Jurnal Mantik, 5 (2) (2021) 1019-1029

Published by:Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik/index

Zero Knowledge Authentication Modification for Drone and Server Communication Security

Choirun Nisa[1], Amang Sudarsono[2], Mike Yuliana[3]

[123] Informatic and Computer Department, Electronical Engineering Polytechnic Institute of Surabaya

E-mail: choyunnisa@gmail.com , amang@pens.ac.id , mieke@pens.ac.id

A R T I C L E I N F O

Article history:
Received: 10/06/2021
Revised: 20/06/2021
Accepted: 10/07/2021

Keywords:
drone communication, authentication, zero-knowledge authentication, secret key generation, secure communication

A B S T R A C T

Drones are now being used in various fields. One of the uses of drones is for delivery service. When delivering goods, the drone needs to report its condition while traveling to the server so that the server can monitor the drone and the server can give some commands to the drone. However, when the drone receives data from the server, there are many vulnerabilities it has against the attacker that can take control of the drone device. Taking control of the drone can occur if the attacker manages to send a script that can take over control of the drone. If the attacker can take control over the drone device, the attacker can steal the drone both physically or data of the drone. For this reason, in this study, we propose an authentication algorithm that can filter out who is allowed to send data to the system. The algorithm used is Zero-Knowledge Authentication. To improve the performance of the Zero-Knowledge Authentication algorithm in terms of authentication speed, we try to combine it with the secret key generated from the Secret Key Generation process. From the results of authentication testing, the effectiveness of the proposed algorithm after a Brute Force attack experiment is 100% for single attacker scenarios and for multiple attackers the effectiveness is 100% in LOS and NLOS conditions. Because drones have limited resources, the authentication time experiment is also performed and the result is that in all scenarios the time required to authenticate is in microseconds so that the proposed algorithm can be implemented on drones with limited resources.

1. Introduction

The development of technology in various fields is currently very fast. One technology that is quite popular today is drones. Drone is an unmanned aircraft or ship that is driven by a remote control or internal computer[1]. Drones can easily fly to places that are difficult for humans to reach, so drones are often used to observe areas that are difficult to reach and dangerous, such as searching for disaster victims in dangerous areas, or simply to monitor forest or plantation areas. In addition, the use of drones that are likely to grow rapidly is for delivering goods to customers as has been developed by Amazon, Google and DHL[2]. By using drones as delivery vehicles, goods will arrive at customers faster than using normal courier services because congestion is avoided, goods will be easier to reach areas that are difficult to reach by normal courier services such as mountains, and use drones as delivery vehicles. will also reduce air pollution when compared to normal courier services[1].

However, to implement drones as a delivery person there is a vulnerability that must be considered, namely the takeover of drone control from the attacker. Drones are one type of IoT and the main vulnerability of IoT is in the takeover of control[3]. If control of the drone has been taken over by the attacker , the attacker can direct the drone to fly to where the attacker wants and the attacker can also steal both physical and data from the drone. Of course this is very detrimental. For this reason, in this study, we propose an authentication algorithm that can identify whether the entity that will send data to the drone is a trusted server or not. If it is trusted, the drone will receive data from the sender, but if it is not trusted, the drone will refuse communication. And vice versa if the drone wants to communicate on the server. The algorithm is Zero-Knowledge Authentication [4] . With Zero-Knowledge Authentication, besides being able to recognize trusted entities, it can also prevent other attackers from attacking , namely packet sniffing. This is because in

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Zero-Knowledge Authentication the process of verifying the entity that will send data is trusted or not is to use a challenge-response in the form of random numbers resulting from computational algorithms that will continue to change . Unlike ordinary authentication, which requires a password or a certain number, if the password falls into the hands of the attacker , the attacker will pretend to be a trusted server (impersonating attack). With challenge-response , the authentication system will be more secure.

Debasmita Dey, GP Biswas [4] in his paper entitled Efficient Entity Authentication Using Modified Guillou–Quisquater Zero-Knowledge Protocol tries to modify the Guillou-Quisquater Protocol algorithm [5] which is zero-knowledge authentication using the Diffie-Hellman key exchange scheme. . So that the new scheme has fewer rounds which makes the computation time faster. This is because the Guillou-Quisquater Protocol algorithm requires a long authentication time due to the large number of challenge-response processes carried out.

