Egyptian Informatics Journal 22 (2021) 269–276

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

IoT security system with modified Zero Knowledge Proof algorithm for authentication

Benfano Soewito [a,*], Yonathan Marcellinus [a]

[a] Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

ARTICLE INFO

Article history:
Received 26 February 2020
Revised 30 August 2020
Accepted 6 October 2020
Available online 29 October 2020

MSC:
00-01
99-00

Keywords:
Data transmission
Authentication
Encyption
Zero Knowledge Proof
Discrete logarithm
Modular arithmetic

ABSTRACT

It has been predicted that more devices will be connected to the internet network along with the development of IoT, and that will increase the complexity in the security system. The most important thing in security systems is the process of encryption and authentication. Various encryption techniques are developed to overcome security problems of the data. One of the methods used for authentication of data is Zero Knowledge Proof. This method works to identify the authenticity of someones statement to proof without showing any knowledge of the statement mentioned. This research will mainly discuss about the security of data transmission system by combining data encryption and data authentication. The proposed data encryption is using Advanced Encryption System and method for authentication of data using the Zero Knowledge Proof. This research will conduct the development methods of authentication Zero Knowledge Proof from previous research and the result was compared with the proposed method based on the simulation results transmission system client and server. Experiments will be conducted using thirty-text data, each of data will be measured on the performance of both encryption and authentication process between the previous method and proposed method. Experimental results show the performance of the proposed method has better speed to process application for security of data transmission systems, with performance to authenticate approximately 5 ms from the client side and server side.

1. Introduction

Basically the security system on the internet network is to maintain the data transmitted so that users can be fulfilled elements of confidentiality, integrity and availability. The transmitted data may include statistical data, financial data, privacy data, etc. To ensure the data can be delivered to the right person, needed a security system to secure the data at the time the data is transmitted.

The security system in internet network will be more complicated, after the announcement of Industry 4.0. Because there will be more a lot of devices that will be connected to the internet network including sensors, smart driving cars, smart cameras, smart TVs, Wi-Fi routes, etc. These are all part of the “internet of things” (IoT) innovation wave, which overall promises to greatly improve our lives, if we can deal with the cybersecurity threats they can pose [9]. Industry experts usually define an IoT device as any object connected to the internet (or to a Local Area Connection). According to the Efficient Gov (efficientgov.com), the devices that connected to internet will be more than 50 billion in 2020, as shown in Fig. 1.

There are so many techniques to obtain a victim data privacy through internet [3]. These techniques generally exploit vulnerabilities of the system used by a victim. Password attacks such as dictionary or brute force target a devices login information by bombarding it with countless password and username variations until it finds the right one. Since most people use a simple password these attacks are fairly successful. The topology of IoT system can be seen in Fig. 2, the user can connect to the devices through cloud that also saved all the data. The cloud can be designed according to characteristic of business process [1]. As shown in Fig. 2, the hacker can guess the password to login into cloud, gateway, and IoT control if the password is not strong enough. Not only that, according to one study, nearly 60 percent of users reuse the same password. So if an attacker gets access to one device, they get access to all devices.

* Corresponding author.
E-mail address: bsoewito@binus.edu (B. Soewito).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.10.001
1110-8665/

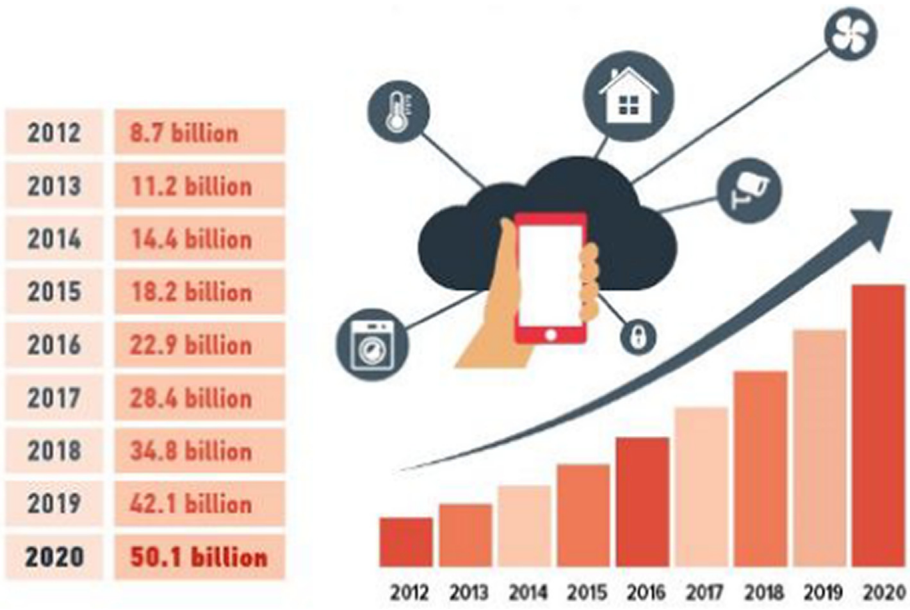

Fig. 1. Number of devices connected to internet.

