (QR) code-based attendance application contributes to reducing paper usage and attendance input errors. However, in its implementation process, the QR-code-based attendance at a Bandung school demonstrates weaknesses. Absent students can fake their attendance for themselves or friends. This type of attack is known as fake QR code generation. This research proposes a security authentication system using the Rivest–Shamir–Adleman (RSA) encryption algorithm and the secure hash algorithm 1 (SHA-1) to secure QR code-based attendance applications from fake QR code generation attacks. The RSA algorithm encrypts QR code data to maintain privacy, while the SHA-1 algorithm ensures data integrity. Based on this method, the mutual authentication process between the QR code data generated by the student and the attendance reading application by the teacher can be established. The results obtained from a series of tests showed that the security system in the student attendance recording application that had been implemented at Madrasah Aliyah (MA) Al-Mukhlishin could detect and prevent fake QR code generation attacks. The test was conducted by changing the impact of the key length on RSA-1024 bits and RSA-2048 bits. The results showed that in RSA-1024 bits, energy consumption of 0.14 J and time of 1.66 s is more efficient than that in RSA-2048 bits, with energy consumption of 0.19 J and time of 2.09 s. Interestingly, if a higher level of security is required, the key length should be increased at the expense of some energy and time efficiency.

KEYWORDS — Authentication System, Student Attendance, QR Code, RSA, Hash.

I. INTRODUCTION

The use of quick response (QR) codes has now been widely used in various applications such as product promotion, electricity payment identity, cell phone credit payment, healthcare [1], and mobile robots along with the development of smartphone technology [2]. Furthermore, the usage of QR codes can also complement other monitoring and automation applications such as smart systems [3] and Bluetooth [4]. It happens because the process of decoding all versions of QR codes that are decoded using the human eye will be very difficult, owing to the encryption of the message into bits that subsequently constitute a square array [5].

Today, QR codes can be used to complete student attendance data at school. Various ways can be done to record student attendance data, namely manually using paper, using radio frequency identification (RFID) technology [6], or using QR codes that utilize smartphones. However, among the ways of recording attendance mentioned above, only QR codes currently have the potential to be applied in student attendance systems along with the development of smartphones since smartphones can be owned by anyone and are easy to use. In addition, it can reduce the use of paper as a medium for recording student attendance which is considered inefficient and expensive [7]. This paperless student attendance data collection method will have a positive impact on environmental issues and improve the quality of education supported by the effectiveness of the administration system.

However, QR code-based student attendance systems have vulnerabilities such as QR code phishing [8], fake QR code generation, malicious QR codes, and data hacking [9]. These types of exploits allow students to cheat in the attendance data collection system. QR code is an object or product recognition code first developed by a Japanese company, Wave Denso Company [10]. It has the form of a matrix code or two-dimensional bar code [11]. QR code technology has been widely used to identify and recognize various products, including online and digital payment systems [10], [11].

One of identified QR code exploitations involves students copying the message format in a QR code using a scanner tool and then recreate the QR code message, or commonly called fake QR code generation, to deceive the student-based attendance data collection system. Students who do not actually attend classes can check their attendance by passing the QR code data to their friends who do. Therefore, a robust security system is required to strengthen the security of the QR code-based attendance system. As a result, this research proposes a QR code-based authentication system based on Rivest–Shamir–Adleman (RSA) and hashing algorithms. This authentication system was developed and tested at a school in Bandung. The test results of this system demonstrate its ability to prevent spoofing and fake QR code generation. The test results of this system will show its ability to prevent spoofing and fake QR code generation.

Numerous studies on QR Codes have been previously conducted, including research to improve the visual performance of QR by using the halftone method and the Berlekamp Reed-Solomon systematic error correction algorithm [12]. Many studies have attempted to increase the security of this QR code system by combining various techniques, such as adding a QR code layer [13], using proxy re-encryption [14], cryptography [15], [16], [17], watermarking [18], blockchain [19], and steganography [20].

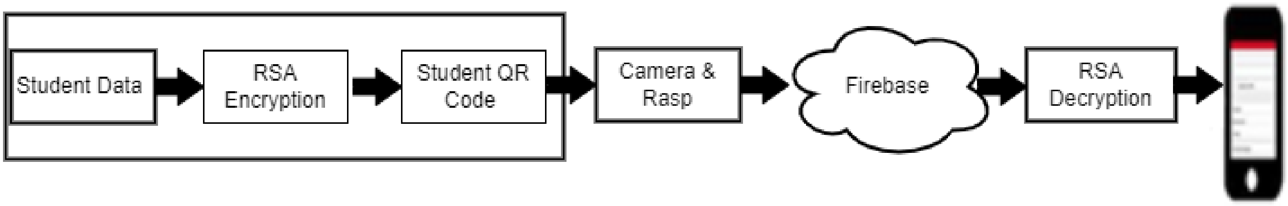

Figure 1. Student attendance system model.