In this study also modify the Guillou-Quisquater Protocol algorithm . If the Guillou-Quisquater Protocol algorithm [5] challenge is a random number, in this study the challenge is in the form of a secret key generated from RSS (received signal strength). Key generation with RSS utilizes the property of wireless channel reciprocity in which the sender and receiver measure the channel characteristics between them at the same time [6]. In addition, the use of the key as a challenge is to increase the authentication time between the drone and the server. In the existing Zero-Knowledge Authentication algorithm , there are many challenge-response processes that must be passed so that it takes a long time for the authentication process. So this algorithm is not suitable to be applied to drones that have limited resources . However, by using the key results of the process Pembangitan Secret Key authentication algorithms combined with e xisting so it only requires one process challenge- response.

The proposed algorithm will measure its performance using brute force attacks and packet sniffing. In addition, a measurement of the length of the authentication time will also be carried out to see how long the authentication time of the proposed algorithm will be.

The contribution of this research is to increase the speed of authentication time for existing algorithms , create new authentication algorithms modified from existing algorithms and also create authentication algorithms that can be implemented on drones or IoT which have limited resources.

2. Literature Review

During the process of delivering goods, the drone needs to send and receive data from the server. However, in these conditions drones are in public areas that have many vulnerabilities. Choudhary et al. [2] in his research entitled Internet of Drones (IoD): Threats, Vulnerability, and Security Perspectives, states that with the growing use of drones, the more vulnerable in terms of data security because communication passes through public radio networks. It was also mentioned that there are several types of attacks that threaten the internet of drones, including traffic analysis, man-in-the-middle, eavesdropping, GPS spoofing, and firmware.

According to M. Farooq et al[3], the main challenge in IoT security is preventing attackers from gaining control over IoT systems because IoT is usually spread in public areas. This also applies to drones because drones also use IoT devices. Gaining control over the IoT system on the drone can occur when the IoT system on the drone is unable to filter who is allowed to send data to the system. These filters are usually called authentication mechanisms. Authentication is a mechanism to filter who is allowed to enter the system or send data to the system by identifying who is trying to send data. With authentication, not everyone can send data or only people who have permission can send data to the IoT system.

Debasmita Dey and G. P. Biswas [4] in their paper entitled Efficient Entity Authentication Using Modified Guillou–Quisquater Zero-Knowledge Protocol try to modify the Guillou-Quisquater Protocol algorithm which is zero-knowledge authentication by using the Diffie-Hellman key exchange scheme. So that the new scheme has fewer rounds which makes the computation time faster.

Z. Li, Q. Pei, I. Markwood, Y. Liu and H. Zhu in their research entitled Secret Key Establishment via RSS Trajectory Matching Between Wearable Devices[7] tried to do Secret Key Generation with 4 processes, namely channel probing, quantization which using the mean-value quantization technique, information reconciliation using the bloom filter technique [8], and privacy amplification using the Karhunen-Loeve Transform (KLT) technique. With this process, a common key is generated between the two entities that will communicate so that the communication process becomes more secure.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

In this study, we will combine a zero-knowledge authentication algorithm with keys generated from the Secret Key Generation process to obtain a more effective authentication algorithm in terms of authentication time so that it can be applied to devices with limited resources such as IoT on drones.
Authentication

3. Reaserch Methods

In this study there are 2 main parts, namely Generating Secret Keys and Authentication. Each part is described below.

3.1 Secret Key Generation

To increase the authentication time using the Zero-Knowledge Authentication algorithm[5], a secret key is needed which will generate a series of random numbers for the challenge-response process in authentication. The secret key is generated from the Secret Key Generation process. Fig 1 is a step in the Secret Key Generation process

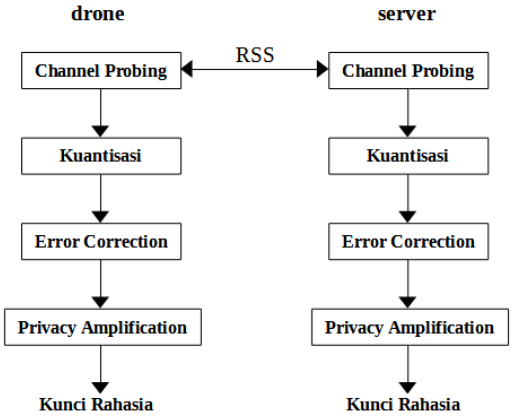

Fig 1. Secret Key Generation Process [7]

