International Journal on
Advanced Science
Engineering
Information Technology

Vol.9 (2019) No. 5
ISSN: 2088-5334

Secure e-Health Record System Using Identity-based Encryption with Embedded Key

Dian Neipa Purnamasari[#1], Amang Sudarsono[#2], Prima Kristalina[#3]

[#] Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia
E-mail: [1]dneipa12@gmail.com, [2]amang@pens.ac.id, [3]prima@pens.ac.id

***Abstract*— The existence of electronic health record in the Internet world can result in the emergence of potentially unauthorized users to access and abuse the data. Therefore, it is necessary to have a high level of security that can maintain the confidentiality of electronic health record data. In cryptography, the longer the key then, the higher the level of security achieved. However, this can lead to slow computing time. Therefore, we propose a security method with an identity-based encryption scheme that is built hybrid using elliptic curve cryptography (ECC) and elliptic curve integrated encryption system (ECIES) algorithms or can be abbreviated as IBE-ECC-ECIES. An additional feature of the proposed method is the creation of key pairs generated by the ECC algorithm and there is an identity that has been embedded in the key pair in order to increase the level of security and uniqueness of the key. This method has been compared based on analysis of performance, computation time and the level of security in the same environment. This method is another IBE hybrid scheme called IBE-ECC-AES. The results of the test showed that the proposed method was superior to 0.3 seconds compared to the comparison method in the key pair generation process. In addition, security in the proposed method can overcome sniffing and chosen-plaintext attacks.**

***Keywords*— electronic health record; IBE; ECC; ECIES; embedded key.**

I. Introduction

Nowadays many health services are connected to the Internet; this can open the gap for information leakage such as the disclosure of the patient's electronic health record data. Health record data is confidential data so it must be protected from users who want to abuse the data. Based on these problems, many researchers have proposed security methods that can protect and overcome the misuse of medical record data.

The concept of Identity-based Cryptosystem (IBC) using the signature scheme was first proposed by Shamir in 1984 [1]. The proposed concept allows users to communicate safely, verify signatures without exchanging keys and without using third parties. After that, many researchers proposed identity-based encryption schemes (IBE) that could be classified as pairing-based and non-pairing. IBE schemes based on bilinear pairs on elliptic curves can be pairs of Weil and Tate [2]–[5]. While some non-pairing based IBE schemes are [6]–[9].

Pairing-based cryptography (PBC) is based on a pair function that maps pairs of points on elliptic curves over finite fields. Pairing is useful in cryptography if it is built correctly so it can produce limited fields large enough to create discrete logarithmic problems that are difficult to calculate. The practicality of the IBE scheme is hindered by the calculation of complex discrete logarithms and causes longer computation time so that this scheme is not suitable if implemented on systems that require high data mobility. The researchers began to turn to non-pairing schemes such as RSA, which is considered a practical solution to overcome computational complexity. Himanshu et al. [9] proposed that the IBE scheme could replace traditional SSL to eliminate the need for site certificates. The proposed IBE scheme is integrated with RSA to produce key pairs. The strength of RSA lies in the level of difficulty in factoring numbers into prime factors so that the key generation process will produce a large enough keyspace. This excess all at once into a weakness in the RSA secret key length that is too large will result in computing time on the process of decryption becomes high and the system becomes unreliable. The solution to reducing the computing time at RSA is to use Elliptic Curve Cryptography (ECC). ECC offers the same level of security with shorter key lengths. Comparisons between RSA and ECC key lengths are shown in Table I.

At present, there are three applications in elliptic curve cryptography namely the Elliptic Curve Diffie Hellman (ECDH), Elliptic Curve Digital Signature Algorithm (ECDSA), and the Elliptic Curve Integrated Encryption System (ECIES). ECDH is an elliptic curve variant that applies a key exchange scheme based on the Diffie-Helman

(DH) mechanism. ECDSA is a variant of the elliptic curve of a digital signature (DSA). The ECIES is an elliptic curve variant using a hybrid scheme. The hybrid scheme used is a combination of asymmetric elliptic curve encryption with symmetric AES. In other words, we make symmetric algorithms from the elliptic curve cryptography.

TABLE I
KEY LENGTH COMPARISON OF RSA AND ECC [10]

Security Level (bits)	**RSA Key Length (bits)**	**ECC Key Length (bits)**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

In this work, we propose a security method to secure health data when data is exchanged. The proposed security method is an IBE scheme that is built hybrid using the ECC and ECIES algorithms or can be abbreviated as IBE-ECC-ECIES. The ECC algorithm is used to generate key pairs, while the ECIES algorithm is used for the encryption and decryption process. This research is advanced research from D. N. Purnamasari et al. [11] which uses the IBE scheme in the same environment, namely the exchange of data on multiple servers. To add a security level to the system, the user's identity is embedded in the key pair. The purpose of this security method is to avoid key exchanges in communication channels, limit user access, and reduce computing time in generating key pairs. This security method will be compared with other hybrid security methods, namely IBE-ECC-AES. This method is a method of development from some research [12]–[14] which combines algorithm ECC and AES. The selection of the AES algorithm is because the proposed method uses the ECIES algorithm, which can convert public keys to elliptic curve cryptography into symmetrical keys. In fact, the AES algorithm also uses symmetrical keys. Therefore, the two algorithms are equal to compare. The purpose of this research is to analyze and test the performance of the proposed security method.

