Scientific Journal of Informatics
Vol. 6, No. 1, May 2019
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Speed and Power Consumption Comparison between DES and AES Algorithm in Arduino

Arcelina Sukiatmodjo[1], YB. Dwi Setianto[2]

[1,2]Informatics Engineering Department, Faculty of Computer Science,
UNIKA Soegijapranata Semarang, Indonesia
Email: [1]15k10044@student.unika.ac.id, [2]setianto@unika.ac.id

Abstract

Telemedicine is commonly used to check or diagnose patients from a long distance. Its application is often combined with sensors as needed, but for delivery, a cryptography algorithm is needed so the data sent safely, illegible, and can not be changed by unauthorized people. Besides that, the algorithm must be light, fast and use less power. In this study, a comparison of the Data Encryption Standard (DES) and Advanced Encryption Standard (AES) algorithms will be implemented in the encryption module. Data from the sensor encrypted and sent to the server. The time and power consumption by DES will be compared with AES. From this research, we can conclude that the encryption time of AES is faster than DES. The average difference speed is 33413 microseconds. Then the power consumption by AES and DES does not have any significant difference, and the addition of sensors causes additional power as well.

Keywords: Telemedicine, DES, AES, Arduino, ACS712

1. INTRODUCTION

Nowadays, telemedicine is commonly used to diagnose patients from a long distance. Telemedicine is the use of information and communication technology combined with medical expertise to provide health services, ranging from consultation, diagnosis and medical treatment, without limited space or carried out remotely [3]. In the implementation, one of the challenges faced in telemedicine today is security. To keep the data sent by embedded systems unchanged, can't be read, opened, or accessed by unauthorized people, a lightweight encryption algorithm is needed so the Arduino easily to executing programs. Besides being lightweight, a fast algorithm is needed and uses less power because examination equipment allows using batteries. The popular or widely used symmetrical cryptographic algorithm is the Data Encryption Standard (DES) algorithm and the Advance Encryption Standard (AES).

The use of DES and AES to protect data has been widely applied. And basically, an embedded-based telemedicine system has also been done too [1]. In order to protect messages when it sent, the message must be secure from the irresponsible parties. In another case, Security Improvisation in Image Steganography using

DES, the research talk about protect the confidentiality and integrity of data against unauthorized access, cryptographic techniques are needed to hide secret messages that are masked secret information within other information or named steganography [7]. Another implementation of DES is SSTL Based Power Efficient on 28nm FPGA [6]. AES implementation has been done by made a smart card with three important card application, there are health card, aadhaar card, and passport cards to make it more secure [5]. AES also implemented on Vehicle Electronic Key System Based on Android Operating System and Arduino Microcontroller [8] and Enhance Data Security in Cloud Computing [9]. ACS712 are used to detect the amount of current flowing through the terminal block and then Arduino processed the data of power usage [4]. They use to monitoring and controlling the lights using LAN and network router and result of power consumption shows in smartphone. Another case, the use of ACS712 is on design and analysis of a low-cost PV analyzer using Arduino UNO [2].

A few researchers focused on the implementation of DES and AES. There have been limited studies concerned on the data security several non-telemedicine devices. Therefore, this research intends to implement the used of DES and AES on Arduino board and connect with ACS712 to calculate the power consumption. From the comparison of the two algorithms tested, the algorithm that has the fastest or most efficient encryption process and uses less power is used to create a telemedicine system.

The objectives of this research are to implement and compare cryptographic algorithms that are fast, lightweight and use minimum power between DES and AES symmetrical cryptographic algorithms. Which will later be used to create a system for sending heart rate data, patient body temperature, or others in telemedicine.

2. METHODS

2.1. DES and AES Implementations on Arduino

The first thing to do in this research is to implement DES and AES algorithms on Arduino with C language. A variable that stores text (in the form of a String) originating from a sensor, is encrypted using a specified key. The encryption results are ciphertext sent to the PHP server. Ciphertext is decrypted with the same key using DES and AES on php. The results are displayed and matched whether they are in accordance with the one sent.

2.2. DES and AES Implementations on PHP

The decryption of data/ciphertext from arduino is done using php. The ciphertext in the form of string is decrypt then matched with the plaintext when it has not been encrypted.

2.3. Installation of Sensors

As a simulation, the sensors used in this study are LDR and Potentiometer. In the same program with DES / AES code, the sensor results are read and encrypted.

2.4. Compare the DES and AES Encryption Speeds

The speed of data encryption in DES and AES is done on the same program using the micros(ms) time unit. The research takes 2 kinds of encryption time, there are encryption time per piece and total encryption time.