In Fig 1 there are 4 processes in Secret Key Generation, namely channel probing, quantization, error correction, and privacy amplification. Channel probing is the process when the drone and server perform a ping command to get RSS (Receive Signal Strength) using the TP-LINK WN7222N device. The channel probing scenario was performed as in [7] with a ping interval of 110 ms.

From the channel probing process, a lot of RSS data is generated. In this study, only 4000 RSS data were used. The 4000 RSS data will go into the quantization process to be converted into a series of bits. The quantization algorithm used is middle-point quantization[7]. The Middle-point quantization algorithm improves the mean-value quantization scheme [8] by increasing the number of generated bits so as to increase the resulting bit value. The RSS data obtained will be divided into several intervals. Then the average value and the midpoint value will be calculated. The midpoint is calculated by adding up each bit in the interval with the last bit of the interval and dividing by 2 consecutively. The midpoint value is the threshold which will determine which sample value will be converted to 0 or 1. If the average value of the interval is greater than the threshold, it will be converted to 1 and if it is less than the threshold, it will be converted to 0.

After getting a series of bits, the next step is the error correction process. This process serves to eliminate errors that exist in the drone and server bit sequences because the bit sequences between the two must be in the same order. The algorithm used in this process is the clove filter algorithm [7]. After this process the sequence of bit sequences on the drone and server becomes the same. However, even if the two sets of bits are the same, a process must be carried out to increase the security and randomness of the two sets of bits. This is necessary because during the quantization process to error correction there is no security process that has the opportunity to leak key data to the attacker. In addition, the more random the generated bits, the more difficult it is for the attacker to imitate the generated key. The output results of each process are shown in Table 1

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Table 1
Output Results of Each Key Generating Process

Proses	Drone	Server
Channel Probing	-41, -44, -45 ..	-42, -42, -43 ...
Kuantisasi	01001110001..	10001100011..
Error Correction	10001110001..	10001110001.
Privacy Amplification	10010101001..	10010101001..

From Table 1 it can be seen that the output of the Secret Key Generation process is a random series of 0 and 1 binary bits

3.2 Authentication

After successfully obtaining the secret key, then the key will be used in the authentication process. In this study, the authentication algorithm used is the Guillou Quisquater which is a zero-knowledge authentication which will be combined with a secret key. Details of the proposed authentication algorithm can be seen in Fig 2 and Fig 3. In Fig 2, when the drone will send data to the server, it must go through the authentication process first. And vice versa, when the server wants to process

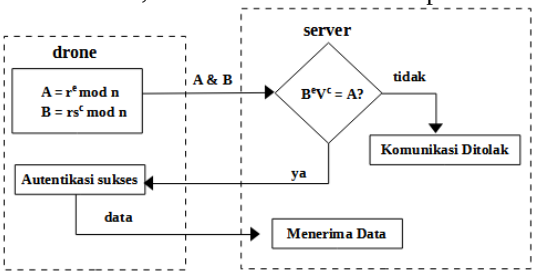

Fig 2. Authentication algorithm on the server side

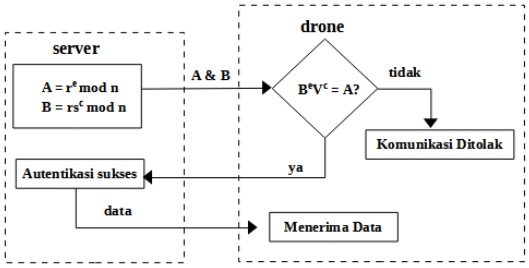

Fig 3. Authentication algorithm on the drone side

communication to the drone then the server must go through the authentication process first. Both sides of the authentication process have the same steps. From here we call the entity that wants to communicate to other entities and must go through authentication with the claimant. While the entity that acts to verify who wants to communicate is called a verifier. Both drones and servers can act as claimants or verifiers. The claimant must calculate the values of A and B as evidence that the claimant is a trustworthy entity where the values of A and B are the following equation.

$$A = r^{e} mod\ n \quad (1)$$

$$B = rp^{c} \text{mod n} \quad (2)$$

with,
r = random number
e = public key
n = modulo

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

p = secret key
c = secret key resulting from key generation

Variables e, n, and p are variables generated by the server using the RSA algorithm [9]. Then these variables will be distributed to the drone using the diffie-hellman key exchange algorithm [10]. The variable distribution stage is carried out before the authentication process and is only carried out once or according to a certain period.

