Design and Build Data Security Using Elliptic Curve Cryptography and Blowfish on IoT Tools in PDAM Companies

Bagus Miftah Rizqullah[1], Gusti Made Arya Sasmita, ST., MT.,[2], I Made Sunia Raharja, S.Kom., M.CS[3]

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Jl. Raya Kampus Unud, Jimbaran, Bali, Indonesia

Abstract — *PDAM (Perusahaan Daerah Air Minum) is a local government-owned company engaged in the distribution of clean water to the general public within the Indonesian local government. PDAM develops its technology and information systems specifically in the field of Internet of Things. The process handled is the prepaid PDAM transaction process which is carried out online. This prepaid PDAM transaction module really needs data security that can maintain the data transmission process on IoT devices. In this study, the design of a data security system using the PGP (Pretty Good Privacy) method is modified using the Eliptic Curve and BlowFish algorithms. The modified PGP method uses the SHA1 algorithm as a digital signature, Eliptic Curve cryptography is used as the Public Key encryption and Blow Fish cryptography is used as the symmetric encryption key. Attempted attacks with the man in middle attack cannot occur because the public key encryption uses the Eliptic Curve algorithm so it takes a very long hacking time.*

Keywords — *Pretty Good Privacy, SHA1, Elliptic Curve Cryptosystem, Blow Fish Cryptosystem, Base64, encryption, decryption, data security.*

I. INTRODUCTION

The very rapid development of information technology has implemented the Internet of Things to automate a device that can be controlled via the internet. The increasing development of IoT devices requires data security that can cover aspects of data security.

Data security aspects can be implemented using the PGP (Pretty Good Privacy) method. PGP method in the process of security uses a combination of a private key and a public key. Previous research conducted by I Wayan Dharma Satriawan, et al with the title "SMS Encryption Application with RSA Method on Android-Based Smartphones" proves that the modified PGP method can secure messages on android applications.

Implementation of the prepaid payment PDAM IoT tool that has transaction data that is vulnerable to theft of digital data via the internet network without being noticed by the data sending party, therefore it is necessary to manage digital data security using the modified PGP method where public key encryption and private key encryption use the Eliptic algorithm. Curve and Blowfish.

II. LITERATURE REVIEW

This chapter discusses the theoretical basis which contains an explanation of the supporting theories that will be used in research, namely the Elliptic Curve and Blowfish cryptography as a process of encryption and description of data in the process of sending data.

A. Kriptografi

Cryptography is the science of protecting information by converting it into a set of random, unreadable characters. Cryptography is effective for securing important information both stored in storage media and transmitted over communication networks.

B. Kriptografi Eliptic Curve

Elliptic curve cryptography is public key cryptography, each user or device will have a key pair which is a public key and a private key. Only users who have the same private key can use the private key, but the public key used will be distributed to those who will send data to the owner of the private key.

C. Kriptografi Blowfish

Blowfish cryptography has a key length of 64-bit to 448-bit. Each round consists of an Expansion key and data encryption. Expansion keys are generally used to generate the contents of an array and encrypt the data using the 16 round feiestel network. This study uses a 160-bit blowfish key length with the key method used is One Time Password (OTP) this method uses random characters that have a character length of 20-bytes and each key produces a different character.

D. SHA-1 Algorithm

SHA-1 is an update of the previous Secure Hash Algorithm (SHA), SHA is called safe because it is computationally designed not to provide messages that match the message digest. SHA1 is 20-bytes long. SHA-1 maps a fixed-length 160-bit string input.

E. Base64 Algorithm

Base64 is a technique for converting binary data into ASCII form. Base64 function is to hide important data and prevent unknown characters. The technique used by Base64 is similar to md5. The Base64 technique used in this study was to prevent unknown characters from the results of the previous method.

III. RESEARCH METHOD

The security protocol in the IoT PDAM prepaid payment tool is a system that can encrypt and decrypt data using two different algorithms.

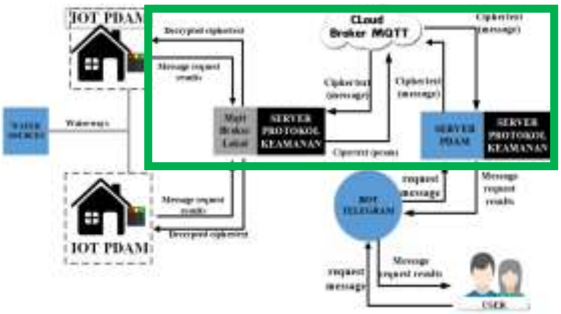


Fig1: General Description

The general description of the data security protocol system on the Internet of Things device that can be seen in the green line box is the limitation of the modified PGP security protocol system overview, where the IoT device will always pass the security protocol to encrypt and decrypt using the Eliptic Curve and Blowfish algorithms. The following are the stages of the data security process.

1. The user sends a request message to the IoT server

2.The IoT server sends a request message to the security protocol server for encryption which results in a ciphertex, this ciphertext is sent via the cloud to the IoT device.