A new type of QR code called two-level QR code (2LQR) has been introduced [21]. It consists of a public level and a private level. The public level functions similar to the storage level of a standard QR code and can be read by traditional QR code applications. On the other hand, the private level requires specialized applications and special input information. These 2LQR codes can be used for private message sharing and authentication. Moreover, dynamic QR codes for payment systems that support SM2, SM3, and SM4 cryptographic algorithms has been proposed [22]. Comparisons were made between these algorithms and other algorithms such as advanced encryption standard (AES) and RSA, using the randomness of the ciphertext and computation time as parameters to measure security performance.

Meanwhile, in contrast to previous studies, the authentication protocol in this study was implemented using a strong encryption algorithm, and on the server side, the encryption results (cyphertext) were stored using a hash copy of the password. Then, the RSA encryption algorithm used in this research was a public key cryptography system where the encryption process only required one key pair that was used simultaneously [23]. The advantage of this algorithm lies in the exponential process and factorization of numbers into two prime numbers, which until now takes a long time to factorize. The algorithm is named after its inventors, namely Ron Rivest, Adi Shamir, and Leonard Adleman (Rivest-Shamir-Adleman), and was published in 1977 at MIT in response to the challenges posed by the Diffie-Hellman key exchange algorithm. The RSA scheme adopts a block cipher scheme, where before encryption is performed, the plaintext is divided into blocks of equal length, where the plaintext and ciphertext are integers between 1 and n, with n typically being 1024 bits and the block length less than or equal to $\log(n) + 1$ with base 2 [23].

A hashing algorithm is an encryption algorithm to convert text into a series of random characters [24]. The number of characters in the hash result is always the same. Hash is a one-way encryption, so a hashed message cannot be restored to its original text. The secure hash algorithm 1 (SHA-1) is one of many hashing algorithms that are commonly used to ensure data integrity. In this research, the SHA-1 was implemented using the hash library. The SHA-1 hashed message used in this research had a data width of 20 bytes and displayed as a 40-digit hexadecimal number. It is believed that this is the best algorithm to improve the security system.

II. METHODOLOGY

The system design is illustrated through block diagrams and flowcharts that explain how the system can work. The method of this system was made to design a security system following the flowcharts previously made, then the validity of the design was reanalyzed. Ultimately, the system was implemented using the concept that had been made into the cloud server and Android application, and the success of the implemented system was marked without any bugs or errors.

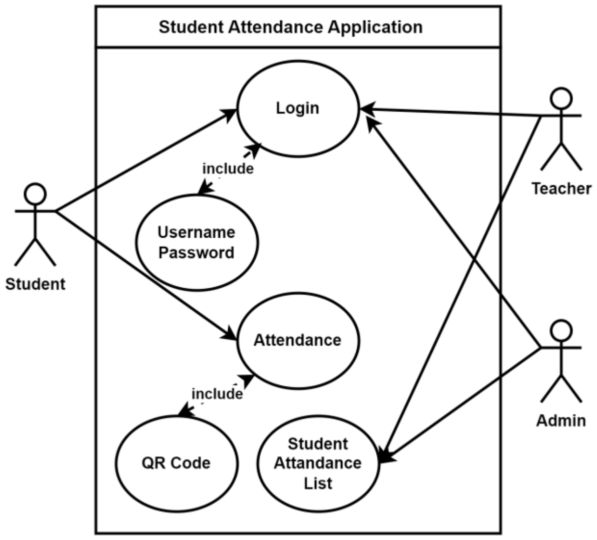

Figure 2. Use case systemin general.

A. DESIGN SYSTEM

The designed system in the process of recording student attendance using a QR Code has several steps that must be completed, as shown in Figure 1. The process of retrieving student attendance data using QR codes started with teachers and students downloading an Android application using an enhanced security system. Users, either teachers or students, must register first if it was their first time using the application. When the class started, students scanned the QR code generated by each student's smartphone using the connected camera module and stored it in the Firebase server database. Student QR codes were generated by encrypting student data using the RSA encryption algorithm. Student QR code reading with the camera module was organized and processed by the Raspberry Pi. In addition, by logging into the teacher application, the homeroom teacher was able to access the list of student attendance on the teacher's phone. All messages had been authenticated from the Raspberry Pi.

Once the initial stages were complete, the next step was to determine how to use the data from the previous steps to generate an electronic log of student attendance records that teachers and administration at the school could access and check. Figure 2 presents the sequence of events when utilizing a QR code for recording attendance presents.

Figure 2 depicts a general system use case diagram on how the students' attendance application must perform. This system was divided into three parts: student, teacher, and administrator. Before students could generate a QR code, they must first log onto their app. The generated QR code would be scanned by the reader. After that it was processed and sent to the database as a student attendance list. Teachers could check the students' status from the student attendance list in the database. Meanwhile, the admin was responsible for managing the database.