Another of techniques used is Sniffing. The sniffing application can be found over the internet, download, and install easily. The concept of this technique is to take the data sent by the sender when the data pass through the router or switch as shown in Fig. 3.

In Fig. 3, PC1 sends data to the PC2, the data passing through the switch, if the sniffer successfully retrieve the data on this transmission process, the data can be modified by the sniffer and forwarded back to the PC2. It is an example of the data theft on the PC, and there is still a possibility this could happen on other devices such as smart phones, tab, or devices in IoT systems. Internet of Things (IoT) and Internet of Everything (IoE) are emerging communication concepts that will interconnect a variety of devices (including smartphones, home appliances, sensors, and other network devices), people, data, and processes and allow them to communicate with each other seamlessly.

The smartphone-enabling technologies such as built-in sensors, Bluetooth, radio-frequency identification (RFID) tracking, and near-field communications (NFC) allow it to be an integral part of IoT and IoE world and the mostly used device in these environments [13]. Smartphone is one of technology that is widely used, due to the needs and circumstances that demand for a high level of mobility. According to the research conducted by [6], smartphone also has vulnerabilities in terms of security; Jeske [6] also discusses the security issues on the Google Navigation application and Waze application on a smartphone [6]. Google Navigation and Waze app are intended for real-time traffic monitoring in a region. A hacker can take or send fake data to the server that causes the server indicate the other users that one way or the area is totally jammed and it is very dangerous because the hacker be able to control traffic. More over the data sent is still in a plain text. It is very dangerous if the plaintext data is retrieved by an unauthorized person. In [6], only focus security for this data with Zero Knowledge Proof Authentication system with plaintext. Therefore, in this paper we introduced our algorithm and compared with algorithms that have been developed earlier. Our proposed method can protect data if the unauthorized person get these data by include the encryption technique in our proposed method.

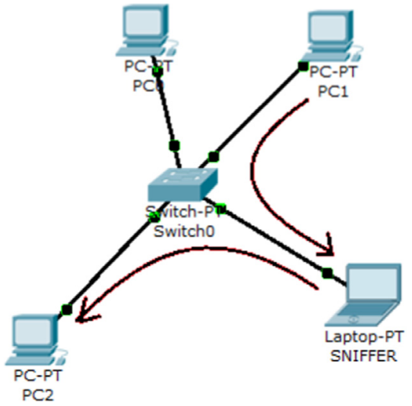

Fig. 3. Diagram of Sniffing process by Laptop SNIFFER.

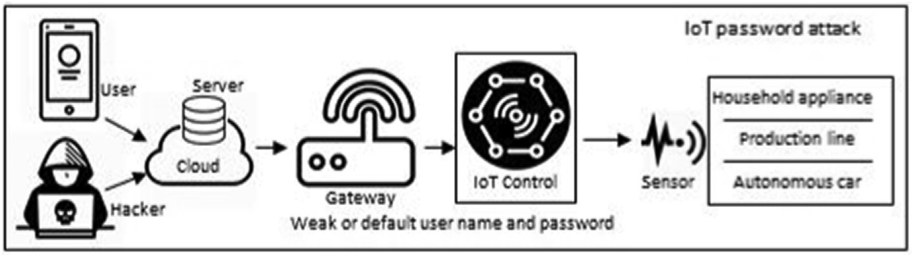

Fig. 2. Topology of IoT system.

2. Background

2.1. Zero Knowledge Proof

Zero knowledge proof (ZKP) is a concept that is very popular and widely used in cryptographic systems. The concept in ZKP, there are two parties involved in the prover and verifier [12]. In the beginning Zero knowledge proof introduced and published by [5] in the paper with titled ”How to Explain Zero-Knowledge Protocols to Your Children”. In this technique, Zero Knowledge Proof allows a prover shows that he has the right or evidence (credential) without showing the actual values to the verifier. Zero knowledge proof system is widely used for authentication because it has properties as follow:

(a) Completeness: If the statement is true, then the verifier will be able to prove that statement true over and over again.
(b) Soundness: If the statement is wrong, there will be no cheating prover to do to be able to convince the verifier that the statement is true. Except with some small chance.
(c) Zero-knowledge: If the statement is true, there will not be any knowledge obtained in addition to the facts. This is demonstrated by a simulation which shows that the statement is true and nothing is gained verifier of the prover.