3. The Raspberry Pi server receives a message / ciphertext where the ciphertext is decrypted by the PGP security protocol so as to produce a request or plaintext message

4. The request message is forwarded to the IOT tool using a local network connected to the Raspberry Pi server.

5. The IoT tool receives a request message and sends the request results via the local network to the Raspberry Pi server

6. The message of the request is encrypted using the PGP security protocol so that it produces a ciphertext from the message resulting from the request, this ciphertext is sent through the cloud network to the security protocol server.

7. The security protocol server receives ciphertext from the Raspberry Pi server, the ciphertext will be decrypted using the PGP security protocol so that it produces the original message or message resulting from the request, the message resulting from this request is sent locally to the IoT server

8. The IoT server receives a message from the request and is forwarded to the user.

A. PGP Encryption

This encryption system uses the PGP method which prioritizes integrity, confidentiality, availability, authentication, and non-repudiation as the basis for the PGP method. The following is a flow chart and an explanation of the encryption system in PGP.

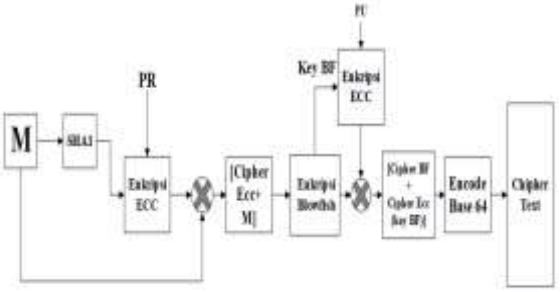


Fig2: PGP Encryption

Explanation:
M : Plaintext ECC: Enkripsi Eliptic Curve
PR: Private Key BF: Blow Fish
PU: Public Key SHA1 : Hasing with SHA1

The encryption process uses the PGP method where the message is hashing using SHA1, the value of SHA1 is encrypted using a private key and sends the public key to the recipient by leaving the message in the header so that the format becomes M#chiphertextEcc. The ciphertextEcc results are encrypted by the Blowfish algorithm using the OTP key so that the message will be encoded using Base64 which functions to convert the results to ASCII form.

B. PGP Description

This decryption system uses the PGP method which prioritizes integrity, confidentiality, availability, authentication, and non-repudiation as the basis for data security. The following is a flow chart and an explanation of the decryption system in PGP.

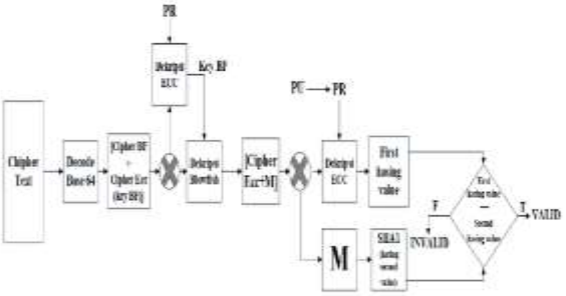


Fig3: PGP Description

Explanation:
PR : Private Key
PU : Public Key
M : Plaintext
ECC : Eliptic Curve Encryption

BF : Blow Fish
SHA1: Hasing uses SHA1
Valid : If the message has the same initial Hasing and End Hasing values
Invalid: If the message does not have the same value between the beginning and end of the case

The decryption process uses the PGP method where the ciphertext will be decoded. Base64 generates ciphertext from blowfish then decrypts Ecc with the private key to get OTP Blowfish and the next step is to take the public key from the Eliptic Curve then decrypt it using the private key, so that the ciphertextEcc will be decrypted and return to value initial hashing, then hashing message / M using SHA1 and the initial value of SHA1 will be compared with the final SHA1, if the same, then the message from the sender does not change but if it is different the message changes.

IV. RESULTS AND DISCUSSION

A. System Implemetation

This implementation uses a prepaid PDAM payment IoT Tool and Raspberry Pi which encrypts and decrypts messages.

a) IOT Server Send Message

The IoT server sends a request message to the IoT server to secure request messages from customers on the Windows PGP security server.

