International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 2
ISSN: 2088-5334

Authentication System and Method for Improving Security Login without Typing Password

Irfan Darmawan[a], Alam Rahmatulloh[b1], Rianto[b2], Ilman Hilmi Oriza[b3]

[a] *Department of Information System, Telkom University, Bandung, Indonesia*
E-mail: irfandarmawan@telkomuniversity.ac.id

[b] *Department of Informatics, Siliwangi University, Tasikmalaya, Indonesia*
E-mail: [1]alam@unsil.ac.id; [2]rianto@unsil.ac.id; [3]ilmanhilmioriza@gmail.com

***Abstract*—Authentication in the login process is an important thing that needs attention. The login process will involve a password that is owned by the user, while the password is private and confidential. If someone uses a weak password, the password is likely to be easily hacked. Authentication security needs to be improved, and hackers will get access to the login system with only a few attack techniques such as SQL Injection or sniffing techniques. Besides, the lack of awareness of users by creating weak passwords is easy to guess. Meanwhile, to create a strong password, consisting of upper- and lower-case letters, a combination of numbers and symbols, it is very difficult to remember. This is a very important problem in the login process. This study discusses the login authentication process that can perform login integration without typing a password, because passwords are generated repeatedly with the One Time Password (OTP) method, and use the Quick Response Code (QR) as its support. To disguise the data in the QR Code, which is applied by the Rivest-Shamir-Adleman (RSA) encryption algorithm, and will be tested on a web-based application. The login integration process, using the QR Code token application that runs on an android phone. Which functions as an OTP token generator, and a web-based application will read information from the QR Code token. The result is that with login authentication, this can increase the security and ease of the authentication process without typing a password.**

***Keywords*—authentication; login; One Time Password (OTP); password; Quick Response (QR) code.**

I. INTRODUCTION

System owners and managers often overlook security problems. Often security is second, or even last, login systems that use databases as users and password authentication are very vulnerable to hacking. Many techniques and methods of attack carried out in hacking, including the most popular one is Injection. SQL Injection is one of the attack techniques used to gain unauthorized access to the system database through login authentication [1]. Other attack techniques, such as sniffing work, when passwords are not encrypted before being sent to a server where hackers can track passwords or user data [2]. Security problems will still be there because security is inversely proportional to comfort. Users generally prefer the convenience of remembering passwords that are easy compared to using strong passwords (security), so this causes weaknesses in a system. Weaknesses in the authentication process by using a password, namely: a combination of weakness, the ability to remember, keylogging, eavesdropping, surfing on the shoulder, and reusing (linkage) [3]. Various studies have been carried out to design an authentication process that is safer but also remains efficient; one example is the authentication of hotspot login using the uniqueness of body parts such as fingerprints [4]. This authentication has been useful in providing security, not yet compiled in implementation, constrained by users with expensive device costs, and the compilation process is hindered by dust or tissue, so it cannot be done. Other authentication methods such as authentication using smart cards [5], this authentication is also proper, but the system does not know if this authentication tool changes hands, so that others can easily use it to process authentication.

In general, authentication only relies on passwords, but to become stronger and harder to read (obfuscate) [6] it can be combined with other methods such as relying on items that are unique and often used like cell phones. The combination of the two will function as a password generator that can be used occasionally, which is commonly known as One Time Password (OTP). In general, OTP is made randomly using a hash function where the password is only valid for a single or one-time login session [7]. There are two modes of use of

tokens, namely the challenge mode where this mode provides a challenge server series of numbers that must be entered into the system to generate tokens, and the second is a self-generated mode where this mode the server does not provide a challenge to the system will generate tokens directly periodically depends on the time when the system is asked to produce a token [8]. The technology used to implement the One Time Password (OTP) method uses the Quick Response (QR) Code technology. The QR Code can store all types of data, can also be scanned from several angles to 360^{o}, and is resistant to damage [9].