Jean Jacques and Louis Guillou Q [5] illustrates the zero-knowledge proof by using a story about a cave that has a secret. In this illustration there are two people, Peggy (prover) and Victor (the verifier).

In Fig. 4 is an illustration of Zero Knowledge Proof, Someone who knows the secret key can open the door between C and D. Suppose Peggy wants to prove to Victor that she knows the secret key to unlock the door, then Peggy and Victor need to do is as follows:

a. Victor stands at point A
b. Peggy walked into the cave towards point C or D
c. After Peggy is no longer visible in the cave, Victor walks to point B
d. Victor order Peggy to:

1. Exit from the left lane or the right lane
2. Peggy will respond, and use the keywords that he had to open the secret door, in case he really had.
3. Peggy and Victor would repeat the steps - steps until n times.

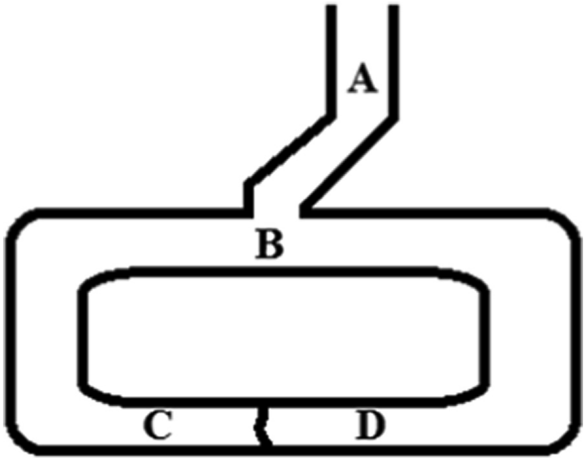

Fig. 4. Illustration of Zero Knowledge Proof.

With this method, Victor would not possible to convince a third party of confidential by Peggy ownership proof and he also can not obtain any information about the secret information that is known by Peggy. This result may not be able to guess correctly Peggy continuously which side Victor will ask her out. Peggy probability guessing would be very small if done repeatedly [10].

2.2. Zero Knowledge Proof of Knowledge discrete logarithms

The application of the Zero Knowledge Proof is done by mathematical calculations, one of calculations is calculation of discrete logarithms. In the paper “Implementing Zero-Knowledge Authentication with Zero Knowledge (ZKA wzk)” [7] discusses the techniques used for web authentication with the calculation of discrete logarithms. The algorithm is based on [2], Zero Knowledge Proof of Knowledge Sigma Protocol using $\mathbf{SPK_1}$ $\{(x)\text{: } Y = go^{X}\}$. There are three processes in this algorithm as follows:

a. Initialization:
1. Given a group **G** and g_0, g_1 obtained from a random element of **G**.
2. Then **G** and g_0 be the public key.
b. Registration:
1. The user enters a username and password.
2. Password was hashwd by a hash function. Obtained x = hashing (password).
3. Users do calculations for Y = g_0^X.
4. Then the user sends (username Y) to the server.
5. Server stores the username Y.
c. Authentication Process:
1. Server generates a random variable a, save it and send it to the client.
2. The user enters a username and password.
3. Client hash password by hash function, and compute x = hash (password).
4. Client calculate Y = g_0^X.
5. Client generates random rx $\in$ G and compute $T_1 = g_0^{rx}$.
6. Client compute c = hash (Y, T_1, a) and $z_x = r_x$ - c x.
7. Client sends (c, z_x) to the server.
8. Server calculates $T_1 = Y^c\ g_0 z_x$ and matching c_2 = hash (Y, T_2, a)
9. If appropriate, the user has been authenticated.

The technique of Zero Knowledge Proof is a common technique used to prove knowledge of the variables associated with the password. Each of the components used in this algorithm are showed in Table 1.

The algorithm can work with the knowledge of the variables associated with the password. The algorithm started by generating G from a number such as 1,2,3,4, .., and g_0 is obtained from the results of random numbers in G. The algorithm was derived using g_0 that came from G, so it is difficult to get a logarithmic discrete calculation. Each component of the Zero Knowledge Proof algorithm is implemented in the client and server systems. In Table 2 shown the variable that are owned by the client and the server.