We structure the remainder of this paper as follows: Section II describes some supporting material and the proposed method. Section III discusses the results of the analysis of performance, computation time, and level of security of the proposed method. The results of the conclusions are summarized in Section IV.

II. MATERIAL AND METHOD

In this section, we briefly explain some of the research that has been done relating to this paper, an explanation of the IBE scheme in general, an explanation of the IBE scheme adopted and a more detailed explanation of the security methods proposed in this paper.

A. Related Works

Several researchers propose security solutions with various cryptographic techniques such as symmetry, asymmetry, and hashing. In a previous study conducted by A. Sudarsono et al. [5] proposed a security solution for the e-Healthcare system by sharing data using the IBE scheme. The algorithm used is the development of Boneh-Franklin (BF-IBE), which has been combined with the AES algorithm. The proposed IBE uses arbitrary strings as public keys. This solution uses a hybrid scheme by specifying the security of 160-bit key lengths of elliptic curve cryptography and the AES algorithm for the encryption and decryption process. The eHealthcare system is suitable for embedded devices or other mobile devices.

C. C. Tan et al. [8] describes the development of the IBE scheme in the body sensor network (BSN) for patient health monitoring. This system is called IBE-Lite, which aims to alleviate the work of IBE by maintaining the properties of conventional IBE and can be run on BSN sensors. The algorithm used is ECC. The use of IBE-Lite so that sensors can produce public keys using arbitrary strings and this sensor cannot create a secret key to decrypt messages.

Himanshu et al. [9] proposed a solution to eliminate the need for site certificates using the IBE scheme. The RSA algorithm is used to generate key pairs and decryption encryption processes. The strength of RSA lies in the level of difficulty in factoring numbers into prime factors so that the key generation process will produce a considerable keyspace. In this scheme, a Public Key Generator (PKG) is required to serve as a key server. There are 4 stages in the proposed scheme including setup, encrypt, extract and decrypt.

S. Al-Alak et al. [12] proposed the security protocol on the ZigBee wireless sensor network at the MAC layer using the AES and ECC algorithms. The purpose of this study is to increase confidentiality in the MAC layer and make a defense against replay attacks. The AES algorithm is used to increase the confidentiality of the MAC layer, while the ECC algorithm is used by the Multiple-key Protocol (MKP) to protect the key in the form of a previous or subsequent key. The number of keys generated from ECC is calculated based on the level of security. The results of the study found that ZigBee applications that use the proposed protocol have a higher level of security than without using it.

S. A. Abbas et al. [15] has proposed a safe and effective method to improve security in cloud computing. The author uses the ECIES algorithm in the modified identity-based cryptography (MIBC) scheme. The use of MIBC is needed to reduce the complexity of key generation and certificate requirements while the ECIES algorithm to provide data confidentiality and data integrity. In the proposed scheme there are 3 important parts in the system, namely the Trusted Authority (TA), Trusted Cloud (TC) and Users. TA is used to generate system parameters and select random private numbers, while users can generate their own key pairs using ECC. Users encrypt information data using ECIES with their own private key and then send it to TC. TC will store user cipher text data without decryption beforehand so that the confidentiality of the data can be maintained. Three stages in the MIBC scheme are proposed including encryption, key generation, and decryption.

K.-L. Tsai et al. [16] which allows authorized users to access and distribute health data anytime and anywhere. Longer encryption key then the level security will be higher but the computing time would cause the longer and worsen the performance of the transmission. The results of the study state that the ECC algorithm is more suitable to be implemented on portable devices. This aims to reduce the burden of communication, improve the security of transmission, and maintain the quality of health data.

M. Yuliana et al. [17] explained the development of an e-health system for inpatients. The method used is the ECC method to design the access structure of users who are authorized and the ECDSA method to verify patient data access. Based on the design of the access structure created, only registered users can access patient health data.

B. Identity-based Encryption (IBE)

IBE is one of many variants in public key cryptographic techniques. IBE was first proposed by Shamir in 1984 [1]. The advantages of the IBE is the public key used can be either an identity card number, username, e-mail address, phone number, home address or any other information that can uniquely represent the identity of a person. Another advantage of this encryption technique is that it is not necessary to determine key pairs before encryption.