Other problems QR Code files are easy to read using several QR scanner applications. So we need a mechanism or algorithm to disguise and secure the data. Rivest-Shamir-Adleman (RSA) is an excellent solution to overcome security and data confidentiality problems, RSA is considered as a secure cryptographic algorithm because the longer a public key is, the more effort that must be spent to solve the key [10]. So, in this study will try the integration of login with a new mechanism utilizing the QR Code in the mobile application that is encrypted using RSA as the login key in the web-based application.

II. Materials and Method

A. Related Works

1) Existing Login System: Reference [3] have implemented two-factor authentication with QR code on web-based applications. This research resulted in a web server application in the form of a prototype of a simple banking authentication application, and a mobile application that functions as a token in the authentication process for the webserver application. Testing of the results of implementing two-factor authentication with a QR code in the case of Internet banking authentication also shows that this method can overcome attacks in the form of system sniffing.

Reference [11] has succeeded in developing a plugin and smartphone application for login system integration that can authenticate a device without typing and remembering using AES default encryption. Integration of the login system without typing passwords can improve account security and can reduce the risk of man-in-the-middle attacks and keyloggers. The password only needs to be filled in the first-time login; for the next login process, the user does not need to fill in the login information again. Android application development has resulted in a password manager application that makes users do not need to remember passwords and have secure password storage media.

Reference [12] has successfully developed a secure login system using a mobile device. The research method can eliminate threats arising from keylogger software, shoulder surfing, and cursor tracking so that the login procedure in this study can be secure in an unknown device or on a public computer.

Innovation in the research that will be carried out is the QR Code generation process is in the mobile application, and the reading process is in the Web application. So that it is expected that QR Code protection can be more guaranteed, then testing added reverse engineering attack techniques to mobile applications.

2) Threats and Weaknesses of Authentication

- *Keylogger.* Keylogger is a type of spyware that aims as a spy who can record activities on the keyboard without being noticed by the user [13]. Data from the recording is then sent to a specific address.
- *Shoulder Surfing.* Shoulder Surfing is a type of attack with direct observation techniques, for example, observing when someone fills in the login form [14]. This attack is usually used to get passwords, pins, or security codes [15].
- *Dictionary Attack.* A dictionary attack is an attack *technique* using dictionary passwords that are already available, and this attack is usually successful against weak passwords [16].
- *Network Sniffing.* Network Sniffing is the activity of monitoring or tapping all packets that pass-through network traffic. So that data such as usernames and passwords can be seen directly if the network does not use security protocols such as the use of secure socket layer (SSL) [17].

B. Research Method

In general, the method proposed in this study, starting from the generation stage and QR Code Token reading, can be seen in Fig. 1.

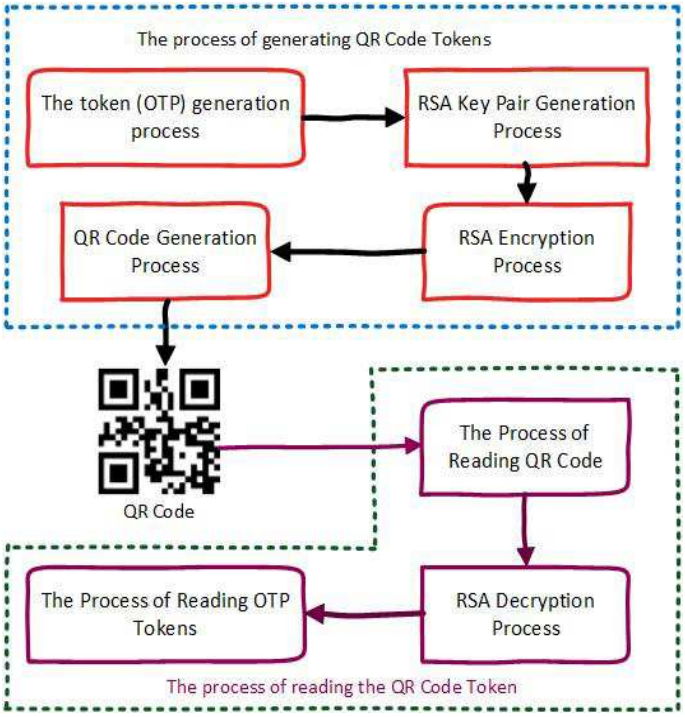

Fig. 1. The proposed method of integrating QR Code login authentication

The first step in Fig. 1 is the OTP token generation process, both the RSA key pair generation process, the third RSA encryption process, the fourth QR Code generation process, until here the output is in the form of a QR Code, then the fifth stage is the QR Code reading process, six RSA decryption processes, and the last is the process of reading OTP tokens.