2.5. Compare the Power Consumption to Process DES and AES

To find out the power consumption in Arduino when executing DES / AES is by using the ACS712 sensor. The ACS712 sensor connected to the adapter and the Arduino that execute DES / AES, then calculate the power during the program.

2.6. Testing

After all sensors and components are installed, testing of the tool is carried out. Starting from data collection from sensors, data encryption process, sending to the server, until finally decrypted and displayed the results of the sensor. The time data used for DES / AES encryption and the power consumption are recorded for comparison, then the results of the research are concluded.

3. RESULT AND DISCUSSION

This research uses C language on Arduino for the encryption process and php for data decryption. The components needed in this research include 2 Arduino UNO, ethernet shield, LAN cable, current sensor (ACS712 5V), adapter, capacitor 100nf, diode 1n4148, jumper cable, light sensor (LDR), and potentiometer. There are 2 arduinos which is the 1st arduino is to execute DES/AES algorithm that connected with sensor and ethernet shield to send data, and the 2nd arduino is to calculate the power consumption by 1st arduino to execute the program.

3.1. Encryption Process on Arduino

There are two cryptography algorithms in Arduino for the data encryption process, there are DES and AES algorithms. Data from the sensor is in the form of a string that will be encrypted and tested alternately using the DES and AES algorithms.

For DES, this project uses a library from the source https://github.com/Octoate/ArduinoDES. Because DES is only able to encrypt data by 8 bytes optimally, modifications are made so that even long data can be encrypted at once. The used method is breaking the existing text / data into per 8 bytes. So, the encryption is done alternately per piece, then put back together into 1 string. But in the implementation using a sensor, the data from the sensor is set so the result becomes fix 8 bytes. The key in DES requires 8 bytes of data.

In AES, libraries are used from source https:// www. arduinolab.net/aes-encryptiondecryption-using-arduino-uno/. The AES plaintext can be optimally

encrypted at a size of 16 bytes. Modification are made too so the encryption can be done with data multiples of 16 bytes. Because the size of the plaintext can only multiply by 8 bytes (in DES) and 16 bytes (in AES), then the text that is not even multiples of 8 or 16 bytes will be lost or not encrypted. For the AES key, 16 bytes of data are needed.

In data encryption, the first thing to do is read data from the LDR sensor and the Potentiometer. The results of the two sensors are combined into a string with a length of 16 bytes. Enter 8-digit key (8bytes) for DES, and 16-digit key (16 bytes) for AES. When the encryption process takes place there are two calculation of processing time, there are time per 8 bytes of data (in DES) and 16 bytes of data (in AES) and the entire time of the encryption process. The time used for the encryption process can be seen in the serial monitor as a comparison material. The program also produces ciphertext which is sent directly to the server via HTTP POST.

In Encryption process there are calculation of encryption time too in the same program. After encryption, the data/ciphertext that has been obtained is sent to the server. And encryption time from both algorithms are compared.

3.2. Decryption Process using PHP

The ciphertext decryption process is done using php, the ciphertext data comes from Arduino sent via HTTP POST using ethernet shield. In DES decryption, this project uses a library from the source https://github.com/gilfether/phpcrypt. No need to separate the encryption data, all string data can be directly decrypted by php files. Enter the same key as used in encryption, because if the key is different then the decrypt result will not same with the plaintext. In AES decryption, this project uses a library from the source https://github.com/phillipsdata/phpaes. Like DES, decryption of AES also doesn't need to separate per 16 bytes. The key is also entered with the same key as used in encryption.

Data from Arduino sent via HTTP POST will be processed by php. The result of the decryption is the plaintext from sensor data result sent by 1st Arduino.The encryption process is done on Arduino because most remote inspection devices, mainly for retrieving patient data are using sensors connected to Arduino. The encryption process is carried out on the same program when reading the results of sensor data. So, the data is directly processed and not read or changed by unauthorized parties. The process also requires a lightweight program because there is a possibility that the tools used for data retrieval require little power, such as from battery.

The decryption process is done using php because the data that has been sent to Arduino via POST HTTP will be received by the user even from different city. In order to read directly, the decryption process is done using php without having to reconnect to Arduino at the destination. The results of the decryption process will be seen on the serial monitor when running DES / AES encryption on Arduino.

3.3. Sensor

There are many kinds of sensors for telemedicine such as pulse sensors, body temperature, heart rate sensors, and etc. In this research, as a simulation, the sensor was replaced with LDR and potentiometer. LDR and potentiometer are connected with an ethernet shield that has been connected to 1st Arduino. The data from both sensors entered into the same coding as DES / AES. So, the data that has been obtained can be directly encrypted and sent to the server.