The IBE scheme is divided into 4 stages with detailed schemes such as the following [18]:

- Setup, which is taking the parameters needed for determining system parameters in the form of a private key (msk) and public key (pkid).
- Extract, which is an extraction algorithm with input private key and user identity (id). The output is a private key (skid).
- Encrypt, which is an encryption algorithm by inputting health data information (m), an identity (id), and a public key (pkid). The output is ciphertext (C).
- Decrypt, which is a decryption algorithm with ciphertext (C) input, an identity (id), and corresponding private key (skid). The output is health data information (ḿ).

C. Identity-based Encryption (IBE) – RSA

The basic idea of IBE-RSA is a public RSA n modulus that is used in certain IBE schemes. In IBE-mRSA a Certificate Authority (CA) is needed that can determine and store the RSA modulus value that is public. There are 4 stages with detailed schemes like the following [11]:

1) Setup: This stage is in the Private Key Generator (PKG) as a key server. Before communication, supporting information needed for the encryption, decryption process was obtained from PKG. to produce a key pair, a selection of values from p, and q is carried out based on the number of RSA bits used. The value of r is obtained from a random number to test the prime numbers of p and q. The result of multiplying p and q is n, the params distributed on each user registered on PKG. The set of the simple residue of a modulo n is expressed in Equation 1.

$$\phi(n) = (p-1)(q-1) \qquad (1)$$

The value of e is obtained based on the relatively prime value of ϕ (n). The public master key used for encryption is pair (e, n). Furthermore, the secret key is generated using Equation 2.

$$e.\ d \equiv 1 \ (\mathrm{mod}\ \phi(n)) \qquad (2)$$

While the d value is the master secret key used for the decryption process.

2) Encrypt: The purpose of this stage is to process the health record data encryption. The system will request at PKG to get a master public key and n. Health records will be converted into a block and will be encrypted by modulo n.

3) Extract: At this stage, a user verification process is if the user's identity is the same as the registered identity, the user will get the master secret key and params. Then the secret key master will be extracted into a secret key.

4) Decrypt: The purpose of this stage is to decrypt the ciphertext using the params and secret key. Ciphertext will be converted into blocks that will be decrypted by modulo n.

D. The Proposed Security Method

The proposed security method uses IBE scheme is a hybrid constructed using ECC non-pairing and ECIES. ECC non-pairing is used to generate a key pair, namely a public key and a secret key. ECC has advantages in terms of key lengths compared to other asymmetric cryptography. The shorter difference in key length does not limit to having the same level of security. The parameters of the elliptic curve domain suggested by Fp include P-192, P-224, P-256, P-384 and P-521 [19]. ECIES is used for the encryption and decryption process. This is because ECIES uses a hash algorithm to produce a message digest [20], so it can reduce computing time in the encryption and decryption process.

In the IBE scheme, the public key can be generated based on the user identity so that it does not require a third party to generate the key pair. IBE scheme in this paper has four stages, among others Setup, Encrypt, Decrypt and Extract. To be clearer, Algorithm 1 describes the public key generation process in Setup.

Algorithm 1 `Setup`

```
1. Obtain the recipient's identity ID, Elliptic Curve P, and
   generate a secure random number r
2. Calculate r = r.setSeed(seed) where seed is
   a hash function of a recipient's identity
3. Calculate PUBLIC_KEY (Q) = (P,r)
```

`Setup`: to generate a public key `Q`, the sender requires a public identity of the recipient. The recipient's identity is needed so that the generated public key is unique and only for the recipient. In Algorithm 1 line 2, the recipient's identity is embedded in the public key. Before communicating, the sender and receiver must determine the elliptic curve to be used so that the message can be sent and received without obstruction.

Algorithm 2 `Encrypt`

```
Obtain health record data M, recipient's identity ID,
Elliptic Curve P, and generate a secure random number r
Calculate r = r.setSeed(seed) where seed is
a hash function of a recipient's identity
Calculate $c_1$ = Encrypt(r,P)
Calculate $c_2$ = Encrypt(M,r,Q)
```

`Encrypt` to encrypt medical record data; the public key is required in the Setup stage and other supporting information such as the approved elliptic curve and the recipient's identity. At this stage, the system will produce two ciphertexts, one of which contains a message of medical record data. In Algorithm 2 lines 3 and 4, the encryption algorithm used is 256-bit AES in ECB mode. 256-bit AES has 14 rounds with four processes namely AddRoundKey, SubBytes, ShiftRows, and MixColumns. In this paper twice encryption is carried out, namely encryption for health record data and encryption for supporting parameters. Before starting the encryption process, the health record data and supporting parameters will be converted into 128 bits block ciphers and entered into the 4x4 matrix, commonly called the initial state. Then the system will combine the state with the cipher key with XOR and proceed with passing the round 13 times. Each round runs the SubBytes process (byte substitution with an S-box table), ShiftRows (shifting each matrix element), MixColumns (multiplying each element with a matrix), and AddRoundKey (XOR calculation between new states with round keys). Then run the final round, which only contains the SubBytes, ShiftRows, and AddRoundKey processes. The results of the final round produce C1 and C2. Each ciphertext contains embedded identity in r so that only the intended recipient can open and know the medical records were sent.