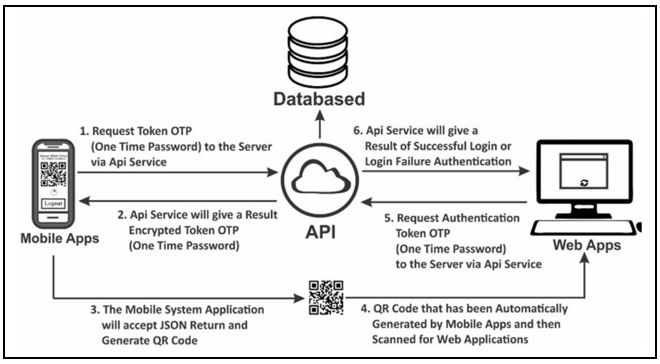

Fig. 2. Integration Login System Architecture

Fig. 2 is an architecture of login integration and interaction between devices. The login integration system created is a web-based application and an Android-based application. The web application functions as a client that will receive authentication data from the android mobile application by scanning a QR Code. While the Android mobile application is a token generator engine generated on a QR Code. The QR Code will be randomized and changed every 30 seconds. The login system architecture consists of several devices that are bridged by the Application Programming Interface (API). Furthermore, the API will check the database and provide the results of checking each device.

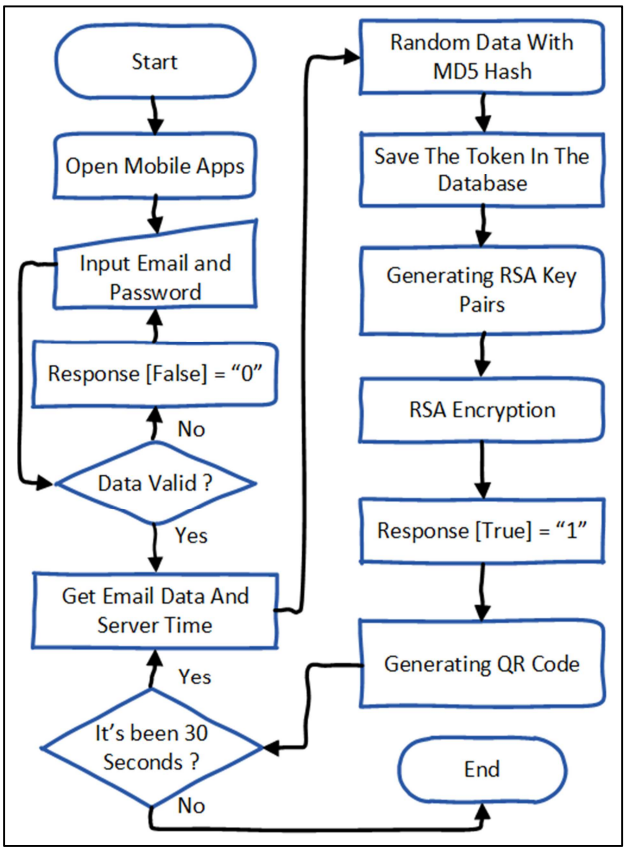

Fig. 3. QR Code Token Generation Flow Diagram

1) QR Code Token Generation Flow Chart: Fig. 3 is a flow diagram of the QR Code Token generation process that occurs in a mobile application, and the generation process is by the stages of the proposed method in Fig. 1 and refers to the document [18], in the system to generate tokens required the initial process is inputting email and password, the process is carried out to validate which users are requesting to generate tokens so that the generated tokens are right for users who request the generation process. The email and password input process are only done once; for the next process, users do not need to authenticate, because the generation process will be carried out automatically by the system, within 30 seconds each generation. Stages of the process are carried out, starting from the initial security of the application by authenticating the login using email and password. If valid, then proceed with retrieving email data and time on the server. The data is used as token data that is encrypted by MD5 Hash (Message-Digest algorithm 5). Then the Rivest-Shamir-Adleman (RSA) key generation and the RSA encryption process are performed. The process continues to be carried out simultaneously and is checked every 30 seconds.