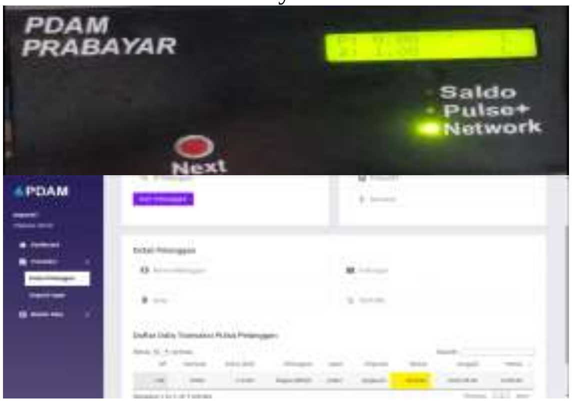
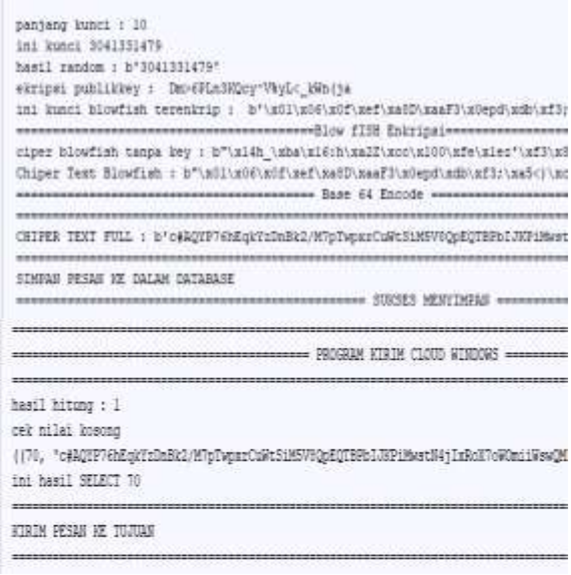


Fig4: IoT Server Send Message

Figure 4 is the process of sending request messages from customers which are forwarded by the IoT server to the security protocol, the security protocol will encrypt the request message with an initial balance on a one liter IoT device.

b) Windows PGP Security Securing Messages

The PGP security server secures messages received from the IoT server and the PGP security server secures the messages immediately.



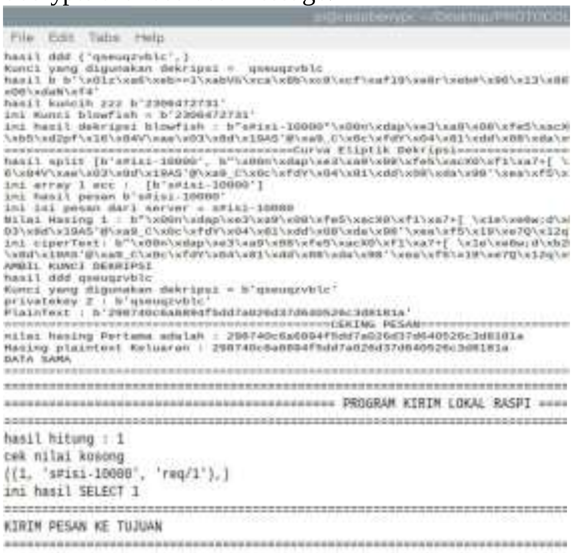
```
panjang kunci : 10
ini kunci 3041351479
hasil random : b"3041331479"
ekripsi publikkey :  Dm>€PLn3KQcy'V%yL<_kWh(j4
ini kunci blowfish terenkrip :  b'\x01\x06\x0f\xef\xa8D\xaaF3\x0epd\xdb\xf3;
=======================================Blow fISH Enkripsi===================
ciper blowfish tanpa key : b"\x14h_\xba\x16:h\xa2Z\xcc\x100\xfe\x1ez'\xf3\x8
Chiper Text Blowfish : b"\x01\x06\x0f\xef\xa8D\xaaF3\x0epd\xdb\xf3;\xa5<)\xc
======================================= Base 64 Encode ====================
============================================================================
CHIPER TEXT FULL : b'c#AQYP76hEqkYzDmBk2/M7pTwpxrCuWtS1M5V0QpEQTBPbIJKPiMwst
============================================================================
SIMPAN PESAN KE DALAM DATABASE
============================================= SUKSES MENYIMPAN ============
============================================================================
======================================= PROGRAM KIRIM CLOUD WINDOWS ========
============================================================================
hasil hitung : 1
cek nilai kosong
((70, 'c#AQYP76hEqkYzDmBk2/M7pTwpxrCuWtS1M5V0QpEQTBPbIJKPiMwstN4jIxRoX7oWOmiiWswQM
ini hasil SELECT 70
============================================================================
KIRIM PESAN KE TUJUAN
============================================================================
```

Fig5: Message Security Process

Figure 5 is the message encryption process using the PGP method where request messages from the IoT server are encrypted and saved to the database, after which the messages are stored through the Cloud network to be sent to IoT devices.