When the claimant sends variables A and B, the verifier side will check whether the result of the calculation with the variable owned is equal to A. The calculation is shown in the following equation.

$$B^{e} V^{c} \bmod n = A \ ? \quad (3)$$

with,
B = variable B sent by claimant
e = public key
c = secret key resulting from the key generation process

while for variable V is a variable that has a value according to the following equation [5].

$$p^{e} V^{} = 1 \bmod n \quad (4)$$

If Equation (3) produces the same value as A, the verifier will send a message to the claimant that the authentication was successful and the claimant can send data to the verifier. However, if it turns out that Equation (3) does not produce the same value as A, the communication from the claimant will be rejected and the claimant cannot send data to the verifier. With this mechanism, the verifier will be able to filter who is allowed to send data to it and only claimants who have the same secret key (c) as the verifier can pass the authentication process and can send data to the verifier.

From Figure 2 and Figure 3, the authentication process only requires 2 processes. That is, the claimant process sends variables A and B as evidence that the claimant is a trustworthy entity and the verification process is carried out by the verifier. With only these 2 processes, the proposed authentication algorithm in theory has improved the performance of the existing algorithm in terms of authentication time. In the existing[5] algorithm, the challenge-response process is carried out e times. Even though e is a public key that has a value of 65537. You can imagine how long the authentication time for the algorithm existing.

4. Research Results and Discussion

This section describes the testing process of the proposed algorithm to determine its performance. There are several test scenarios carried out on Line of Sight (LOS) and Non Line of Sight (NLOS) conditions. The test consists of testing the authentication time if there is a change in distance, testing the authentication time if there is a change in the bit length of the random number used, testing the security of the authentication algorithm using brute force attacks , and comparing the authentication time with the existing algorithm . The details of the test and its results and analysis are described as follows.

4.1 Testing Authentication Time with Increasing Distance

In this test, the drone as the claimant will try to communicate with the server. For communication, the drone uses a Raspberry pi 3B with the WiFi module on the Raspberry pi. While the server is a laptop with Linux OS. The test is carried out in 2 conditions, namely LOS and NLOS and in each condition there will be 2 scenarios, namely static and dynamic scenarios. Fig 4 and Fig 5 are the test locations and the placement of the device during testing.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Fig 4. Testing the authentication time by varying the communication distance under conditions LOS

Fig 5. Testing the authentication time by varying the communication distance under NLOS conditions

Figure 4 shows the location of the tests carried out and the placement of the device when the tests were carried out under LOS conditions. The drone will fly with a vertical distance of 10 meters and the horizontal distance to the server is varied, namely 5, 10, 20, 30, and 40 meters. There is no barrier between the drone and the server, so it is called the LOS condition. In LOS conditions there are 2 test scenarios carried out, namely static and dynamic. Static scenarios are scenarios where the drone remains in position and does not move. While in the dynamic scenario the drone moves to the right and left as far as 2 meters. The results of the test under LOS conditions are shown in Table 2.

While Figure 5 shows the test on NLOS conditions, namely between the server and the drone being blocked by trees and buildings. Scenario and changing the distance are the same as in the LOS condition. The test results under NLOS conditions are shown in Table 3.