3.4. Calculate the Encryption Speed

The encryption speed measurement is done in the DES / AES program using micros() function. There are two calculation of time used for the encryption process. The first time is encryption per piece. Second, the encryption time of all strings / data. The result of micros() is time in microseconds, note that 1 micros = 0.000001 seconds. The result can be seen in the serial for the encryption process per piece and total time for comparison.

3.5. Power Consumption to Process DES and AES

The power consumption by the 1st Arduino is measured by ACS712 sensor that connected with 2nd Arduino. This research used the ACS712 sensor with the addition of diode 1n4148 and 100nf capacitors. The use of diodes is to control the direction of the current so that the current through the diode is only one direction. While the use of capacitors is to filtering the ripples from the rectification so we obtained the smooth and flat waves. Both of these components are intended so that the results of the ACS712 sensor can be more stable.

3.6. The Circuit

Here the complete circuit scheme of the research:

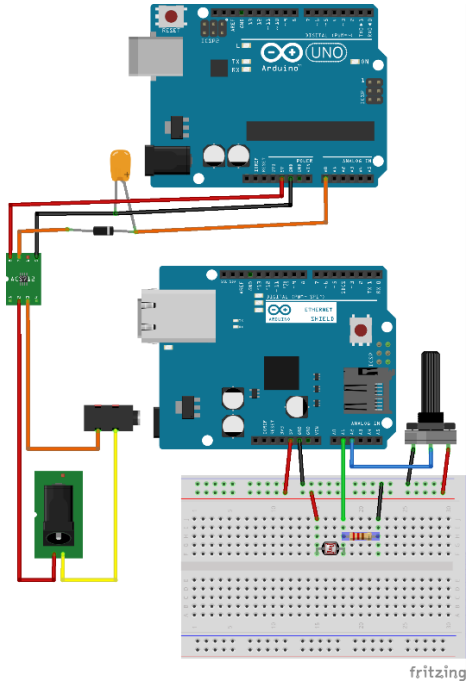

Figure 1. Complete circuit scheme

After the data from the LDR and Potentiometer sensors are read, the data will be processed in the 1st Arduino. Data from both sensors will be merge into 16-digits character. Then, encrypt it using DES/AES in the same program with reading sensors data. The ciphertext sent to MySQL server and saved. The ciphertext also sent to php to decrypt. The result of encrypt and decrypt can be seen on serial monitor. The power obtained by the 1st Arduino comes from the jack that is connected to the adapter. Then the 1st Arduino is connected to the ACS712 sensor to do the power reading used on 1st Arduino. Furthermore, the ACS712 sensor is connected to the 2nd Arduino by passing diodes and capacitors so the results of the obtained power measurements are more stable and accurate. The power measurement results appear in the serial monitor for a comparison between the power output used by DES and AES.

3.7. Testing

3.7.1. Encryption Time without Loop

Table 1. Encryption time without loop (microseconds)

*	DES (8 digit)		DES (16 digit-2loop)			AES (16 digit)	
	Encrypt	Total	1st piece	2nd piece	Total	Encrypt	Total
Without Sensor	17836	19648	17836	17844	39040	1528	5856
1 Sensor	17840	19596	17844	17848	39168	1536	5696
2 Sensors	17848	19604	17844	17836	39288	1536	5704

After testing, data obtained during the encryption process as shown in the table. The total encryption time of DES 16 digits is 2 times longer than 8 digits. This is because in DES 16 digits the data will be divided into 2 pieces (8 digits each) and then encrypted. For encryption time of DES 16 digits with AES 16 digits, AES is faster than DES. Can be seen in Table 1 in encryption per pieces with sensors or without sensors, and total encryption time that AES has a significant difference. The difference in average total encryption time between AES and DES 16 digit is 33413 microseconds, where AES is faster than DES in same data length. Installation of sensors in DES encryption is 8 digits, DES 16 digits and AES 16 digits do not have a significant effect on the encryption process. This is because without the sensor the length of data to be encrypted is set according to the digits listed. The difference average in the amount of encryption time per piece with the total encryption time is 3481 microseconds. This is due to the data splitting process being per 8 digits (DES) or 16 digits (AES), then encrypted per piece and then combined again.