As shown in Table 3, we can calculate as follows:

$T_1 = T_2$
$g_0^{rx} = Y^c\ g_0^{zx}$
$g_0^{rx} = (g_0^{x})^c\ g_0^{(rx\ -\ cx)}$
$g_0^{rx} = g_0^{cx}\ g_0^{(rx\ -\ cx)}$
$g_0^{rx} = g_0^{(cx\ +\ rx\ -\ cx)}$
$g_0^{rx} = g_0^{rx}$

It can be proved that c = hash (Y, T_2, a) and the user knows the value of x.

Table 1
Components of Zero Knowledge Authentication.

Component	Type	Description
G	Public value	A set of numbers based on a formula.
g_0	Public value	A number that is obtained from G and an element of G.
x	Clients secret value	The result of hashing password which inputted by the prover.
Y	Servers secret value	An alias of provers password used by verifier to calculate and proof of knowledge.
a	Servers secret value	Random tokens that are generated when prover login.
T_1,r_x,z_x,c	Calculation result	Other variables used in calculation.

Table 2
Variable in authentication process.

Prover (User)	Verifier (Server)
g_0	g_0
Password	Y

Table 3
Authentication process.

No.	User (Prover)	Verifier (Server)
1	–	random a
2	receive a	send a
3	Calculate x = hash (password)	–
4	Calculate Y = g_0^x	–
5	get random rx	–
6	Calculate T_1 = g_0^{rx}	–
7	Calculate c = hash (Y, T_1, a)	—
8	calculate zx = rx - cx	–
9	send c, zx	receive c, zx
10	–	calculate T−2 = Y^c g_0^{zx}
11	–	compare if c = hash (Y, T_2, a)

2.3. Advanced Encryption System

In 1997, the National Institute of Standards and Technology (NIST) of the United States announce the Advanced Encryption Standard (AES) to replace the Data Encryption Standard (DES). AES is built with a goal to securing the government data in various departments. A new algorithm needed to embeded in AES. NIST made a contest to choose a new algorithm for AES. In 1998, NIST announced that there were 15 proposals have been received and were evaluated. After a selection process, in 1999, NIST announced that there were only 5 algorithm were chosen as candidate to be the AES, the algorithms are:

1. MARS
2. RC6
3. Rijndael
4. Serpent
5. Twofish

Each of these algorithms undergo a variety of trials. In October 2000, NIST announced that Rijndael algorithm was selected as for the new AES standard. Then on 26 November 2001, NIST announced the final products of the Advanced Encryption Standard [4]. Table 4 shows the difference of variation of each of the proposed algorithm. The length of the key has an effect on Rijndael and Twofish algorithm as shown in 4.

After extensive review of the Rijndael algorithm selected the Rijndael algorithm as the algorithm for the Advanced Encryption System (AES). NIST has announced the Rijndael algorithm has a good level of security, performance, efficient, practical and good flexibility. Rijndael algorithm was developed by John Daemen and Vincent Rijmen from Katholieke University Leuven [4]. According to [8] highly efficient AES encryption used in both hardware and software implementations, has good security and a high rate of speed. In the application of the hardware, AES encryption is very useful especially in the wireless security as well as military communications and mobile telephone.

The evaluation results obtained are based on three characteristics as follow:

1. Security: Include resistance to attack, mathematical complexity, randomness of the output and safety compared with other algorithms.
2. Cost: Includes encryption speed, required memory, and there is no license agreement that algorithm should be available for free with no royalty around the world.
3. Algorithm and implementation characteristics: The algorithm must be applied in a variety of hardware and software systems, and algorithms should be relatively simple.

In Table 5 is shown a comparison of security, speed, and memory of the AES finalists, the comparison was only comparing the four finalists performed by [14].

2.4. Discrete logarithm

Discrete logarithm is a mathematical problem by the following equation a^x mod n = b, where a, n, b, x are positive integer numbers and n is prime [11]. Discreate logarithmic calculation until now is believed to still be very difficult to solve. There is no efficient solution to the calculations used the computer to solve this problem, so that the discrete logarithm algorithm widely used for public-key cryptography. If we want to compute a^x mod n = b, then it will be easy to get the value of b, but if we do the opposite operation to find the value of the exponent (x), it will be more difficult and requires a very long time. Here is an example of the discrete logarithm problem:

a. 329 mod 17 = b
b. Then b definitely worth between 0–16, in this case the answer is 12.
c. Conversely, if 3^x mod 17 = 12, then to find the value of the exponent x would be more difficult than finding the value of b.