Algorithm 3 `Extract`

```
1. Obtain user identity id, identity submitted ID, and
   generate a secure random number r
2. If (id.equals(ID)) then
3.     Calculate r = r.setSeed(seed) where seed is
       hash function of ID
4.     SECRET_KEY (S) = (r)
5. else
6.     SECRET_KEY (S) = "null"
7. end if
```

`Extract`: to get the secret key `S` that matches the public key, the user verifies the identity sent by the sender. If the identity is the same as the user's identity then the process is run on Algorithm 3 lines 3 and 4 to get the secret key.

Algorithm 4 `Decrypt`

```
Obtain ciphertext c1 c2, and secret key S
Execute Decrypt(c1, c2, S) to obtain M
```

`Decrypt:` to run the decryption process, the recipient must have the secret key S obtained from the Extract stage. C1 and C2 are entered into the 4x4 matrix and pass 14 rounds. The process passed is the opposite of the encryption process so that each round has four processes, namely AddRoundKey, Inverse MixColumns, Inverse ShiftRows, and Inverse SubBytes. The result of the decryption process is the original message M. The decryption process is shown in Algorithm 4.

III. Results and discussion

The proposed method has been implemented in the simulation uses client and multiple server scenarios are illustrated in Fig. 1. The IBE scheme is built hybrid using ECC non-pairing and ECIES. The purpose of this study is to analyze and test the performance of the proposed scheme. There are several parameters for testing including elliptic curve variants, the amount of data sent, the effect of the hardware used, and the level of security of the proposed method.

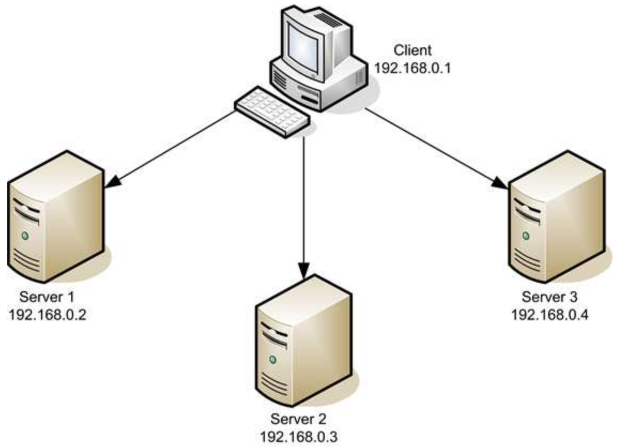


Fig. 1 Illustration of Communication Client and Multiple Server

Testing the proposed method is done by simulating using the Linux operating system (OS). We use four host OS, including 3 as multiple servers and 1 as a client. Communication between clients and multiple servers uses socket programming, where multiple servers and clients are connected to a LAN network. The client will send health data that has been encrypted using the IBE scheme. The results of the encryption process and the original data will represent the truth of the health data sent. The following hardware specifications used on the server and client are shown in Table II.

TABLE II
Server and Client Hardware Specifications

CPU	Intel Pentium CPU 2020M 2.40 GHz
Host O/S	Windows 7 Ultimate 64-bit
RAM O/S	6 GB
Software	Virtual Box v5.2.8 , Java 1.8.0_171
VM O/S	Debian GNU/Linux 8 (Jessie)
RAM VM O/S	1 GB

A. Performance analysis on the IBE scheme

In this test, the proposed method will be compared with other IBE hybrid schemes, namely IBE-ECC-AES that illustrated in Fig. 2. This scheme is used as a comparison because ECIES is a standard elliptic curve based encryption algorithm using symmetrical and asymmetric key concepts [21], while AES is an encryption algorithm that uses symmetrical key concepts.

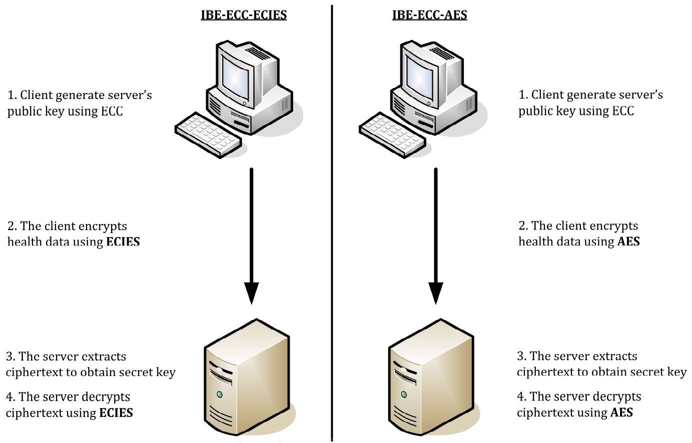

IBE-ECC-ECIES
IBE-ECC-AES
1. Client generate server's public key using ECC
2. The client encrypts health data using ECIES
3. The server extracts ciphertext to obtain secret key
4. The server decrypts ciphertext using ECIES
1. Client generate server's public key using ECC
2. The client encrypts health data using AES
3. The server extracts ciphertext to obtain secret key
4. The server decrypts ciphertext using AES

Fig. 2 Illustration of performance analysis on the IBE scheme

This test is carried out in a simulation using the concept of client and server connected to the same network. The initial data used is patient health monitoring data. The patient data format used in this test is shown in Table III.