TABLE 2
TEST RESULTS AUTHENTICATION TIME VS DISTANCE IN LOS CONDITIONS

Distance (meters)	Static Scenario Authentication Time (seconds)	Dynamic Scenario Authentication Time (seconds)
5	0.000285	0.000313
10	0.000271	0.000305
20	0.000363	0.000387
30	0.000415	0.000421
40	0.000417	0.000419

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

TABLE 3
TEST RESULTS AUTHENTICATION TIME VS DISTANCE IN NLOS . CONDITIONS

Distance (meters)	Static Scenario Authentication Time (seconds)	Dynamic Scenario Authentication Time (seconds)
5	0.000330	0.000406
10	0.000238	0.000313
20	0.000455	0.000441
30	0.000446	0.000408
40	0.000455	0.000471

Table 2 shows the results of testing the authentication time for changes in distance in LOS conditions with static and dynamic scenarios. In static conditions, the farther the distance between the verifier and the claimant , the more authentication time required. However, the difference in the value of authentication time to the distance from one another is not so significant. The highest value is when the verifier and claimant are placed 40 meters away, with the required authentication time is 0.000417. And if the distance is increased by more than 40 meters, it is certain that the authentication time will also increase. Meanwhile in the dynamic scenario, the trend is not as stable as in the static scenario. This is due to the influence of the wind that was quite strong at that time. However, if the static and dynamic scenarios are compared, then the dynamic scenarios have lower authentication times

longer than the static scenario. This is because in the dynamic scenario there is movement to the left and right, which can increase the horizontal distance between the claimant and the verifier.

Table 3 shows the test results under NLOS conditions. In the NLOS condition, the trend is the same as the LOS condition, namely in the static scenario, the farther the distance between the claimant and the verifier, the more the authentication time will be. In the dynamic scenario, the trend is not stable. This is due to the instability of the distance between the claimant and the verifier . Meanwhile, the dynamic scenario has a longer average authentication time compared to the static scenario under NLOS conditions.

If the static scenario between LOS and NLOS conditions is compared, the authentication time in the NLOS condition has a longer average than the authentication time in the LOS condition. Because in NLOS conditions there is a barrier so that communication is not directly forwarded to the verifier. If seen from Table 2 and Table 3, the required authentication time is quite fast with the authentication time value shown in 4 digits behind the comma. This shows that the proposed algorithm is quite suitable to be applied to devices with limited resources .

4.2 Authentication Time Test with Random Number Length Change

The next test is to measure the authentication time if the length of the random number (r) is changed. The lengths used are 8, 64, 128, 256, and 512 bits. The scenario in this test is almost the same as the previous test, namely in LOS and NLOS conditions. But the horizontal distance between the drone and the server is fixed at 10 meters. Fig 6 and Fig 7 show the scenario and the placement of the device in this test. In Fig 6, the drone is placed in front of the server at a distance of 10 meters horizontally and flies at a height of 10 meters. This horizontal distance will not change. In this condition, the length of the random number used will be changed. There is no barrier between the drone and the server, so this condition is called LOS. In addition to

Fig 6. Authentication time testing by varying the length of random numbers under LOS conditions

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

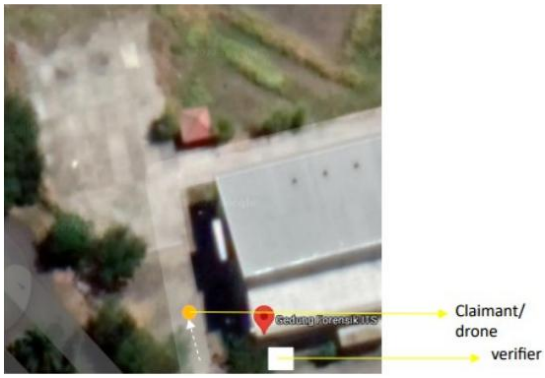

Fig 7. Testing authentication time by varying the length of random numbers under NLOS conditions

In this condition, there are also two static and dynamic scenarios. The results of the authentication time test by varying the length of random numbers under LOS conditions are shown in Table 4. While in Fig 7 the server is next to the building and there is a tree in front of it so this test is called the NLOs condition. the scenario carried out is the same as in the LOS condition. The results of testing the authentication time by varying the length of random numbers under NLOS conditions are shown in Table 5.