Discrete logarithm problem is very difficult to solve, especially when the value of n (primes number) is very large [15]. There are several solutions are used to find the value of the exponent on the discrete logarithm such as the Baby-step giant, function field sieve, the index calculus algorithm and others. These solutions are less effective since there is no solution algorithm that runs in polynomial time.

Table 4
The difference of each algorithm.

Algorithm	Difference	Key Setup
MARS	Key Size 4–39 32-bit words	Constant
RC6	Word size w, No. of rounds r, key size 0–255 bytes	Constant
Rijndael	Block length of 128, 192, or 256 bits	Increasing
Serpent	Key size 0..256 bits	Constant
Twofish	Key size 0..32 bytes	Increasing

Table 5
The difference of security, speed, and memory of AES Candidate.

Algorithm	Security	Speed		Memory	
		Encryption	Key	RAM	ROM
RC6	Adequate	High End	Average	Average	Average
Rijndael	Adequate	High End	High End	High End	High End
Serpent	High	Low End	Average	Average	Average
Twofish	High	Average	High End	High End	Average

2.5. Modular arithmetic

Modular arithmetic is a mathematical system for integers, where numbers wrapped in a limited circle after reaching a certain value (modulus). All numbers are crossing the same point on the point of the circle are congruent. The modern approach to modular arithmetic developed by Carl Friedrich Gauss in his book “Disquisitiones Arithmeticae” published in 1801. Modular arithmetic likes ordinary arithmetic. Several modular operations can be factored as mathematical operations in general. This is very useful especially in the cryptographic verification.

Most of modular arithmetic is used in cryptography, because the calculation of the discrete logarithm problem is difficult. Modular arithmetic is easier done on the computer, because it limits the operation so that the results obtained from the k-bit operations modulo n will not be more than n. One of the methods that can be used to calculate the operations modulo large integers (a^m mod n) quickly is fast exponentiation algorithm [10].

3. Proposed methodology

Our study proposed a method of development of Zero Knowledge Proof in the recognition process the data owner with one of the best encryption algorithm that has become standard in American government, namely the Advanced Encryption System that uses the Rijndael algorithm. Development of the method aims to secure the transmission of data that has been encrypted and can only be decrypted by the person entitled, on this transmission also performed recognition process data sender (authentication). There are two phases to the development of this method, because there is a change in the way of getting the value g_0 from Zero Knowledge Proof methods by Brandon [7]. The two phases of our proposed method are as follows:

a. Registration: At this stage, the sender (client) enter the data required for authentication then performed calculations to get a value that will be sent to the server but not the data of the actual sender. In this condition Zero Knowledge Proof works fine. The steps of proposed methodology for registration process:
1. The user enters a username and password.
2. Password hashing using the MD5 hash function. Obtained x = hashing (password).
3. Value for g_0 and g_1 are obtained from a random value that will hashed using MD5.
4. N value obtained from the random value of prime numbers.
5. Users do calculations for Y = g_0^x mod N (1)
6. Then user send (username, g_0, Y, N) to the server.
7. Server stores the username, g_0, Y, N.
8. Client stores the value of x.

b. Authentication: At this stage will delivery data. First request a session value (used in one session data transmission) and a one-time token (used in a single packet of data sent) were obtained randomly from the server. Furthermore, encryption is applied using the Advanced Encryption System and also calculations for the code that will be inserted into the encrypted data to be transmitted. Then the data is sent to the receiver (server) and performed the authentication process. If valid, the authentication process will be decrypted, if not valid then the data will be rejected. The steps of authentication as follow:
1. Client obtain session ID from server.
2. Server generates random variable a, save it and send it to the client.
3. Client hash the password to the MD5 hash function, and compute x = hash (password).
4. Client calculate Y = g_0^x mod N
5. Client generates random rx values obtained from random value. Then calculate T_1 = g_0^{rx} mod N (2)
6. Client compute c = hash (Y, T_1, A) and z_x = r_x-cx. (3)
7. Client select the data to be sent and do encryption with AES.
8. Furthermore, client combines data that has been encrypted with the value of c, zx, username, sessionID and sent to the destination (server).
9. The server receives data from the client and do the authentication process first. Server calculates T_2 = ((Y^c mod N) (g_0^{zx} mod N)) mod N and matching c = hash (Y, T_2, a) where the value of Y obtained from matching username and a value obtained from the database. The value of a is found from matching the sessionID and database.
10. If the authentication is valid, then the data is received and decrypted, if the authentication process is not valid, then the data will direject and does not need to be decrypted.