2) QR Code Token Reading Flow Chart:

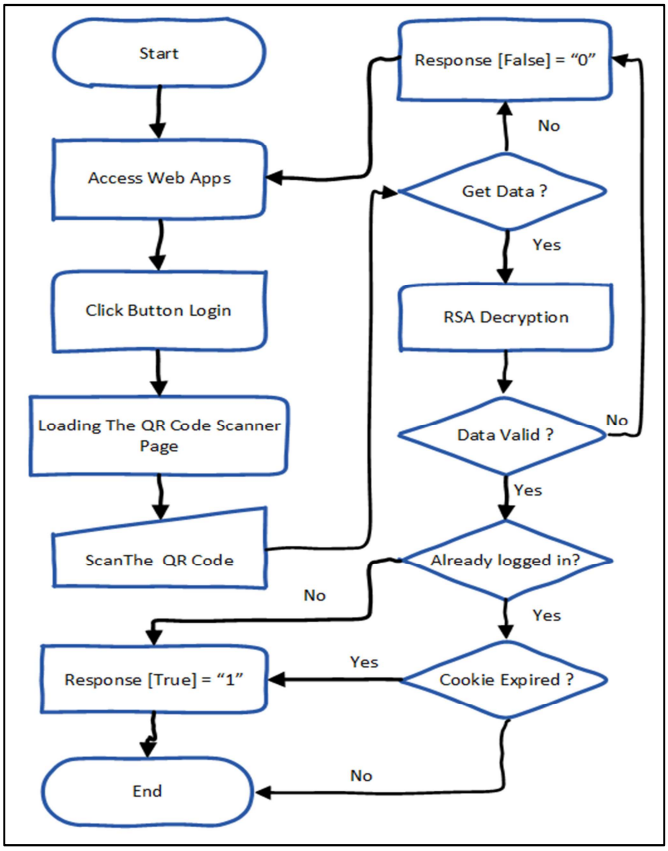

Fig. 4. QR Code Token Reader Flow Diagram

Fig 4. is a flowchart of reading QR Code Tokens that occur in a web application, the reading process is following the stages of the proposed method in Figure 1. The reading process will give a response 1 or login successfully if the QR Token that has scanned data is valid, the session/cookie user status is not yet never logged in. If the user is logged in, then to be able to log in on another device must wait for the user's cookie to expire. So, at this stage, the checking is carried out four times. Starting from checking the data, then whether the data is valid. Followed by checking the status of the user is logged in or not. Finally, check whether the user's cookie has expired or not.

III. Result and Discussion

The mobile application is programmed using java language, and for web applications using PHP and HTML programming languages. For generating QRCode, using the Zxing library.

A. Testing

1) Functional Testing:

TABLE I
FUNCTIONAL TESTING RESULTS

No	Description	Response
1	Mobile login authentication with incorrect email and password input.	Rejected
2	Mobile login authentication with correct email and password input.	Accepted
3	OTP token generation.	Accepted
4	RSA key pair generation.	Accepted
5	Token encryption with RSA encryption.	Accepted
6	QR Code generation.	Accepted
7	QR Code Reading.	Accepted
8	QR Code reading with timeout valid.	Rejected
9	QR Code reading with a token that has been used.	Rejected
10	Reading the QR Code in a different browser when users have not logged out on the first browser.	Rejected
11	Decrypt token with RSA decryption.	Accepted
12	OTP token reading.	Accepted
13	OTP token reading with timeout valid.	Rejected
14	OTP token reading with the token that has been used.	Rejected
15	OTP token reading on the browser is different when users have successfully logged in to the first browser.	Rejected

In table 1 is the result of functional testing that is by testing the correct input data and wrong data, then see the response of the system. The desired hope is that the response from the system can receive valid data and be able to reject the wrong data.