TABLE 4
TEST RESULTS AUTHENTICATION TIME VS LENGTH OF RANDOM NUMBERS IN LOS L CONDITIONS

Length (bits)	Static Scenario Authentication Time (seconds)	Dynamic Scenario Authentication Time (seconds)
8	0.000352	0.000363
64	0.000396	0.000362
128	0.000306	0.000551
256	0.000445	0.000571
512	0.000556	0.000629

TABLE 5
TEST RESULTS AUTHENTICATION TIME VS LENGTH OF RANDOM NUMBERS IN NLOS . CONDITIONS

Length (bits)	Static Scenario Authentication Time (seconds)	Dynamic Scenario Authentication Time (seconds)
5	0.000367	0.000385
10	0.000377	0.000378
20	0.000405	0.000439
30	0.000480	0.000565
40	0.000566	0.000643

Table 4 and Table 5 show the results of testing the authentication time on changes in the length of the random variables used. In LOS conditions, static and dynamic scenarios have the same trend, namely the longer the random variable used, the more authentication time required. Between static and dynamic scenarios there is a fairly small difference with authentication time in dynamic scenarios which has a longer average than in static scenarios. The time required for authentication is still on the micro-second scale, which means the length of the random variable can still be increased to improve system security.

Meanwhile for NLOS conditions as shown in Table 5, the trend between static and dynamic scenarios is the same, namely the longer the variable bit used, the more authentication time will be. The test results show that the length of the random variable has an effect on the authentication time even though the difference is quite small between scenarios. Meanwhile, the LOS and NLOS conditions did not have a significant difference so that the LOS and NLOS conditions did not affect the length of authentication time.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

4.3 Authentication Security Testing with Brute Force Attacks

After testing the authentication time, the next test is testing the authentication security using brute force attacks. Brute force is an active attack where the attacker tries to bypass authentication by trying to enter certain passwords on the system. If this brute force attack is successful then the attacker can certainly send data to the system which often endangers the system. A brute force attack performed using a *hydra*. The scenario and the placement of the device are the same as in the random number test. However, this test uses a *single attacker* and *multiple attacker* test scenarios . The *single attacker* scenario is a scenario where the number of attackers is only one. Meanwhile, for *multiple attackers, the* number of attackers is two. Table 6 and Table 7 show the results of this test.

From Table 6 it can be seen that in all scenarios, *single attacker* and *multiple attacker* testing *brute force attacks* on LOS conditions by *attackers* failed in all tests. This means that the proposed algorithm is quite safe. Meanwhile, Table 7 shows the results of testing *brute force attacks* under NLOS conditions. The test results under NLOS conditions are the same as in LOS conditions, namely in all scenarios of *brute force attacks* from the *attacker* failing all tests. In addition, from the results of Table 6 and Table 7, it can also be seen that the test conditions have no effect on the results of the security test.

TABLE 6
BRUTE FORCE ATTACK TEST RESULTS ON LOS CONDITIONS

Test to-	*Single Attacker*	Multiple Attackers
1	fail	fail
2	fail	fail
3	fail	fail
4	fail	fail
5	fail	fail
6	fail	fail
7	fail	fail

TABLE 7
BRUTE FORCE ATTACK TEST RESULTS ON NLOS . CONDITIONS

Test to-	*Single Attacker*	Multiple Attackers
1	fail	fail
2	fail	fail
3	fail	fail
4	fail	fail
5	fail	fail
6	fail	fail
7	fail	fail

4.4 Comparison of the Authentication Time of the proposed Algorithm with the *existing* Algorithm

The last test is to compare the authentication time of the proposed algorithm with the *existing* algorithm [5].

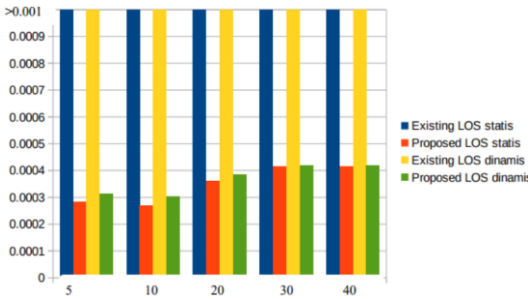

Fig 8. Comparison graph of the authentication of the proposed algorithm with the existing algorithm [5] in LOS conditions

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

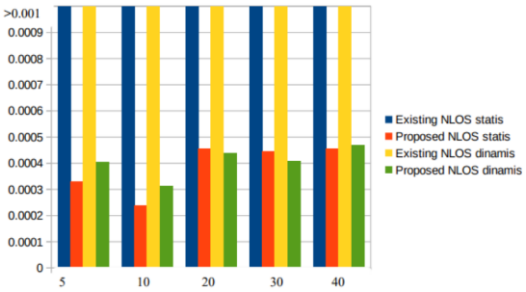

Fig 9. Comparison graph of the authentication of the proposed algorithm with the existing algorithm [5] in NLOS conditions

The scenario that used is to vary the horizontal distance between the drone and the server. Fig 8 and Fig 9 show the results of this test.