c) Raspberry pi 3 PGP server Receiving Message

The Raspberry Pi PGP server receives request messages from customers via the cloud which then decrypts the received message.



```
hasil ddd ('qseuqzvblc',)
Kunci yang digunakan dekripsi =  qseuqzvblc
hasil b b'\x01z\xa6\xeb==1\xabV6\xca\x0b\xc0\xcf\xaf19\xa8r\xeb#\x90\x13\x8
x00\xdaN\xf4'
hasil kuncih 222 b'2306472731'
ini Kunci blowfish = b'2306472731'
ini hasil dekripsi blowfish : b"s#isi-10000"\x00n\xdap\xe3\xa9\x00\xfe5\xacX
\xb5\xd2pf\x16\x04V\xae\x03\x8d\x19A5'@\xa9_C\x0c\xfdY\x04\x01\xdd\x08\xda\x
=========================================Curva Eliptik Dekripsi=============
hasil split [b's#isi-10000', b"\x00n\xdap\xe3\xa9\x00\xfe5\xacX0\xf1\xa7+[ \
6\x04V\xae\x03\x0d\x19A5'@\xa9_C\x0c\xfdY\x04\x01\xdd\x00\xda\x90'\xea\xf5\x
ini array 1 ecc :  [b's#isi-10000']
ini hasil pesan b's#isi-10000'
ini isi pesan dari server = s#isi-10000
Nilai Hasing 1 : b"\x00n\xdap\xe3\xa9\x00\xfe5\xacX0\xf1\xa7+[ \x1e\xe0w;d\x
03\x8d\x19A5'@\xa9_C\x0c\xfdY\x04\x01\xdd\x08\xda\x90'\xea\xf5\x19\xe7Q\x12q
ini ciperText: b"\x00n\xdap\xe3\xa9\x00\xfe5\xacX0\xf1\xa7+[ \x1e\xe0w;d\xb2
\x8d\x19A5'@\xa9_C\x0c\xfdY\x04\x01\xdd\x08\xda\x90'\xea\xf5\x19\xe7Q\x12q\x
AMBIL KUNCI DEKRIPSI
hasil ddd qseuqzvblc
Kunci yang digunakan dekripsi = b'qseuqzvblc'
privatekey 2 : b'qseuqzvblc'
Plaintext : b'298740c6a6084f5dd7a026d37d640526c3d8181a'
=============================================CEKING PESAN===================
nilai hasing Pertama adalah : 298740c6a6084f5dd7a026d37d640526c3d8181a
Hasing plaintext Keluaran : 298740c6a6084f5dd7a026d37d640526c3d8181a
DATA SAMA
============================================================================
============================================================================
========================================== PROGRAM KIRIM LOKAL RASPI ====
============================================================================
hasil hitung : 1
cek nilai kosong
((1, 's#isi-10000', 'req/1'),)
ini hasil SELECT 1
============================================================================
KIRIM PESAN KE TUJUAN
============================================================================
```

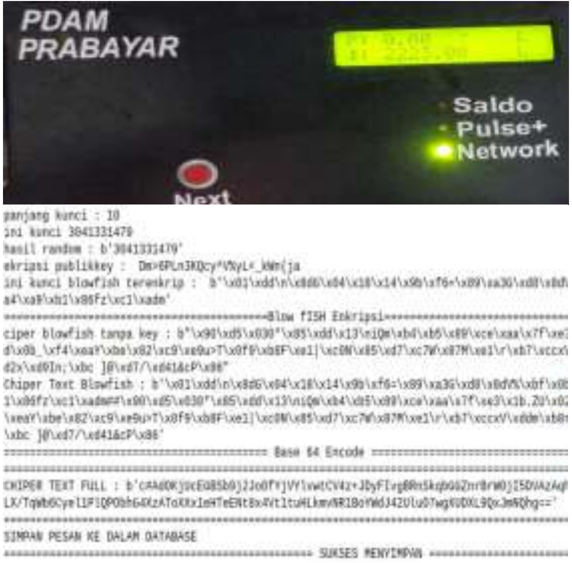
Fig6: PGP Raspberry Pi Receiving Messages

Figure 6 is the process of receiving encrypted and decrypted messages using a private key that has been defined in the database. The decryption results are stored in the database and sent over the local network according to the device address.

d) IoT Tool Receives Request

The IoT device receives a request message from the local network of the PGP security server on the

Rasberry Pi and the IoT device will fill the water pulse.