2) QR Code Token Reader Testing: The results of the QR Code reading experiment 10 times, using different tokens, produced almost the same experiment, one of the experiments was:

TABLE II
QR CODE TOKEN READER TESTING RESULTS

Token	QR Token Time	QR Token	Session Users Status	Result
c6a016	Valid	Not used	Logout	Successfully
	Valid	Not used	Login	Failed
	Valid	Used	Logout	Failed
	Valid	Used	Login	Failed
	Expired	Not used	Logout	Failed
	Expired	Not used	Login	Failed
	Expired	Used	Logout	Failed
	Expired	Used	Login	Failed

From the results of the experiment in Table 2 shows that the QR Token will be valid when "QR Token time is still valid or valid" then "QR Token is not used or not used" and "Status session users have logged out or are not using a web application."

3) Testing Network Sniffing Attack: The tool used to test the attack technique of tapping data packets (network sniffing) using Wireshark software in this test will be tested whether the QR Code Token before authentication has been successfully encrypted or not.

Fig. 5. Initial Attack of Network Sniffing

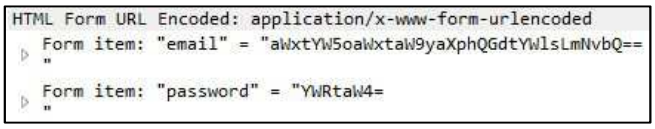

Fig. 6. Results of Network Sniffing Login Mobile Attack

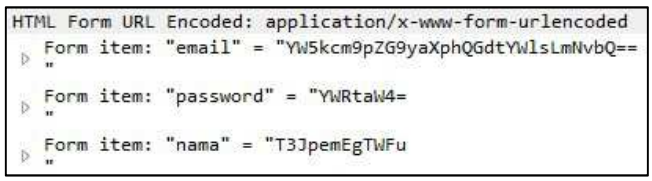

Fig. 7. Results of Mobile Registration Network Sniffing Attacks.

Fig. 5-7 is a network sniffing attack test. The filtering results of intercepting the data packets that have been carried out can be seen in Fig. 5. The result is a data packet with the HTTP protocol that refers directly to the target URL destination 192.168.43.55/web/service/ so that the initial tapping to get the target URL has been successful. The results from the intercepts can be seen in Fig. 6 and Fig. 7. From the test results that can be seen in Fig. 5-7, which shows that the data was successfully protected by applying an encryption algorithm. Although the data can be stolen, it cannot be read because it has been disguised, and other mechanisms must be taken to unpack or decrypt the data.

4) Testing Reverse Engineering Attack Techniques in Mobile Applications: Reverse engineering testing is carried out only to test when installing applications on rooted mobile phones and to decompile the project. Aims to secure java files that contain program functions. The tools used are Show Java and WinRAR.

Fig. 8. Rooted device detected

Fig. 8 is a display that appears when a user successfully installs a mobile application and is detected as rooted users. Even though the cellphone has been rooted, the QR Code generates functions that cannot be opened, so the confidentiality of the generation and scrambler as a token remains secure. Applications built can overcome attacks on rooted mobile phones.

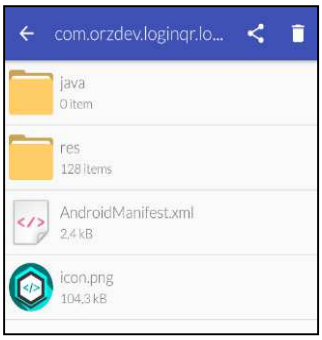

Fig. 9. Reverse Engineering attacks with Java Show Tools

The second test is decompiled in the application installer with the *.APK format. This test uses Java Show Tools by entering application projects in research that has been built into the *.APK packaging format. The results of the project decompile can be seen in Fig. 9

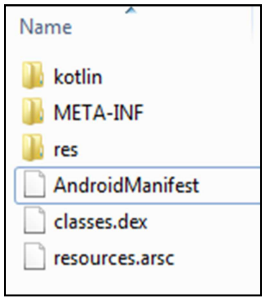

Fig. 10. Attack Reverse Engineering with WinRAR Tools.