TABLE III
PATIENT DATA FORMAT

Data Format	Data Length	Description
Flag	1 byte	"#"
ID	10 bytes	Identity of Patient
Delimiter	1 byte	"|"
Timestamp	19 bytes	Timestamp
Delimiter	1 byte	"|"
Data	7 bytes	Blood Pressure Data
Delimiter	1 byte	"|"
Data	3 bytes	SPO2 Data
Delimiter	1 byte	"|"
Data	2 bytes	Airflow Data

The stages in the IBE scheme that will be observed are setup, encrypt, extract and decrypt. Each stage will be compared according to the elliptic curve used. There are 5 elliptic curves that have been recommended by Fp, including P-192, P-224, P-256, P-384 and P-521 [19]. Testing is done by sending 10 data in one shipment for 10 times.

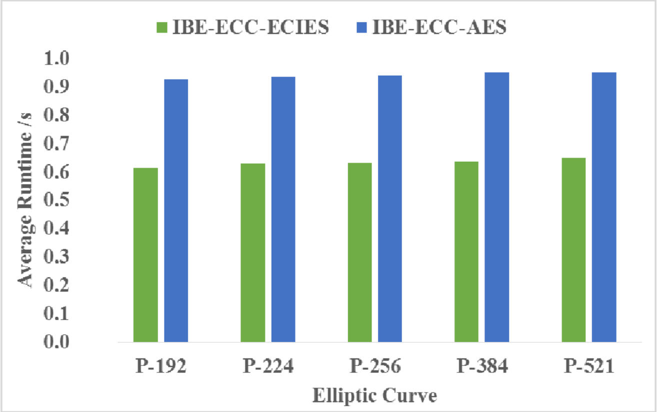


Fig. 3 Comparison of IBE-ECC-ECIES and IBE-ECC-AES in the Setup Stage

In the setup stage, the system will generate the public key needed for the encryption process. The first thing to do is to determine the elliptic curve that will be used. This elliptic curve is based on GF(p) where p is a large prime number. Fig. 3 shows that to produce a public key for each elliptic curve variant, IBE-ECC-AES takes longer than IBE-ECC-ECIES. This is because the ECIES algorithm is an ECC-based encryption algorithm so that the public key generated by ECC can be directly used in the IBE-ECC-ECIES scheme, while the IBE-ECC-AES is needed additionally to convert public keys, which initially use elliptic curves with asymmetrical concepts to be symmetrical. Computation time at the setup stage required by the proposed method is 0.6 s while the comparison method requires the time of 0.9 s. Computation time differences in both schemes ranged from 0.3 s. The next step is encrypt stage, the results of computational time comparisons between IBE-ECC-ECIES and IBE-ECC-AES will be shown in Table IV.

TABLE IV
COMPARISON OF IBE-ECC-ECIES AND IBE-ECC-AES AT THE ENCRYPT STAGE

Elliptic Curve	IBE-ECC-ECIES (s)	IBE-ECC-AES (s)
P-192	0,210	0,048
P-224	0,218	0,048
P-256	0,221	0,049
P-384	0,225	0,050
P-521	0,227	0,053

In the encrypted stage, there is an encryption process of health data with a public key obtained at the setup stage. In Table IV it is found that IBE-ECC-ECIES has a slower time than IBE-ECC-AES. This is because the IBE-ECC-ECIES scheme uses an asymmetric ECIES algorithm based on standard elliptic curves to encrypt so that the time taken will be longer when compared to the IBE-ECC-AES scheme that uses AES symmetric algorithms for the encryption process. The difference in the encryption algorithm used affects the computation time at the encrypt stage, the proposed method has a computation time of 0.2 s while the comparison method has a time of 0.05 s or 4 times faster than the proposed method. The result of the encryption process is a ciphertext containing encrypted health data that will be sent to the server. The ciphertext received by the server will be decrypted using a secret key. The secret key is obtained in the extract stage by verifying the user's identity. The results of computational time comparisons between IBE-ECC-ECIES and IBE-ECC-AES will be shown in Fig. 4.