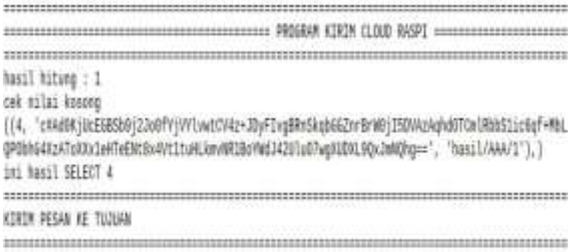
```
panjang kunci : 10
ini kunci 3041331479
hasil random : b'3041331479'
ekripsi publikkey :  Dm>6PLn3KQcy*VNyL<_kWm(ja
ini kunci blowfish terenkrip :  b'\x01\xdd\n\x8d6\x04\x18\x14\x9b\xf6=\x89\xa36\xd8\x8d
a4\xa9\xb1\x86Fz\xc1\xadm'
========================================Blow fISH Enkripsi================================
ciper blowfish tanpa key : b"\x90\xd5\x030"\x85\xdd\x13\niQm\xb4\xb5\x89\xce\xaa\x7f\xe
d\x0b_\xf4\xeaY\xbe\x82\xc9\xe9u>T\x0f9\xb8F\xe1|\xc0N\x85\xd7\xc7W\x87M\xe1\r\xb7\xccx
d2x\xd0In;\xbc ]@\xd7/\xd41&cP\x86"
Chiper Text Blowfish : b'\x01\xdd\n\x8d6\x04\x18\x14\x9b\xf6=\x89\xa36\xd8\x8dV%\xbf\x0
1\x86fz\xc1\xadm#\x90\xd5\x030"\x85\xdd\x13\niQm\xb4\xb5\x89\xce\xaa\x7f\xe3\x1b.Z0\x0
\xeaY\xbe\x82\xc9\xe9u>T\x0f9\xb8F\xe1]\xc0N\x85\xd7\xc7W\x87M\xe1\r\xb7\xccxV\xddm\xb8
\xbc ]@\xd7/\xd41&cP\x86'
========================================= Base 64 Encode ==================================
===========================================================================================
CHIPER TEXT FULL : b'c#Ad0KjUcEG8Sb0j2Jo0fYjVYlwwtCV4z+JDyFIvg8RnSkqbG6ZnrBrW0jI5DVAzAq
LK/TqWb6CyelIF1QPObhG4KzATxKXx1mHTeENt8x4Vt1tuHLkmvNR1BoYWdJ42Ulu07wgXUDXL9QxJmNQhg=='
===========================================================================================
SIMPAN PESAN KE DALAM DATABASE
======================================================= SUKSES MENYIMPAN ======================
```

Fig7: PDAM IoT Tools Receive Requests

Figure 7 is the process of filling water pulses on an IoT device that will send the message of the request. The message from the request will be received by the PGP security system and encrypt the message from the request, then the ciphertext is stored in the database.

e) Sending Message Request Results

Message from the request from the Raspberry Pi to the Windows server, this process is carried out after getting a message from the IoT device and encrypted by the PGP security system.



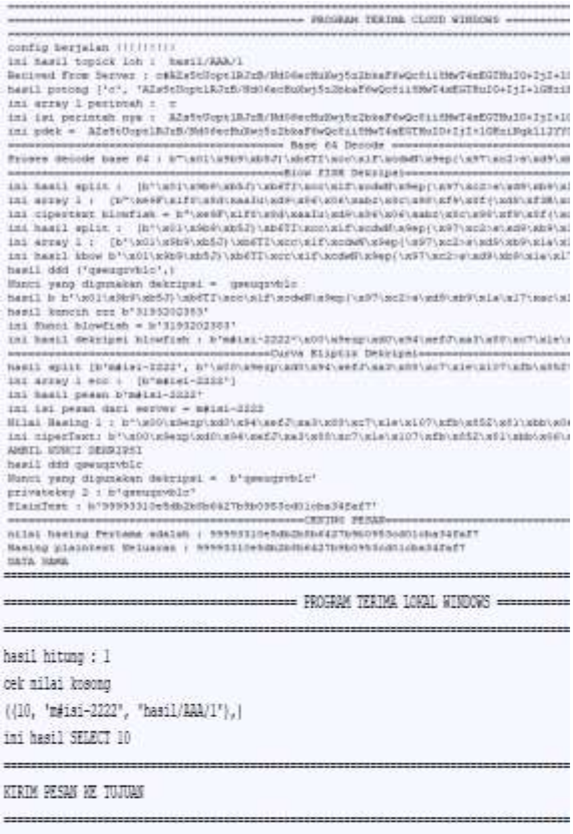
```
===========================================================================================
============================================ PROGRAM KIRIM CLOUD RASPI ====================
===========================================================================================
hasil hitung : 1
cek nilai kosong
((4, 'c#Ad0KjUcEG8Sb0j2Jo0fYjVYlwtCV4z+JDyFIvg8RnSkqbG6ZnrBrW0jI5DVAzAqhd0TOmlRbb51ic6qf+MbL
QPObhG4KzAToXXx1eHTeENt8x4Vt1tuHLkmvNR1BoYWdJ420lu07wgXUDXL9QxJmNQhg==', 'hasil/AAA/1'),)
ini hasil SELECT 4
===========================================================================================
KIRIM PESAN KE TUJUAN
===========================================================================================
```

Fig8: Send Result Requests to Windows Server

Figure 8 is the process of sending the request results from the Raspberry Pi server to the Windows server, where the destination address is the hasil/ AAA/1 and sent via the cloud network.