B. STUDENT APPLICATION DESIGN

There are several features in the student attendance application.

1) REGISTRATION PAGE

On the registration page, there are several input boxes that must be filled in by Madrasah Aliyah (MA) Al-Mukhlishin Bandung students, such as name, NISN, username, email, and password. After filling in the data, students can click the send button to transfer the data to the database server.

2) STUDENT LOGIN PAGE

The student login page contains two input boxes, namely username and password. When the student clicks the login button, the application will carry out authentication using the data stored in the database during registration.

3) STUDENT PROFILE APPLICATION

The student profile page contains three fields, namely name, NISN, and QR code image, which are then generated dynamically based on the data registered by the student. The QR code image generated on this page determines the student's attendance.

4) ATTENDANCE NOTIFICATION

The notification feature is on the same page as the student profile application. The student profile notification will appear if the student successfully attends.

All features in the student attendance application have a specific process flow, which can technically be described in Figure 3. The students must create an account by entering their full name, NISN, and username. When the student took attendance, the data were added with a timestamp and then encrypted with the RSA technique, resulting in a ciphertext. This ciphertext was decoded into a QR code using the ZXing library, and the result was a bitmap image. This encoded bitmap image recorded student attendance and could be scanned with the Raspberry Pi camera. If the student did not fill in the full name, NISN, and username, the program notified the student to register.

Under this authentication system, the format of the attendance data cannot be read if the student uses a third-party application to read the QR code. If a student attempts to create a fake QR code, they can only read and duplicate the previously encrypted QR code data. Moreover, the timestamp field can identify these actions, so fake QR code generation attacks can be detected and blocked. To break the authentication, the attacker needs to break an RSA of 1024 or 2048 bits, which has yet to be cracked. The largest RSA ever cracked was the 768-bit RSA by Paul Zimmermann *et al*. on 12 December 2009. It took them two years to break this RSA [25].

C. RASPBERRY PI PROGRAM DESIGN

In this research, the Raspberry Pi functioned as a controller to read student attendance data from the QR code generated by the student application. Students must scan the QR code from the application to the web camera on the Raspberry Pi. The Raspberry Pi was equipped with a buzzer to check whether the student's QR code was successfully read. The last component is a push button that was used to reset the Raspberry Pi program during an event of errors or bugs. The two components above were connected to the digital pins on the Raspberry Pi, as shown in Figure 4.

Figure 4 illustrates the flowchart of the Raspberry Pi program. The program initialized the components connected to the Raspberry Pi, such as push buttons and buzzers. In addition, pyzbar was also included in the software to process QR codes. The application built a frame of 400 pixels and used this camera data to determine if there was a QR code. If the pyzbar library detected a QR code, the program would display a box surrounding the QR code and activate the buzzer twice, indicating that the QR code data were read. In the final process, this encrypted data would be transmitted to the Firebase.

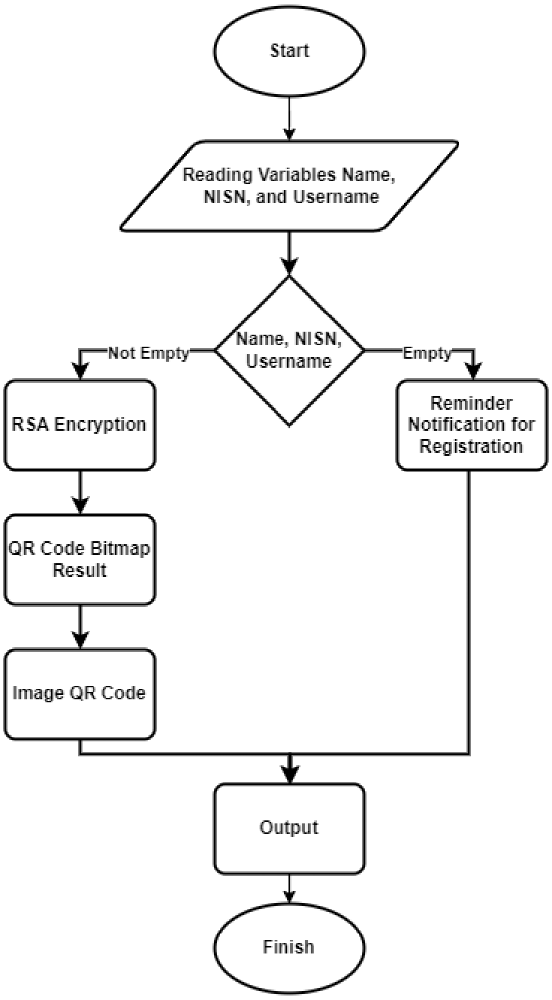

Figure 3. Flowchart of the student attendance application.

D. TEACHER APPLICATION DESIGN

The teacher application displays several features.