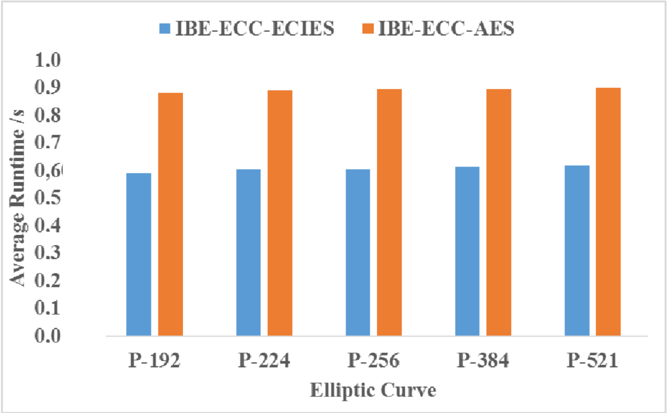


Fig. 4 Comparison of IBE-ECC-ECIES and IBE-ECC-AES at the Extract Stage

The recipient to get the secret key based on the identity uses the extract phase. The identity verification process occurs by comparing the identity sent in the form of ciphertext and the identity of the recipient. Fig. 4 shows that the extraction time at IBE-ECC-ECIES takes 0.6 s while the IBE-ECC-AES takes 0.9 s. The time stated in the extraction process is a combination of request time, extraction time, response time and other processing times. Other processing times in question are open files, read files, etc., which also require several milliseconds (ms).

TABLE V
COMPARISON OF IBE-ECC-ECIES AND IBE-ECC-AES AT THE DECRYPT STAGE

Elliptic Curve	IBE-ECC-ECIES (s)	IBE-ECC-AES (s)
P-192	0,206	0,043
P-224	0,210	0,045
P-256	0,212	0,047
P-384	0,220	0,049
P-521	0,224	0,051

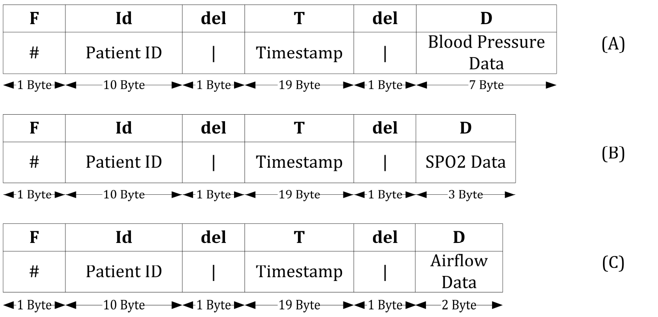
The next step is decrypt stage, the results of computational time comparisons between IBE-ECC-ECIES and IBE-ECC-AES will be shown in Table V. The computational time required by the two schemes to decrypt health data is relatively faster with the encryption of health data. Systems that use the IBE-ECC-ECIES algorithm are slower in the decryption of health data with a computation time of 0.2 s, this is compared to systems that use the IBE-ECC-AES scheme which requires 0.05 s to do the decryption process.

Based on the results of the comparison above it was found that the proposed IBE-ECC-ECIES algorithm has a relatively slower time in encrypt and decrypt stage, which is equal to 0.2 s. This is because the proposed method uses the ECIES algorithm that uses asymmetric and symmetric concepts so that there is an elliptic curve calculation that takes longer when compared to its comparison scheme, which only uses an encryption algorithm with the symmetric concept. But this is inversely proportional to the setup and extract stages that the proposed method is superior because both use the asymmetric concept with its public key generation algorithm.

B. Testing computing time on the communication channel

In testing this computational time implemented on the client and multiple server communications, four host OSs are used as a client and 3 as multiple servers. Each server will get data with different data lengths. The client will break the data into three parts as shown in Fig. 5.

Before the data is sent, each data will go through an encryption process using the public key of each server to form a ciphertext. The desired results from this test are information data obtained in accordance with the information data sent. There is a possibility that the data information is not the same as the one sent, this can be caused by an incompatible secret key or damage to the data received.



F	Id	del	T	del	D
#	Patient ID	|	Timestamp	|	Blood Pressure Data
1 Byte	10 Byte	1 Byte	19 Byte	1 Byte	7 Byte

(A)

F	Id	del	T	del	D
#	Patient ID	|	Timestamp	|	SPO2 Data
1 Byte	10 Byte	1 Byte	19 Byte	1 Byte	3 Byte

(B)

F	Id	del	T	del	D
#	Patient ID	|	Timestamp	|	Airflow Data
1 Byte	10 Byte	1 Byte	19 Byte	1 Byte	2 Byte

(C)

Fig. 5 The format of data to be sent to (A) Server 1 (B) Server 2 and (C) Server 3

The stage in the IBE scheme that will be observed is the total time between the encrypt and decrypt stages. In this test the proposed method will be compared with the comparison method as in the previous test, the difference lies in the observed parameters. Each scheme will be compared according to the amount of data sent. Testing is done by sending 10 data on the first shipment, then adding 10 data in the next shipment until the data sent to 150 data in one shipment.