f) Windows Server Receives Result Message

Message results from the request from the Raspberry Pi server, where the message resulting from the request is sent according to the address and decrypted after being received by the Wimdows PGP server.



```
===========================================================================================
========================================== PROGRAM TERIMA CLOUD WINDOWS ===================
===========================================================================================
config berjalan !!!!!!!!!
ini hasil topick loh :  hasil/AAA/1
Recived From Server : c#AZe5tUopt1RJzB/Hd0ecHuXwj5zZbkaF6wQcfiiMwT4mEGTHuIO+IjI+1
hasil potong ['c', 'AZe5tUopt1RJzB/Hd0ecHuXwj5zZbkaF6wQcfiiMwT4mEGTHuIO+IjI+1G8
ini array 1 perintah :  c
ini isi perintah nya :  AZe5tUopt1RJzB/Hd0ecHuXwj5zZbkaF6wQcfiiMwT4mEGTHuIO+IjI+1
ini pdek =  AZe5tUopt1RJzB/Hd0ecHuXwj5zZbkaF6wQcfiiMwT4mEGTHuIO+IjI+1G8ziRgk11JY
========================================== Base 64 Decode ==================================
Prosee decode base 64 : b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\x
========================================Blow FISH Dekripsi==================================
ini hasil split :  [b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\xb9\x
ini array 1 :  [b"\xe6F\x1f\x8d\xaaIu\xd9\x94\x06\xabz\xcc\x90\xf9\xff(\xd8\xf38\x
ini cipertext blowfish = b"\xe6F\x1f\x8d\xaaIu\xd9\x94\x06\xabz\xcc\x90\xf9\xff(\x
ini hasil split :  [b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\xb9\x
ini array 1 :  [b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\xb9\x1a\x
ini hasil kbow b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\xb9\x1a\x1
hasil ddd ('qweuqrvblc',)
Kunci yang digunakan dekripsi =  qweuqrvblc
hasil b b"\x01\x9b9\xb5J)\xb6TI\xcc\x1f\xcdwM\x9ep(\x97\xc2>e\xd9\xb9\x1a\x17\xac\x
hasil kuncih zzz b'3193202383'
ini Kunci blowfish = b'3193202383'
ini hasil dekripsi blowfish : b'm#isi-2222"\x00\x9exp\xd0\x94\xefJ\xa3\x88\xc7\x1e\
========================================Curva Eliptik Dekripsi=============================
hasil split [b'm#isi-2222', b"\x00\x9exp\xd0\x94\xef\xa3\x88\xc7\x1e\x107\xfb\x85f
ini array 1 ecc :  [b'm#isi-2222']
ini hasil pesan b'm#isi-2222'
ini isi pesan dari server = m#isi-2222
Nilai Hasing 1 : b"\x00\x9exp\xd0\x94\xef2\xa3\x88\xc7\x1e\x107\xfb\x852\x81\xbb\x0
ini ciperText: b"\x00\x9exp\xd0\x94\xef2\xa3\x88\xc7\x1e\x107\xfb\x852\x81\xbb\x06\
AMBIL KUNCI DEKRIPSI
hasil ddd qweuqrvblc
Kunci yang digunakan dekripsi =  b'qweuqrvblc'
privatekey 2 : b'qweuqrvblc'
PlainText : b'99993310e5db2b0b6427b9b0953c801cba345ef7'
=============================================CEKJIKI PESAN=================================
nilai hasing Pertama adalah : 99993310e5db2b0b6427b9b0953c801cba345ef7
Hasing plaintext Keluaran : 99993310e5db2b0b6427b9b0953c801cba345ef7
DATA SAMA
===========================================================================================
======================================== PROGRAM TERIMA LOKAL WINDOWS =====================
===========================================================================================
hasil hitung : 1
cek nilai kosong
((10, 'm#isi-2222', 'hasil/AAA/1'),)
ini hasil SELECT 10
===========================================================================================
KIRIM PESAN KE TUJUAN
===========================================================================================
```

Fig9: Receive and send Request Results

Figure 9 is the process of receiving the request results where when an incoming message is decrypted using a key set on the Windows server, the result of the request message will be sent to the I0T server via the local network.

g) IOT Server Receives Results

The IoT server receives the requested message from the PGP security server after the decryption process is done. Delivery will be made based on the destination address that has been stored in the database.

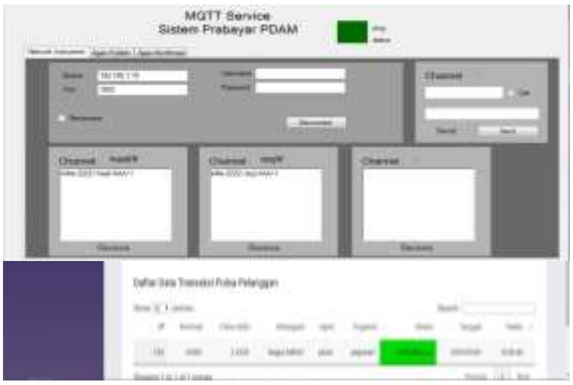


Fig9: Results of Request for Water Top-up

Figure 9 is the result of the process of securing data on the IoT server using a modified PGP security system. The data resulting from the request message is not lacking or changing and this process takes 4 seconds to secure data from the Windows server to the Raspberry Pi server.

B. Experiment with Various Text Data

Experiment with various data such as letter combinations, numbers, and the number of very long sentences This experiment aims to find out how many characters the PGP security system can secure.