Another way to decompile the *.APK project is to convert the *.APK format to *.ZIP format and then do the data extract, and the results are shown in Fig. 10. The result is still the same, and only the file format is *.XML and the java file that contains many functions and classes are unreadable.

B. Login System Integration

1) Web Application:

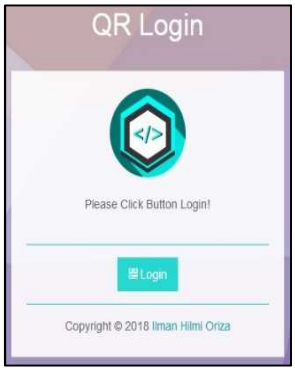

Fig. 11. Start Page of Web Login

Fig. 11 is a web application page view. On this initial page, the application only has the login button function, and the login button function is to open the QR Code scanner page. When the login button at the beginning of the page is clicked, then the QR Code scanner page appears (Fig. 12).

Fig. 12. QR Code Scanner page.

On this page users only scan the webcam of each device that users use to open a web page, then the next process will be carried out by the system to capture the scanned data and send it to the API made to decrypt and process data authentication, then API will give a response from the results of data authentication.

2) Mobile Android Application: Fig. 13 shows a mobile login page. The mobile login stage is not only to authenticate the data entered in the login, but the login process will process the token generation via the API that has been created. The login process is only done once for the users no longer need to login due to session users will be saved by the system. Before the user enters the correct login data, the token is not generated.

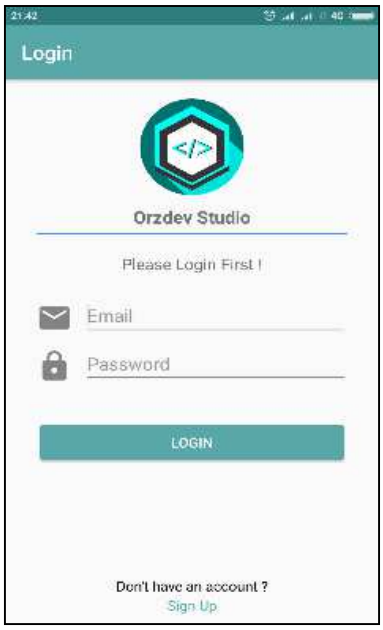

Fig. 13. Mobile Application Login Page

Fig. 14 is a display when the API sends a data authentication response when the authentication data is successful.

Fig. 14. Mobile Application Main Page

Fig. 14 will display the main page of the mobile application. The system will immediately generate the encrypted token into the QR Code image. The time the token will rotate, when the time runs out, the system will request the token again through the API. When the mobile system server has generated the data token, it will immediately re-generate the QR Code image according to the re-generated token data.

IV. Conclusion

Research conducted concluded that QR Code with RSA encryption on the proposed login integration mechanism can improve the security of several attack techniques such as Sniffing, SQL Injection, and Reverse Engineering. The system has overcome the problem of weak passwords and is difficult to remember, thus increasing the ease and security of the authentication process login. With the One Time Password (OTP) method, the handling of QR Code Tokens can be improved. QR Code Tokens can be used if "QR Token time is valid or valid," then "QR Token is not used or not used" and "session user status session does not use a web application."

Also, this research has complemented previous research that has implemented a QR Code generating system on web applications that are more vulnerable to attack. In this study, generate QR Code is done on the Android application and has been tested. Test results show that the application cannot be attacked with reverse engineering or rooted device methods. Further researchers are expected to create a new library from the results of research conducted, can develop QR Code tokens with multi-level users, and can try other cryptographic algorithms for the process of making tokens or the process of encryption and decryption of tokens.

References

[1] E. A. Dharmawan, E. Yudaningtyas and M. Sarosa, "Perlindungan Web pada Login Sistem Menggunakan Algoritma Rijndael," EECCIS, pp. 77-84, 2013.