1) REGISTRATION

The registration page contains several input boxes that can be filled in by teachers at MA Al-Mukhlishin Bandung, such as username, full name, NIP, email, and password to log in. After the teacher clicks the registration button, the data will be transferred to the database server.

2) TEACHER LOGIN PAGE

The login page contains several input boxes such as username and password. When the teacher clicks the login button, the program will carry out authentication using the information entered during registration.

3) TEACHER PROFILE APPLICATION

The profile page contains the student attendance generated at a particular lesson time. The design flow of this teacher attendance application can be seen in Figure 5. This figure shows the flowchart of the teacher application. After creating

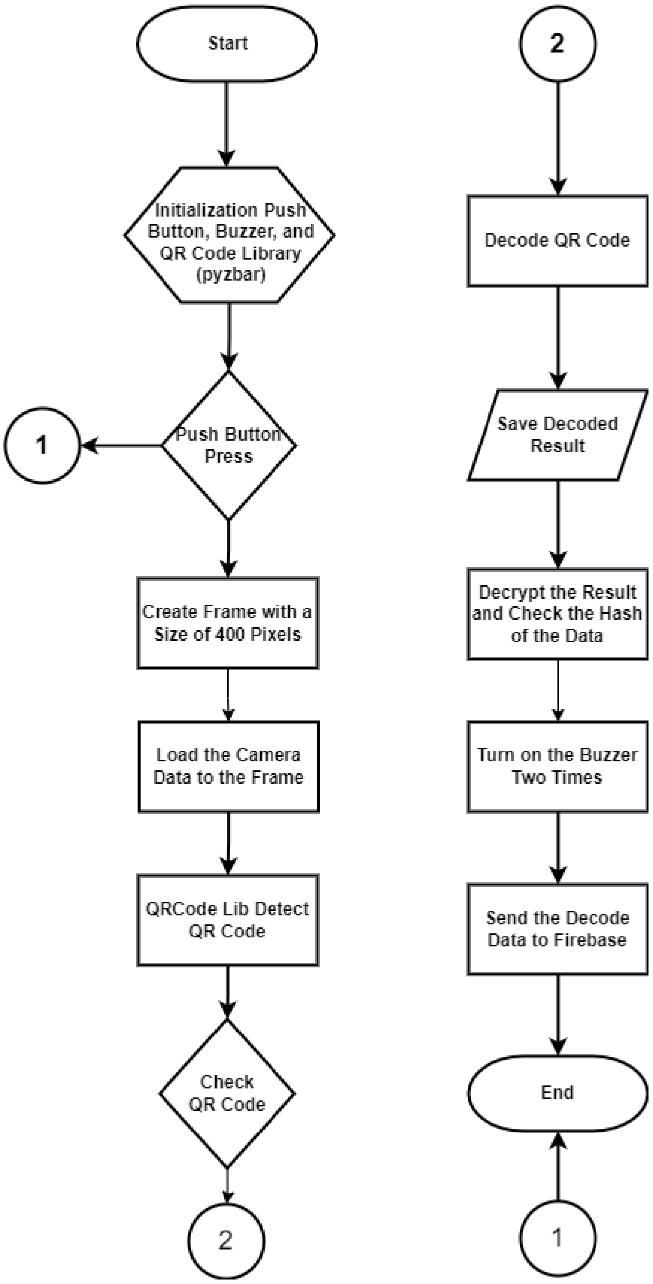
Start
Initialization Push Button, Buzzer, and QR Code Library (pyzbar)
Push Button Press
1
Create Frame with a Size of 400 Pixels
Load the Camera Data to the Frame
QRCode Lib Detect QR Code
Check QR Code
2
2
Decode QR Code
Save Decoded Result
Decrypt the Result and Check the Hash of the Data
Turn on the Buzzer Two Times
Send the Decode Data to Firebase
End
1

Figure 4. Flowchart of Raspberry Pi program.