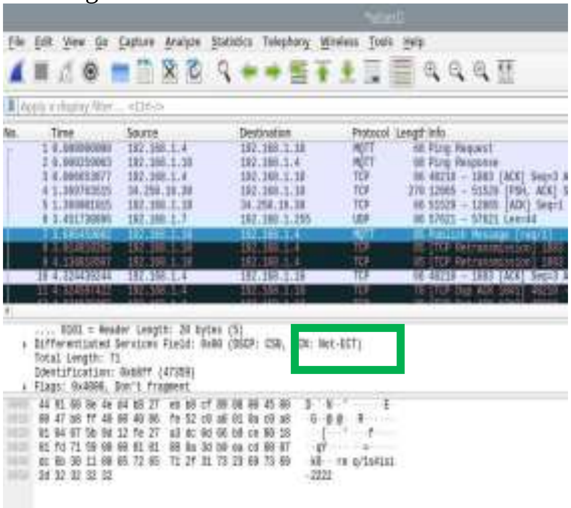
TABLE I Experiments of Various Text Data

NO	Data Text/Size	After Encryption	Character increase
1	Test1 (5)	176	3420%
2	Keamanan (8)	188	2250%
3	Algoritma ECC (12)	188	1470%
4	Algoritma ECC dan Blowfish (23)	208	8000%
5	Proses Mengamankan Data-Data (26)	208	7000%
6	ABCDEFGHIJK LMNOPQRSTUVW XYZ123455678910 (38)	220	4500%

Table 1 shows the results of data that have several characters with a few and a lot of characters. Experiments with various text data prove that the modified PGP method works 100% safe to use on text data because it uses data security standards and Public Key encryption.

C. Network Scan Experiment with WireShark

This experiment is conducted to prove data security on a modified PGP server, this experiment uses WireShark as a supporting application for scanning local networks.

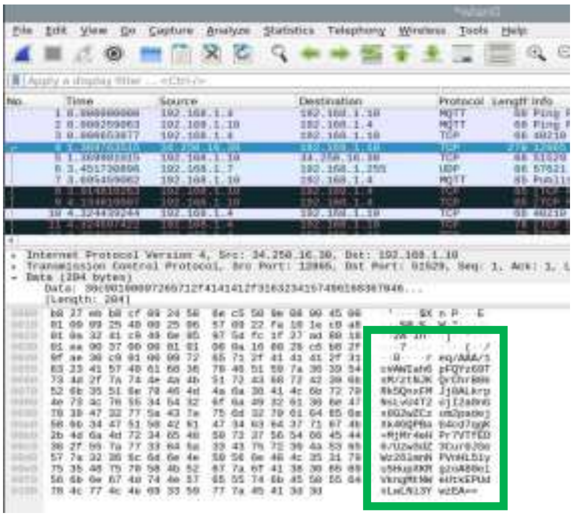


```
No.  Time         Source        Destination   Protocol Lengt Info
   1 0.000000000  192.168.1.4   192.168.1.10  MQTT      60 Ping Request
   2 0.000259063  192.168.1.10  192.168.1.4   MQTT      60 Ping Response
   3 0.000653877  192.168.1.4   192.168.1.10  TCP       66 48210 → 1883 [ACK] Seq=3
   4 1.309703515  34.250.16.30  192.168.1.10  TCP      270 12005 → 51529 [PSH, ACK]
   5 1.309881015  192.168.1.10  34.250.16.30  TCP       66 51529 → 12005 [ACK] Seq=1
   6 3.451730806  192.168.1.7   192.168.1.255 UDP       86 57621 → 57621 Len=44
   7 [illegible]  192.168.1.10  192.168.1.4   MQTT      85 Publish Message [tag/1]
   8 [illegible]  192.168.1.10  192.168.1.4   TCP       85 [TCP Retransmission] 1883
   9 4.138050807  192.168.1.10  192.168.1.4   TCP       85 [TCP Retransmission] 1883
  10 4.324439244  192.168.1.4   192.168.1.10  TCP       66 48210 → 1883 [ACK] Seq=3
  11 [illegible]  192.168.1.4   192.168.1.10  TCP       78 [TCP Dup ACK ...] 48210

    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 71
    Identification: 0xb8ff (47359)
  ▸ Flags: 0x4000, Don't fragment

[illegible]
```