Figure 8 shows the results of the comparison between the existing and the proposed scheme in LOS conditions. From Figure 8, it can be seen that in LOS conditions the *existing* scheme takes more than 0.001 seconds in both static and dynamic scenarios. The proposed algorithm is able to increase the speed of authentication time by more than 100%. This is because the proposed algorithm only performs one challenge submission process while the existing algorithm must send several challenges first. Likewise in the NLOS condition in Figure 9. In the Fig it can also be seen that the *existing* algorithm takes more than 0.001 seconds to authenticate. So it can be concluded that the proposed algorithm has increased the speed of authentication in all scenarios.

5. Conclusion

After several tests were carried out to test the quality of the proposed algorithm, it was found that the proposed algorithm was able to reduce the authentication time of the existing algorithm up to 100%. In addition, from testing the authentication time by changing the distance between the drone and the server in LOS and NLOS conditions, it was found that the farther the distance between the drone and the server, the authentication time will increase. However, the increase in authentication time per 10 meters is not significant. Meanwhile, in testing the authentication time by varying the length of the random number, it was found that the longer the bits of the random number used, the authentication time will also increase. However, the increase in authentication time in this test is also not significant, ranging from 0.000050 seconds. From the several results of the authentication time test, it was found that in all the tests the authentication time was in units of micro seconds. Which means that the authentication time of the proposed algorithm is still relatively fast, so it is still suitable if used on devices with limited resources such as drones. In addition, the test results also show that the proposed algorithm is quite safe from brute force attacks with the effectiveness in single attacker and multiple attacker scenarios is 100%.

6. References

[1] V. Gatteschi et al., "New Frontiers of Delivery Services Using Drones: A Prototype System Exploiting a Quadcopter for Autonomous Drug Shipments," 2015 IEEE 39th Annual Computer Software and Applications Conference, Taichung, 2015, pp. 920-927.

[2] Choudhary, Gaurav & Sharma, Vishal & Gupta, Takshi & Kim, Jiyoon & You, Ilsun. “Internet of Drones (IoD): Threats, Vulnerability, and Security Perspectives”, 2015.

[3] M. Farooq., M. Waseem., A. Khairi., & S. Mazhar.. “A Critical Analysis on the Security Concerns of Internet of Things (IoT)”. International Journal of Computer Applications, vol. 111, no. 7, pp. 1–6, 2015.

[4] Debasmita. D., & G. Biswas. “Efficient Entity Authentication Using Modified Guillou–Quisquater Zero-Knowledge Protocol”. 2020.

[5] S. Paramanik. “Comparison of Zero Knowledge”, NATIONAL INSTITUTE OF ROURKELA TECHNOLOGY, 2014.

[6] A. Sudarsono, M. Yuliana and P. Kristalina, "A Reciprocity Approach for Shared Secret Key Generation Extracted from Received Signal Strength in The Wireless Networks," 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA), Bali, 2018, pp. 170-175.

[7] C. Nisa., A. Sudarsono., & Y. Mike. "RSS-based Secret Key Establishment using MiddlePoint Quantization and Clover Filter Algorithm". International Electronics Symposium (IES), IEEE, 2020.
[8] Z. Li, Q. Pei, I. Markwood, Y. Liu and H. Zhu, "Secret Key Establishment via RSS Trajectory Matching Between Wearable Devices," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, pp. 802-817, March 2018.
[9] X. Zhou & X. Tang. "Research and Implementation of RSA Algorithms for Encryption and Decryption". The 6th International Forum on Strategic Technology, 2011, IEEE.
[10] AS Rawat and M. Deshmukh, "Efficient Extended Diffie-Hellman Key Exchange Protocol," 2019 International Conference on Computing, Power and Communication Technologies (GUCON), 2019, pp. 447-451.