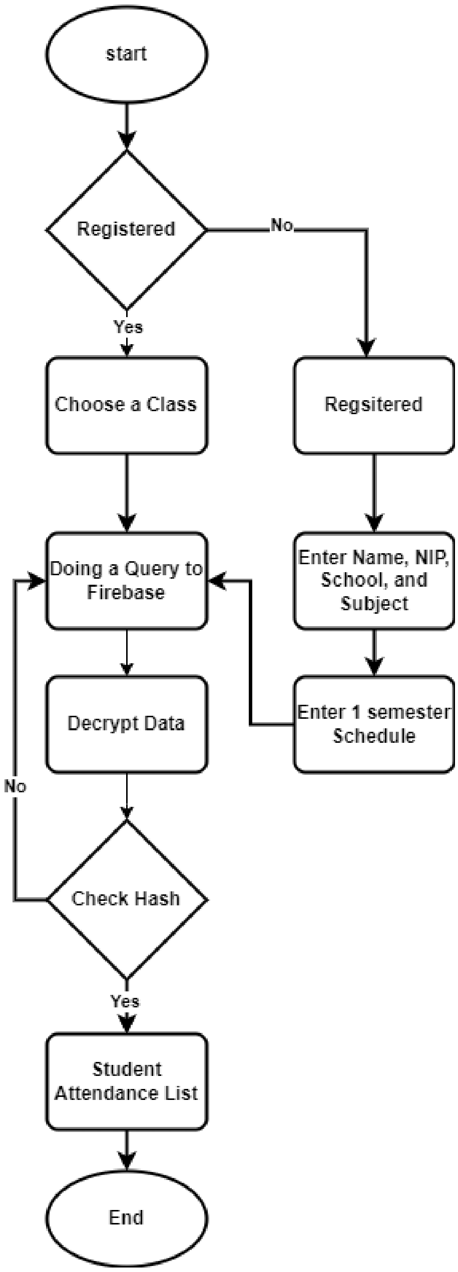

Figure 5. Flowchart of the teacher attendance application.

an account, teachers could generate attendance reports and lesson schedules. The attendance report data were acquired from a query made from Firebase. Once the teacher application received the data, the data were decrypted using the private key stored in the application. If the student data were not decrypted successfully or the data did not match the schedule specified by the teacher, the data were considered invalid and the student was not listed in the attendance report.

Student data in the teacher application were verified utilizing asymmetric cryptography and timestamps. Several experimental attack scenarios were created to test the performance of the authentication system, such as duplicating QR code data and constructing student QR codes with lesson schedules. During the implementation of the system, it was found that the application could not read some data, even though they were in the correct format. In addition, it was found that the encryption feature in QR codes could reduce the reading distance of QR codes. This problem was solved by rescanning the QR code on the scanner during class.

E. THE SECURITY SYSTEM

The security system used in this research was the RSA algorithm and SHA-1. The RSA algorithm was used to guarantee data confidentiality, while SHA-1 was used to guarantee data integrity. The message embedded in the QR code was implemented in the student mobile app, which is described below:

1. Select N, where N is the size of the QR code image with a width of 350 and a height of 350.
2. Select any message generated from three fields: username, student name, and hashed password.
3. Generate the payload *ms*, encrypt it with the RSA algorithm, and obtain the ciphertext *S* using the public key from the Ku server.
4. Then, generate a QR code using *N*, *S* + *Tn*, where *Tn* is the timestamp of each class.

The QR code extraction process was implemented on the Raspberry Pi and the teacher mobile application, which is described below:

1. Read the received QR code to get $S + Tn$ by determining the QR code dimension N.
2. Extract S and Tn, then decrypt S using the private key from Kp server.
3. Authenticate each field of the data in the database.
4. If the data is successfully authenticated, send S to the database.
5. The teacher mobile application can retrieve S from the database and decrypt it using Kp, then display it on the teacher application.

F. HARDWARE DESIGN

This research used hardware in the form of a Raspberry Pi to decode QR data from the camera, decrypt the data, and perform the QR code authentication process. Valid authentication data were then sent to the database. Raspberry Pi was also equipped with several other features, including a marker to ensure whether the QR code could be read/authenticated using a buzzer and a reset button to reset the program on the Raspberry Pi in the event of problems.

G. TESTING SCENARIO

Performance testing was performed on Android devices and Raspberry Pi devices to assess application performance after the application of this authentication method.

1) ANDROID PROGRAM TESTING

Android application testing was divided into two types: testing program execution time and power consumption using simulation software in Android Studio. Testing was conducted 30 times in total. The login menu and attendance report were the two program elements measured by execution time. The execution time of each program component was summed up in program execution testing, and power consumption testing was done using QR codes from actual students.

2) RASPBERRY PI PROGRAM TESTING

Testing was carried out on the Raspberry Pi to measure the performance of the QR code reading process and the encryption process. The parameters used to measure Raspberry Pi performance were execution time and memory usage. The third-party application used to measure program execution time was the timeit module, while the top application was used to measure memory usage.

III. RESULT AND DISCUSSION

This paper aims to develop a secure QR code image for each student at MA Al-Mukhlishin Bandung for each class they attended. The system is similar to [22], which developed dynamic QR codes to prevent attacks on QR codes. The difference lies in the use of hardware and the algorithm. More expensive hardware and standardized encryption algorithms for finance were implemented in [22]. Meanwhile, the device in this research was tested based on the parameters of computational performance, optimal time, and distance the system could work. The interface of this application is presented in Figure 6.

When the class started, students scanned the secure QR code image generated by each smartphone. The unique QR Code was captured using the camera module, and the Raspberry Pi processor interpreted the data. These functions were implemented in the student Android app, teacher Android app, database, and Raspberry Pi, as shown in Figure 6.