[2] D. M. Khairina, "Analisis Keamanan Sistem Login," Jurnal Informatika Mulawarman, vol. Vol. 6 No. 2, pp. 64-67, 2011.

[3] R. S. Gusman, "Analisis dan Implementasi Two Factor Authentication dengan QRCode Pada Aplikasi Berbasis Web," UT - Computer Science, pp. 1-22, 2013.

[4] A. D. Tumuli, X. N. Najoan and A. M. Sambul, "Implementasi Teknologi Biometrical Identification untuk Login Hotspot," E-Journal Teknik Informatika, Vols. Vol.12, No. 1, pp. 1-5, 2017.

[5] J. Wei, W. Liu and X. Hu, "Secure and Efficient Smart Card Based Remote User Password Authentication Scheme," International Journal of Network Security, Vols. Vol.18, No.4, pp. 782-791, 2016.

[6] A. Rahmatulloh and R. Munir, "Pencegahan Ancaman Reverse Engineering Source Code PHP dengan Teknik Obfuscation Code pada Extension PHP," in Konferensi Nasional Informatika, Bandung, 2015.

[7] K. I. Santoso, E. Sediyono and S., "Studi Pengamanan Login Pada Sistem Informasi Akademik Menggunakan Otentifikasi One Time Password Berbasisis SMS dengan Hash MD5," Sistem Informasi Bisnis, pp. 7-12, 2013.

[8] I. G. N. A. Jayarana, A. A. K. A. Cahyawan and G. M. A. Sasmita, "Dynamic Mobile Token for Web Security using MD5 and One Time Password Method," International Journal of Computer Applications, Vols. Volume 55-No 6, pp. 1-6, 2012.

[9] A. Rahman and A. Rahmawati, "Sistem Pengamanan Keaslian Ijasah Menggunakan QR-Code dan Algoritma Base64," JUSI Vol. 1, No. 2, pp. 105-112, 2011.

[10] Z. Arifin, "Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi yang Aman," Jurnal Informatika Mulawarman, pp. 7-14, 2009.

[11] M. Arifin, A. Bejo and W. Najib, "Integrasi Login Tanpa Mengetik Password pada Wordpress," JNTETI, Vol. 6, No. 2, pp. 162-167, 2017.

[12] K. Adhatrao, A. Gaykar, R. Jha and V. Honrao, "A Secure Method For Signing In Using Quick Response Codes With Mobile Authentication," International Journal of Student Research in Technology & Management, vol. Vol 1(1), pp. 1-11, 2013.

[13] M. I. Zulfa and E. Subiyanta, "Pemanfaatan Spyware Untuk Monitoring Aktivitas Keyboard Dalam Jaringan Microsoft Windows," Jurnal Emitor, vol. Vol. 15 No. 01., pp. 11-14, 2007.

[14] Y. Kita, F. Sugai, M. Park and N. Okazaki, "Proposal and its Evaluation of a Shoulder-Surfing Attack Resistant Authentication Method: Secret Tap with Double Shift," International Journal of Cyber-Security and Digital Forensics (IJCSDF), vol. 2(1), pp. 48-55, 2013.

[15] M. Kumar, T. Garfinkel, D. Boneh and T. Winograd, "Reducing Shoulder-surfing by Using Gaze-based Password Entry," pp. 1-7, 2007.

[16] Z. Musliyana, T. Y. Arif and R. Munadi, "Peningkatan Sistem Keamanan Autentikasi Single Sign On (SSO) Menggunakan Algoritma AES dan One-Time Password Studi Kasus: SSO Universitas Ubudiyah Indonesia," Jurnal Rekayasa Elektrika, vol. Vol.12 No.1, pp. 21-29, 2016.

[17] M. F. Adriant and I. M., "Implementasi Wireshark Untuk Penyadapan (Sniffing) Paket Data Jaringan," Seminar Nasional Cendekiawan, pp. 224-228, 2015.

[18] 1. ISO/IEC, Information Technology – Automatic Identification and Data Capture Techniques – Bar Code Symbology – QR Code, Swizerland: International Standard, 2000.