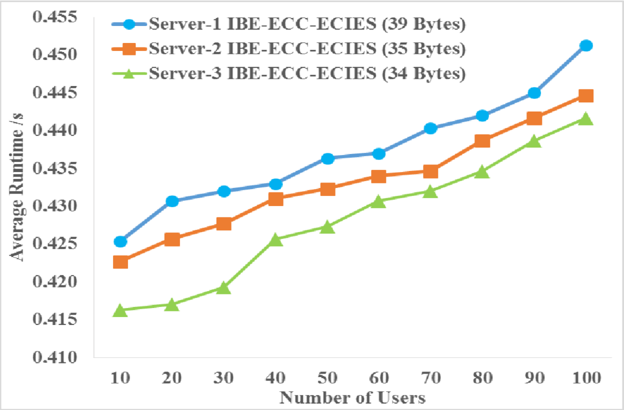


Fig. 6 Encrypt-Decrypt Computing Time on IBE-ECC-ECIES Scheme

The longer the data sent to the server, the total time needed at encrypt and decrypt stage will be longer. Fig. 6 shows that the IBE-ECC-ECIES scheme takes more than 0.4 seconds for each server. This is directly proportional to the amount of data sent. The more data that is sent, the computing time needed at encrypt and decrypt stage becomes longer. While for IBE-ECC-AES shown in Fig. 7.

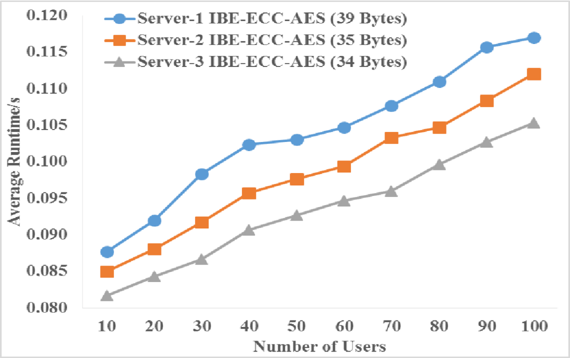


Fig. 7 Encrypt-Decrypt Computing Time on IBE-ECC-AES Scheme

Fig. 7 shows the total time needed by IBE-ECC-AES for the encrypt and decrypt stages of 0.1 seconds for each server. With the same amount of data and data length, the proposed scheme has up to 4 times slower time than the comparison scheme for the decrypt encrypt stage. The delay in computation time on the proposed scheme is relatively faster when compared to using the standard elliptic curve algorithm.

C. *Effect of Hardware Specifications on the proposed Method*

In this paper, we tested the effects of the hardware used in the proposed method (IBE-ECC-ECIES). The hardware used is the 2020M Intel Pentium CPU with 2.40 GHz. CPU Power on this device will be set to 50%, 70%, and 100%. CPU capability will be compared with the amount of data sent. The amount of data sent starts from 10 data to 150 data in one shipment.

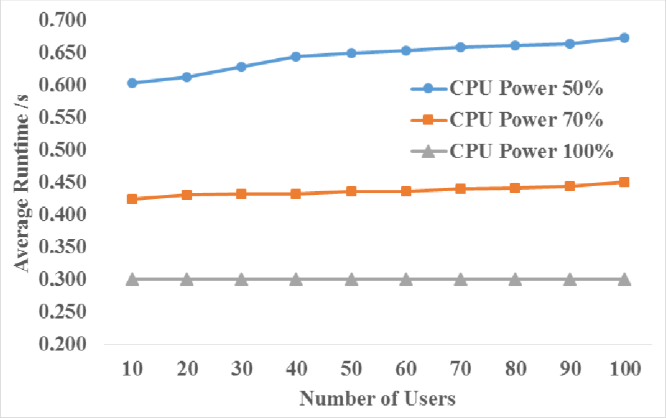


Fig. 8 Comparison of CPU Power to the proposed method

The stage in the IBE scheme that will be observed is the total time from encrypts and decrypts stage. Fig. 8 shows that when CPU Power is set to 50%, the total time needed to encrypt and decrypt ranges from more than 0.6 seconds for sending 10 data. The increase in the amount of data sent also affects the total time needed by the system, such as when sending 150 data, the time needed by the system ranges from 0.67 seconds. When the CPU capability is 50%, the time difference for adding data from 10 data to 150 data ranges from 0.07 seconds.

When CPU Power is set to 70%, the time needed by the system ranges from less than 0.43 seconds for sending 10 data, while sending 150 data takes more than 0.45 seconds. When the CPU capability is 70%, the time difference for adding data from 10 data to 150 data ranges from 0.02 seconds.