Figure 6. Implementation of the system.

A. RASPBERRY PI DEVICE IMPLEMENTATION

In the implementation stage, the Raspberry Pi could read secure QR code images generated by student applications. The Raspberry Pi functions as follows:

1. The Raspberry Pi will detect the presence of a QR code from the camera module by using the prybar library.
2. The pyzbar library will decode the QR code image into a string format that has been encrypted using the RSA encryption method. The barcode scan results are provided in CSV format for debugging purposes.
3. The decoded QR code is in the form of ciphertext with the following format: username#StudentName#hash_sandi#
4. The password will be decrypted using the RSA private key, and the Raspberry Pi will request username and password data on Firebase and verify them. If the verification result is successful, the buzzer will signal that the data has been confirmed.
5. Finally, the Raspberry Pi will send a timestamp to Firebase to record student attendance.

B. THE RESULT OF ANDROID DEVICE TEST

Performance testing of the Android program was carried out after the implementation stage to measure system performance. This test aims to determine the speed of the authentication process that has been implemented. In this test, the speed measured was the speed of the Android application process in carrying out the authentication process as well as the amount of energy required to carry out the student attendance authentication process. This measurement was done with the help of Android Studio software to measure each attendance process. The test was conducted ten times, then the average of the measurement process was calculated to represent the data. The test results that determine the duration of program execution time are shown in Table I.

It can be seen that there is no time difference in displaying student attendance data in the Android application. When the Android program created an encrypted QR code, the QR code with 1024-bit RSA encryption was 0.36 s slower than that of without encryption and 0.42 s faster than that of 2048-bit RSA encryption. The amount of energy required for an encrypted QR Code using RSA-2048 was 57.4 mJ more compared to the encryption process with RSA-1024, and required 88.5 mJ more

TABLE I
TEST RESULTS PROGRAM EXECUTION TIME DURATION

Test Items	Results		
	Without Encryption	*RSA-1024 bit*	*RSA-2048 bit*
Time displays attendance students by teachers	1.661 s	1.66 s	1.66 s
Time of displaying QR Code by students	0.97 s	1.33 s	1.76 s
Energy consumption to encrypt student's data	0.10 J	0.14 J	0.19 J

TABLE II
QR CODE READING TEST RESULTS

No	Test Items	Results					
		5 cm	*6 cm*	*8 cm*	*11 cm*	*13 cm*	*14 cm*
1	Without Encryption	Unreadable	Readable	Readable	Readable	Readable	Readable
2	RSA-1024 bit	Unreadable	Readable	Readable	Readable	Readable	Unreadable
3	RSA-2048 bit	Unreadable	Readable	Readable	Readable	Unreadable	Unreadable

power compared to the data encryption process without using QR codes.

C. RASPBERRY PI DEVICE TEST

After the implementation stage, performance testing of the Android program was carried out to measure system performance. Program performance testing was carried out to determine the execution of the program and determine the appropriate encryption to secure this QR code-based attendance system.

1) TESTING THE READABILITY OF QR CODE BY THE RASPBERRY PI

This QR Code reading test measured the ability of the Raspberry Pi to read the QR code based on the length of the encryption key. The results of the QR code reading test can be seen in Table II.

Table II shows that the minimum distance to read a QR code was 6 cm for all methods used, while the reading distance range for each method had a difference of 1 cm with RSA-2048 bits having the lowest reading distance range of 5 cm, at a distance of 6–11 cm.

2) QR CODE READING PERFORMANCE TESTING BY THE RASPBERRY PI

This QR code performance test compared program performance based on the length of the encryption key applied to the QR code. The results of the performance testing of the Raspberry Pi program can be seen in Table III.

Table III shows the results of measuring Raspberry Pi performance using three parameters: CPU usage, memory usage, and execution time. In measuring the CPU usage parameter, the encryption process made the CPU work harder and longer than using RSA. It is shown by the increased CPU usage, which was greater than 20% compared to RSA encryption, and the encryption process differed between 2–4 s when using RSA encryption. However, the memory usage between programs with RSA encryption and only RSA varied by 2–4 MB when using the RSA algorithm.

TABLE III
RASPBERRY PI READING PERFORMANCE TEST RESULTS

No	Test Items	Results		
		CPU Usage (%)	*Memory (MB)*	*Execution Time (s)*
1	Without Encryption	6	130.9	4.261
2	RSA-1024 bit	28	132.9	6.285
3	RSA-2048 bit	35	134.1	8.452

As compared to [22], the SM3+SM2 algorithm had a faster processing time of 2.841 times/s in executing the authentication process, whereas it was almost three times faster than the proposed system. This lower performance was due to the hardware used by [22], which was more powerful than the hardware proposed in this system. In addition, the SM3+SM2 algorithm was faster than the RSA algorithms and SHA-1. However, through further examination, the use of hardware such as that used by [22] is unnecessary due to its excessive cost and the presence of many unnecessary interfaces.