3.7.2. Encryption Time with Loop

Table 2. Encryption time with loop (microseconds)

*	DES (8 digit)		DES (16 digit-2loop)			AES (16 digit)	
	Encrypt	Total	1st piece	2nd piece	Total	Encrypt	Total
Without Sensor	17839,6	19361,2	17841,2	17842,4	38574,8	1531,2	5060
1 Sensor	17840	19420,8	17841,2	17842,4	38928,8	1531,2	4909,6
2 Sensors	17840,8	19429,2	17839,6	17843,2	38942	1532	4964,4

The table above is almost the same as the previous encryption table except that this table comes from an encryption program that is repeated. So, the reading of the data, the process of encryption and sending is done repeatedly. The aim is to find out the power consumption by Arduino for the DES or AES encryption process. The data above obtained from the results of an average of 10 times result of the looping process. It can be seen in Table 2 that the DES 8 digits, DES 16 digits and AES the encryption processes do not have significant different result from encryption without looping in Table 2. Just like without looping, the difference in average total encryption time between AES and DES 16 digit in this loop is 33837 microseconds, which is not so different with previous data. With or

without sensors also does not have a significant effect on the speed of encryption on this iterative process. The difference in the amount of encryption time per piece with the total encryption time is same with the encryption time without loop.

3.7.3. Power Consumption

After an experiment using ACS712, the data obtained as follows:

Table 3. Power consumption (milliampere)

*	DES (8 digit)	DES (16 digit)	AES (16 digit)
Without Sensor	552,164	596,514	568,72
1 Sensor	582,612	599,72	507,253
2 Sensors	601,32	632,922	573,302

The data above is in mA (milliampere) where obtained from an average of 10 times the process of calculating the current through ACS712. The power consumption by DES 16 digits is only slightly different from DES 8 digits. Likewise, with the difference in power consumption by DES 16 digits with AES 16 digits, does not have any significant difference. The use of 1 sensor and 2 sensors affect the power consumption on Arduino. The more sensors used, the more power needed as well.

4. CONCLUSION

The conclusion of this study is that the DES and AES algorithms can be implemented on Arduino. For the encryption process, AES is faster than DES. AES has a difference in average speed of 33413 microseconds, where AES is 6x faster than DES in same data length. The use of sensors does not significantly affect the speed of data encryption. For power consumption, DES and AES do not have significant differences. The addition of sensors affects the power consumption, the more sensors used, the more power needed as well. The final result from this research is that AES is more suitable for telemedicine system than DES, both in terms of encryption speed and power consumption.

5. ACKNOWLEDGMENT

This research is part of the Research "Telemedicine System with a Holistic Concept to Provide Standard Diagnosis Materials". This research was funded by the Kemenristekdikti DRPM Fiscal Year 2019.

6. REFERENCES

[1] Adhie, R. P., Hutama, Y., Ahmar, A. S., & Setiawan, M. I. (2018). Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC). *In Journal of Physics: Conference Series (Vol. 954, No. 1, p. 012009). IOP Publishing*.

[2] Anand, R., Pachauri, R. K., Gupta, A., & Chauhan, Y. K. (2016). Design and analysis of a low cost PV analyzer using Arduino UNO. *In Power*

Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on (pp. 1-4). IEEE.

[3] Jamil, M., Khairan, A., & Fuad, A. (2015). Implementasi Aplikasi Telemedicine Berbasis Jejaring Sosial dengan Pemanfaatan Teknologi Cloud Computing. *Jurnal Edukasi dan Penelitian Informatika (JEPIN), 1(1).*

[4] Kusriyanto, M., & Putra, B. D. (2016). Smart home using local area network (LAN) based arduino mega 2560. *In Wireless and Telematics (ICWT), 2016 2nd International Conference on (pp. 127-131). IEEE.*

[5] Nusrath, A. (2015). Multipurpose Smart Card Using Advanced Encryption Standard Algorithm. *In International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), 115-118.*

[6] Pandey, B., Thind, V., Sandhu, S. K., Walia, T., & Sharma, S. (2015). SSTL Based Power Efficient Implementation of DES Security Algorithm on 28nm FPGA. *International Journal of Security and Its Application, 9(7), 267-274.*

[7] Ramaiya, M. K., Hemrajani, N., & Saxena, A. K. (2013). Security improvisation in image steganography using DES. *In Advance Computing Conference (IACC), 2013 IEEE 3rd International (pp. 1094-1099). IEEE.*

[8] Ramdhansyah, A. F., Ariyanto, E., & Nuha, H. H. (2015). Implementasi Advanced Encryption Standard (AES) Pada Sistem Kunci Elektronik Kendaraan Berbasis Sistem Operasi Android Dan Mikrokontroler Arduino. *In Seminar Nasional Informatika (SEMNASIF).*

[9] Rewagad, P., & Pawar, Y. (2013). Use of digital signature with diffie hellman key exchange and AES encryption algorithm to enhance data security in cloud computing. *In Communication Systems and Network Technologies (CSNT), 2013 International Conference on (pp. 437-439). IEEE.*