The proposed technique is the result of the development of a technique previously reserach conducted by Brandon [7]. One of them is added a variable N in the form of prime numbers used to calculate modulus in a few equations. The proposed of development of zero knowledge proof algorithm, will be generate the more digits that are generated from a variable value, especially in the form of N primes and it will be more difficult to solve in the discrete logarithm problem and requires a very long time. In addition, the client only need to enter the password once at the time of registration and if clients want to send data to the server, they do not need to enter it again. The calculation of proposed method as follow:

a. T_1 = g_0^{rx} mod N
b. T_2 = ((Y^c mod N) (g_0^{zx} mod n)) mod N
c. Then obtained the equation: g_0^{rx} mod N = ((Y^c mod N) (g_0^{zx} mod N)) mod N
d. Y = g_0^x mod N
e. z_x = r_x - cx
f. Do substitution:
g_0^{rx} mod N = ((Y^c mod N) (g_0^{zx} mod n)) mod N
g_0^{rx} mod N = (((g_0^x mod N) ^{c} mod N) (g_0 $^{(rx-cx)}$ mod N)) mod N
g_0^{rx} mod N = (((g_0^{cx} mod N) mod N) ($g_0^{(rx-cx)}$ mod N)) mod N

Table 6
The difference of previous work and proposed method.

Jun and Brandon	Proposed method
$Y = g_0^x$	$Y = g_0^x \bmod N$
$T_1 = g_0^{rx}$	$T_1 = g_0^{rx} \bmod N$
Client sends (c, z_x) to the server	Client select the data to be sent and do encryption with AES
-	and combines data that has been encrypted with the value of c, zx, username, sessionID and sent to the destination (server)

Meet the modulo nature of identity: (A mod N) mod N = a mod N.
$g_0^{rx} \bmod N = ((g_0^{cx} \bmod N)\,(g_0{}^{(rx-cx)} \bmod N)) \bmod N$
Meet the inverse modulo multiplication properties:
A mod N = ((ab mod N) (b^{-1} mod N)) mod N.
$g_0^{rx} \bmod N \equiv (((g_0^{cx} \bmod N) \bmod N)\,(g_0^{(rx\,-\,cx)} \bmod N)) \bmod N$

From these equations can be used to prove that c = hash (Y, T_2, a) and the development of the authentication process is still valid.

The summary of the differences between Brandon and proposed method can be seen in Table 6.

The algorithm proposed in this study was developed using the Python programming language and data in text format was used for data retrieval and measurement of the algorithm's performance.

4. Results and discussion

Comparison of transmission security systems between Zero Knowledge Proof by Brandon with Zero Knowledge Proof Algorithm that has been modified along with the Advanced System Encrpytion of each algorithm for each experiment can be seen in Figs. 5–7.

From this comparison in Fig. 5 and Fig. 6, it appears that the algorithm Zero Knowledge Proof of Fig. 5 generating poor performance because every time we make a calculation, especially when the process has a calculation power, it will take a very long time. However, the calculation process depends on the value of the exponent, if the exponent value is large then the calculation will be done very long time. But in some process of sending data, calculation of exponent with small digits, will take more rapidly so that the algorithm becomes unstable in terms of the process therefore need the limitation, especially for the value of g_0 and zx that are derived from a random value. This limitation resulted in algorithms become easier to crack because of the limited range of values. So that it becomes a weakness of Zero Knowledge Proof that developed by Brandon. The Zero Knowledge Proof algorithms that we have been modified much faster calculation performance and is not affected by the value of the exponent. This is because the process of calculating the exponent developed again with the calculation of the modulus which is the fast exponentiation algorithm [10]. In addition, the methods developed here also utilize one of the advantages of discrete logarithm calculation, in which a^x mod n = b, if the value of n is prime and very big then to find the value of x will be very difficult and requires a very long time to be able to find the value of x [15].

Table 7 shows a comparison of the range of the number of digits exponent value of each variable in the algorithm to the time required for the calculation process.

In Table 7, comparison of the number of digits in the exponent value shows that a very significant number of the time to process the algorithm. In the method of Zero Knowledge Proof of Brandon showed greater exponent digits in the time required to complete the calculation will be longer. While the Zero Knowledge Proof algorithm that has been modified magnitude of the exponent does not affect the calculation.

Fig. 7 shows a comparison of performance between Zero Knowledge Proof with Modified Zero Knowledge Proof (our proposed). The comparison shows that the result of proposed method has better performance from the results of the previous method. The proposed method can process the authentication to be faster than previous methods. In addition, by using the proposed method the variation rate (combination) of the calculation will be higher than previous method, in other words can be said to be more secure, especially for the transmission of a security system that is continuous because there is a one-time token mechanism where each time the token data transmission will change. And from the comparison in Fig. 7 shows the algorithm depend on the data size, the amount of data sent affect the time required for data transmission process.