Fig10: Before Encrypted



```
No.  Time         Source        Destination   Protocol Length Info
   1 0.000000000  192.168.1.4   192.168.1.10  MQTT      60 Ping R
   2 0.000259063  192.168.1.10  192.168.1.4   MQTT      60 Ping R
   3 0.000653877  192.168.1.4   192.168.1.10  TCP       66 48210
   4 1.309703515  34.250.16.30  192.168.1.10  TCP      270 12005
   5 1.309881015  192.168.1.10  34.250.16.30  TCP       66 51529
   6 3.451730806  192.168.1.7   192.168.1.255 UDP       86 57621
   7 3.605459062  192.168.1.10  192.168.1.4   MQTT      85 Publis
   8 3.914810283  192.168.1.10  192.168.1.4   TCP       85 [TCP
   9 4.138050807  192.168.1.10  192.168.1.4   TCP       85 [TCP
  10 4.324439244  192.168.1.4   192.168.1.10  TCP       66 48210
  11 4.324597422  192.168.1.4   192.168.1.10  TCP       78 [TCP

▸ Internet Protocol Version 4, Src: 34.250.16.30, Dst: 192.168.1.10
▸ Transmission Control Protocol, Src Port: 12005, Dst Port: 51529, Seq: 1, Ack: 1, L
▾ Data (204 bytes)
    Data: 30c001000007266712f4141412f3163234157480168387046...
    [Length: 204]

[illegible]
```

Fig11: After Encrypted

Network scanning using WireShark aims to ensure that incoming and outgoing data is secured by the modified PGP security protocol. The results of network testing by sniffing are 100% safe because the encrypted data is different from the original data.

V. CONCLUSION

The conclusion of the modified PGP security system using the Eliptic Curve and Blowfish algorithms is that it can successfully use public key encryption to secure text data on the integrated PDAM prepaid IoT payment tool. Text data is converted into ciphertext which is difficult to translate and tested by scanning the network using Wireshark. The results of the network scanning trial are data that appear in the form of ciphertext so that the original data is unknown. The modified PGP method is also useful for information systems and IoT tools that will be developed.

VI. REVERENCES

[1] Satriawan, I. W. D., Made, I. G., Sasmita, A., & Bayupati, I. P. A. (2016). Aplikasi Enkripsi Sms Dengan Metode RSA Pada Smartphone Berbasis Android. Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi), 2(2), 127–134.

[2] Piarsa, I. N. (2012). Steganografi Pada Citra Jpeg Dengan Metode Sequential Dan Spreading. Lontar Komputer, 2(1), 52–63.

[3] Sibarani, E. B. H., Zarlis, M., & Sembiring, R. W. (2017). Analisis Kripto Sistem Algoritma Aes Dan Elliptic Curve Cryptography (Ecc) Untuk Keamanan Data. InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan), 1(2), 106–112. https://doi.org/10.30743/infotekjar.v1i2.71.

[4] Damanik, P. S. E. A. (2019). Implementasi Algoritma Elliptic Curve Cryptography (ECC) Untuk Penyandian Pesan Pada Aplikasi Chatting Client Server Berbasis Desktop. Jurnal Riset Komputer (JURIKOM), 6(4), 395–400.

[5] Febrianto, A., & Apriani, J. (2017). "*PERBANDINGAN SISTEM PENGAMANAN EMAIL MENGGUNAKAN TEKNIK PUBLIC KEY ENCRYPTION DAN PRETTY GOOD PRIVACY (PGP)*" (Studi Kasus : AMIK Dian Cipta Cendikia Bandar Lampung). Jurnal Cendikia, 13(2), 17–25.

[6] Ivan Kurniawan Prasetyo. (2012). My study in Information Technology: Fungsi Hash MD5 dan SHA-1. 11 Juli 2012.

http://studyinformatics.blogspot.com/2012/07/fungsi-hash-md5-dan-sha-1.html

[7] jaya santoso sirait, Rumani, & Paryasto, M. W. (2017). Implementasi Kriptosystem menggunakan metode Algoritma ECC dengan Fungsi Hash SHA-256 pada sistem ticketing online Implementation of Crypthosystem using Method Algorithm ECC with Function of Hash SHA- 256 in online ticketing system. E-Proceeding of Engineering, 4(3), 4138–4146.

[8] Kurniawan, D. (2018). Perencanaan Aplikasi Pengamanan Data Text Menggunakan Blowfish dan RC6. Pelita Informatika Budi Darma, 17, 254–260.

[9] Laila, N., & Sinaga, A. S. R. (2019). Implementasi Steganografi LSB Dengan Enkripsi Vigenere Cipher Pada Citra. ScientiCO : Computer Science and Informatics Journal, 1(2), 47. https://doi.org/10.22487/j26204118.2018.v1.i2.11221

[10] Meko, D. A. (2018). Jurnal Teknologi Terpadu Perbandingan Algoritma DES , AES , IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data Donzilio Antonio Meko Program Studi Teknik Informatika , STIMIK Kupang Jurnal Teknologi Terpadu. Jurnal Teknologi Terpadu, 4(1), 8–15.

[11] Pramana Hostiadi, D., & Suradarma, I. B. (2017). Implementasi Pengamanan PGP Pada Platform Zimbra Mail Server. Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, 8(1), 41. https://doi.org/10.24843/lkjiti.2017.v08.i01.p05

[12] Shruti Sekra, Samta Balpande, Karishma Mulani, "*Steganography Using Genetic Encryption Along With Visual Cryptography"* SSRG International Journal of Computer Science and Engineering 2.1 (2015): 1-5.