IV. CONCLUSION

An authentication system for QR code-based attendance systems using RSA and hashing algorithms has been developed to overcome the vulnerability in student attendance applications, where students can forge their QR code to make them recorded as present without coming to class. Examples of a security standard to address this security issue are the use of cryptography, steganography, and the addition of a security layer to the QR code message. This research proposed a public key encrypted QR code using the RSA algorithm. Moreover, SHA-1 function was added to guarantee message integrity.

Several performance parameters, including execution time, QR Code reading distance, and program performance, were used to measure the performance of this security system. Based on the research that had been done, the implementation of the QR code RSA encryption algorithm-based security system for the attendance can function properly and in accordance with the design that has been realized. The implementation of the security system in the QR code application for student attendance and the RSA algorithm in the QR code-based application was applied to three devices: the student Android application device, the teacher Android application device, and the Raspberry Pi device. Teacher Android application devices was used to display validated student attendance data; student Android application devices were used to generate encoded QR codes; and Raspberry Pi devices were used to scan student QR codes.

Raspberry Pi devices and presence-based applications could carry out the authentication process properly. Raspberry Pi devices and QR code-based applications could also communicate with Firebase cloud servers properly. The maximum reading of the QR code was 13 cm, while the minimum reading distance was 6 cm. The optimal reading distance was between 8 cm and 11 cm.

The use of the RSA encryption method and hashing algorithm can prevent students from making illegal absences, which can be done by using fake QR code generation attacks. The addition of security protocols to QR code attendance can result in decreased system speed. At each stage of the procedure, the decrease in speed was no more than 1 s, and the total delay generated in the process was still less than 12 s. Based on ITU-T G.1010, the target delay for massive data transfer is 15 s, so it can be said that the method used is still in accordance with the ITU-T data transfer target.

Besides, the addition of security protocols to the QR code presence increases the consumption of energy on the Android device by 0.09 J. If it is converted to mAh at 5 V, its value is 0.0022 mAH, meaning that it only reduces the 4,500 mAHh battery by no more than 1.76% every hour.

Attendance security design with QR code, RSA, and hashing algorithms will be more precise and faster if using a symmetric encryption algorithm such as AES in the authentication process. However, the use of symmetric encryption algorithms requires modifications to the key distribution process to be more complex for each classroom and subject to improve the security system.

CONFLICTS OF INTEREST

The researcher declares that the article entitled “Implementation of QR Code Attendance Security System Using RSA and Hashing Algorithm” is free from conflicts of interest.

AUTHORS’ CONTRIBUTIONS

Methodology and implementation, Arif Indra Irawan; writing—original draft preparation, Maya Rahayu; writing—reviewing and editing, Istikmal; project management, Iman Hedi Santoso; funding acquisition, Iman Hedi Santoso.

ACKNOWLEDGMENT

This work was only possible thanks to financial support from the Directorate of Research and Community Service at Telkom University. In addition, thanks to all individuals, institutions, and companies that cannot be mentioned one by one. Thank you for your help in successfully completing this research.

REFERENCES

[1] V. Uzun, “QR-code based hospital systems for healthcare in Turkey,” *2016 IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, 2016, pp. 71–76, doi: 10.1109/COMPSAC.2016.173.

[2] M.E. Çoban, B. Çubukçu, R. Yayla, and U. Yüzgeç, “Raspberry Pi based robot application using QR code: QR-Robot,” *2019 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, 2019, pp. 119–123, doi: 10.1109/UBMK.2019.8907129.

[3] A.D.B. Sadewo, E.R. Widasari, and A. Muttaqin, “Perancangan pengendali rumah menggunakan smartphone Android dengan konektivitas Bluetooth,” *J. Pengemb. Teknol. Inf. Ilmu Komput.*, vol. 1, no. 5, pp. 415–425, May 2017.

[4] P. Tilala, A.K. Roy, and M.L. Das, “Home access control through a smart digital locking-unlocking system,” *TENCON 2017-2017 IEEE Region 10 Conf.*, 2017, pp. 1409–1414, doi: 10.1109/TENCON.2017.8228079.

[5] S. Tiwari, “An introduction to QR code technology,” *2016 Int. Conf. Inf. Technol. (ICIT)*, 2016, pp. 39–44, doi: 10.1109/ICIT.2016.38.

[6] M.S. Akbar *et al.*, “Face recognition and RFID verified attendance system,” *2018 Int. Conf. Comput. Electron. Commun. Eng. (iCCECE)*, 2018, pp. 168–172, doi: 10.1109/iCCECOME.2018.8658705.