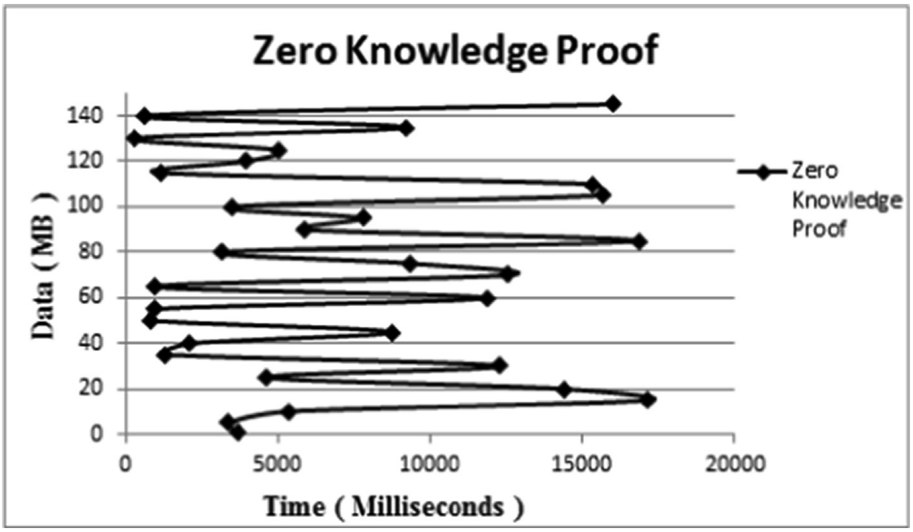

Fig. 5. Performance of Zero Knowledge Proof.

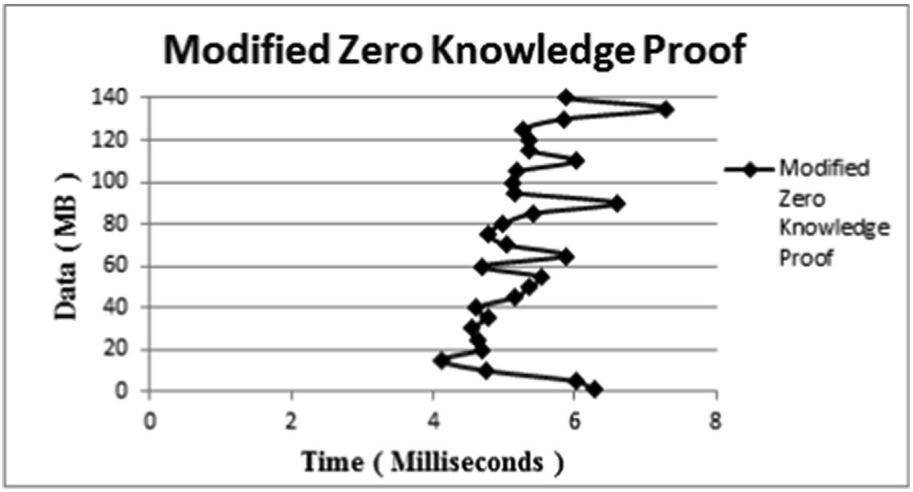

Fig. 6. Performance of Modified Zero Knowledge Proof.

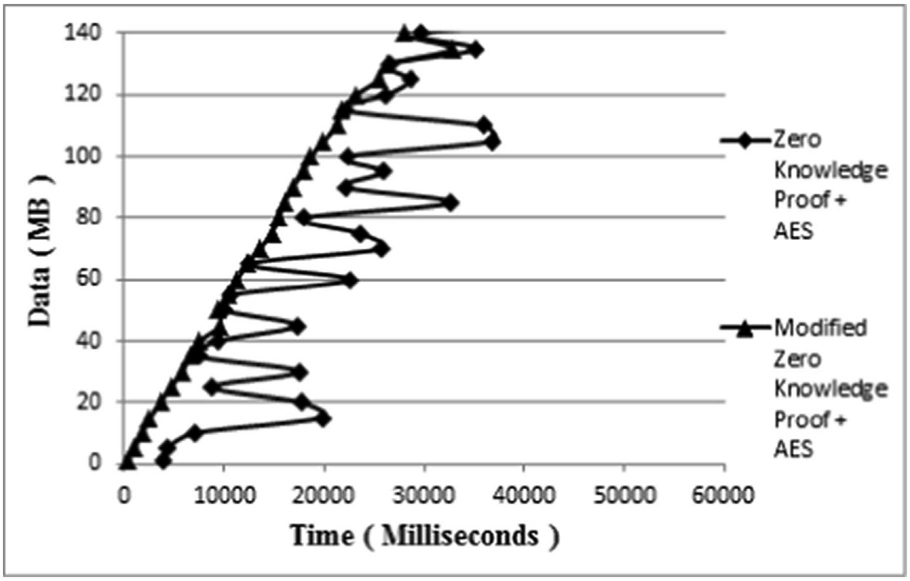

Fig. 7. Comparison of performance.

Table 7
Comparison of time.

Metode	Number of exponent	Range of time
Zero Knowledge Proof	0–5 digit	0–15 s
	>5 digit	> 15 s
Modified Zero Knowledge Proof	0–5 digit	3–8 milli second
	>5 digit	3–8 milli second

5. Conclusion

The focus on this study is to design a security system with encrypted data transmission. We used Zero Knowledge Proof algorithm for authentication and encryption of data using Advanced Encryption System. Zero Knowledge Proof Authentication System utilizes discrete logarithm method as a process of calculating the authentication data. To test our method, experiments performed by making a simulation of the Zero Knowledge Proof methods that have been developed and the method of Zero Knowledge Proof of Brandon as a comparative study. To make the process of data transmission, the process is divided into two processes, namely the process of registration and authentication (and transmission) process. In registration process, user register a username and password, but the password is not sent to the server, but rather a variable Y representing the password sent to the server. In the process of transmission and authentication, the user did not need to enter the password again, and only need to enter the data to be sent then the system will do encryption process and calculation, then delivery made to the server, after the server receives and checks the data, and the data will be decrypted. From the results of experiments conducted, previous Zero Knowledge Proof method has poor performance on the level of security of data transmission systems with the authentication process to be long range in between 1–15 s. The proposed method has a performance level which is much better with long range authentication process is less than 1 s. In addition, the proposed method can have a range of values is much larger calculations so as to break into the security will be much more difficult.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Andreadis G, Fourtounis G, Konstantinos-Dionysios B. Collaborative design in the era of cloud computing. Adv Eng Softw 2015;81:66–72.

[2] Camenisch JL. Group signature schemes and payment systems based on the discrete logarithm problem, ETH series in information security an cryptography. Germany: Hartung-Gorre-Verlag; 1998.

[3] Canavan JE. The fundamentals of network security. Artech House Telecommunications Library, 685 Canton Street, Norwood, MA 02062; 2001..

[4] Daemen J, Rijmen V. The design of Rijndael: AES-the advanced encryption standard. Belgium: Springer; 2001.

[5] Jacques JQ, Guilou LC, Berson TB. How to explain zero-knowledge protocols to your children. Adv Cryptol 1989;435:628–31.

[6] Jeske T. Floating Car Data from Smartphones: What Google and Waze Know About You and How Hackers Can Control Traffic. Technical Report. Institute for Security in Distributed Applications, Hamburg University of Technology, 21079 Hamburg, Germany; 2013..

[7] Jun LJ, Brandon. Implementing zero-knowledge authentication with zero knowledge. Pyhton Papers Monograph, Proc PyCon Asia-Pacific 2010;2:1–19..

[8] Karthigaikumar P, Rasheed S. Simulation of image encryption using aes algorithm. IJCA Spec Issue Comput Sci -New Dimensions Perspectives 2011;4:166–72.

[9] Makhdoom I, Abolhasan M, Abbas H, Ni W. Blockchain's adoption in iot: The challenges, and a way forward. J Network Comput Appl 2019;125:251–79.

[10] Schneier B. Applied cryptography. Minneapolis, MN: John Wiley and Sons Inc.; 1996. p. 55419.

[11] Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput 1997;5:1484–509.

[12] Soewito B, Marcellinus Y, Hapsara M. Secure wireless ad hoc networks using zero knowledge proof. J Comput Sci 2014;10:2488–93.

[13] Soewito B, Wiguna A, Suharjito Diana. Bluetooth low energy: comparing 4 trilateration models in indoor positioning system. Int J Commun Antenna Propag 2018;8:500–9.

[14] Subramanyan B, Chhabria VM, Babu TGS. Image encryption based on aes key expansion. In: Proceedings of the 2011 second international conference on emerging applications of information technology. p. 217–20.

[15] Witno A. Theory of numbers. North Charleston, SC, 29418, USA: BookSurge; 2008.