[7] E. Susanto, D. Perdana, A.I. Irawan, and R. Yasirandi, “Pengembangan sistem presensi menggunakan quick response code dinamis untuk Madrasah Aliyah Al Mukhlisin Bandung,” *J. Rekayasa Elekt.*, vol. 15, no. 2, pp. 139–144, Aug. 2019, doi: 10.17529/jre.v15i2.13769.

[8] K.S.C. Yong, K.L. Chiew, and C.L. Tan, “A survey of the QR code phishing: The current attacks and countermeasures,” *2019 7th Int. Conf. Smart Comput. Commun. (ICSCC)*, 2019, pp. 1–5, doi: 10.1109/ICSCC.2019.8843688.

[9] A. Averin and N. Zyulyarkina, “Malicious QR-code threats and vulnerability of blockchain,” *2020 Glob. Smart Ind. Conf. (GloSIC)*, 2020, pp. 82–86, doi: 10.1109/GloSIC50886.2020.9267840.

[10] “Hubungan antara QR code dan dunia industri dan perdagangan,” Pusdiklat Industri, 2020.

[11] T.M. Fernandez-Carames and P. Fraga-Lamas, “A review on human-centered IoT-connected smart labels for the Industry 4.0,” *IEEE Access*, vol. 6, pp. 25939–25957, 2018, doi: 10.1109/ACCESS.2018.2833501.

[12] L. Tan *et al.*, “Visual secret sharing scheme for color QR code,” *2018 IEEE 3rd Int. Conf. Image Vis. Comput. (ICIVC)*, 2018, pp. 961–965, doi: 10.1109/ICIVC.2018.8492724.

[13] S. Liu, Z. Fu, and B. Yu, “Rich QR codes with three-layer information using Hamming code,” *IEEE Access*, vol. 7, pp. 78640–78651, Jun. 2019, doi: 10.1109/ACCESS.2019.2922259.

[14] N.V. Akhil, A. Vijay, and D.S. Kumar, “QR code security using proxy re-encryption,” *2016 Int. Conf. Circuit Power Comput. Technol. (ICCPCT)*, 2016, pp. 1–5, doi: 10.1109/ICCPCT.2016.7530286.

[15] A. Mendhe, D.K. Gupta, and K.P. Sharma, “Secure QR-code based message sharing system using cryptography and steganography,” *2018 1st Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, 2018, pp. 188–191, doi: 10.1109/ICSCCC.2018.8703311.

[16] V. Malathi, B. Balamurugan, and S. Eshwar, “Achieving privacy and security using QR code by means of encryption technique in ATM,” *2017 2nd Int. Conf. Recent Trends Chall. Comput. Models (ICRTCCM)*, 2017, pp. 281–285, doi: 10.1109/ICRTCCM.2017.36.

[17] P.-Y. Lin and Y.-H. Chen, “QR code steganography with secret payload enhancement,” *2016 IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2016, pp. 1-5, doi: 10.1109/ICMEW.2016.7574744.

[18] Y.-M. Wang *et al.*, “Secured graphic QR code with infrared watermark,” *2018 IEEE Int. Conf. Appl. Syst. Invent. (ICASI)*, 2018, pp. 690–693, doi: 10.1109/ICASI.2018.8394351.

[19] L.F. Freitas, A.R. Nogueira, and M.E.V. Melgar, “Visual authentication scheme based on reversible degradation and QR code,” *2020 4th World Conf. Smart Trends Syst. Secur. Sustain. (WorldS4)*, 2020, pp. 58–63, doi: 10.1109/WorldS450073.2020.9210412.

[20] M. Alajmi, I. Elashry, H.S. El-Sayed, and O.S.F. Allah, “Steganography of encrypted messages inside valid QR codes,” *IEEE Access*, vol. 8, pp. 27861–27873, Feb. 2020, doi: 10.1109/ACCESS.2020.2971984.

[21] I. Tkachenko *et al.*, “Two-level QR code for private message sharing and document authentication,” *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 3, pp. 571–583, Mar. 2016, doi: 10.1109/TIFS.2015.2506546.

[22] Y. Zhou, B. Hu, Y. Zhang, and W. Cai, “Implementation of cryptographic algorithm in dynamic QR code payment system and its performance,” *IEEE Access*, vol. 9, pp. 122362–122372, Aug. 2021, doi: 10.1109/ACCESS.2021.3108189.

[23] A.G. Konheim, *Computer Security and Cryptography*. Hoboken, USA: John Wiley & Sons, 2007.

[24] Y. Zhao, Y. Li, and S. Wang, “Asymmetric deep hashing for person re-identifications,” *Tsinghua Sci. Technol.*, vol. 27, no. 2, pp. 396–411, Apr. 2022, doi: 10.26599/TST.2021.9010014.

[25] T. Kleinjung *et al.*, “Factorization of a 768-Bit RSA Modulus,” in *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed., Heidelberg, Germany: Springer Berlin, 2010, pp. 333–350, doi: 10.1007/978-3-642-14623-7_18.