Whereas when CPU Power is set 100% then the time needed by the system to send 10 data to 150 data has almost the same time, which is around 0.3 seconds. So that to find out the difference in the time of adding data from 10 data to 150 data, it takes a range of milliseconds. The greater the capacity of the CPU that is used, the more time it takes to encrypt and decrypt the process, the more constant and unaffected by a large amount of data sent.

Based on differences in computing time when CPU power is changed to 50%, 70%, and 100%, it should be observed that sending data between 10 data with 100 data does not have distant computation time. So that it can be said that the encryption algorithm used is quite reliable.

D. *Testing Security of Proposed Method*

Man In The Middle Attack (MITM) is a type of attack that is very dangerous and can occur anywhere, both on websites, cellular phones, and traditional communication equipment such as correspondence. The aim of the MITM is intercepting communications confidential data such as sniffing. Sniffing is often referred to as a passive attack because the attacker does nothing but monitor the data passing. The greatest strength of MITM is not only in its sniffing abilities but also in the ability to intercept and change communication. So that MITM attacks can be called active attacks. In testing system security, there are two attacks, namely sniffing and chosen-plaintext attack.

1) *Sniffing:* The attacker acts to monitor data between two parties that communicate with each other. The Attacker will analyze data traffic using Wireshark. In this scenario, the attacker must be in the same network as the sender and receiver as in Fig. 9. This is so that the attacker is freer to observe the conversation between the sender and the recipient.

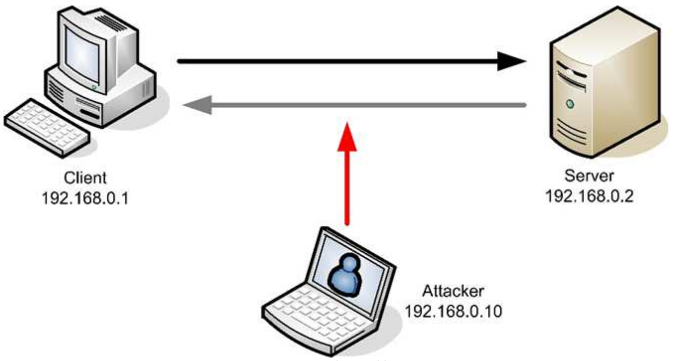
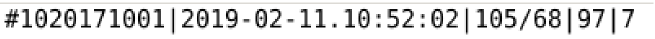


Fig. 9 Illustration of Sniffing Scenario

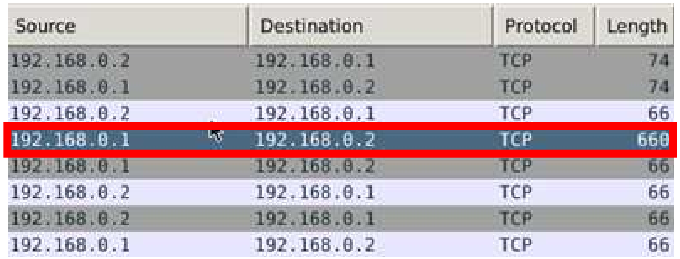
The client uses the IP Address 192.168.0.1 and the server uses the IP address 192.168.0.2. The attacker must be in the same network in order to see communication between the client and the server. The IP address used by the attacker is 192.168.0.10. The original health data used in this test is shown in Fig. 10.



```
#1020171001|2019-02-11.10:52:02|105/68|97|7
```

Fig 10. Original Data

The above health data will be encrypted using the IBE-ECC-ECIES scheme with the P-384 curve. Data that will be sent is not original health data but in the form of ciphertext. Fig. 11 shows the communication between the client and the server captured by the attacker using Wireshark.

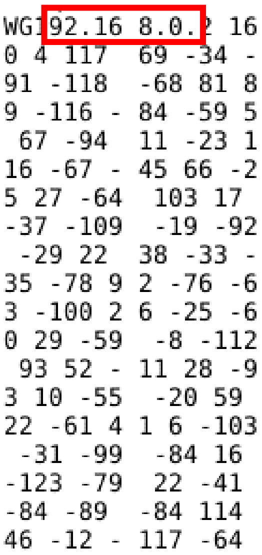


Source	Destination	Protocol	Length
192.168.0.2	192.168.0.1	TCP	74
192.168.0.1	192.168.0.2	TCP	74
192.168.0.2	192.168.0.1	TCP	66
192.168.0.1	192.168.0.2	TCP	660
192.168.0.1	192.168.0.2	TCP	66
192.168.0.2	192.168.0.1	TCP	66
192.168.0.2	192.168.0.1	TCP	66
192.168.0.1	192.168.0.2	TCP	66

Fig. 11 Communication Results captured by the attacker

Fig. 11 shows that the attacker can know the communication carried out by the client and the server. The red box in Fig. 11 explains that the sender's IP address is 192.168.0.1, which is the IP address of the client, and the recipient's IP address is 192.168.0.2, which is the IP address of the server. The protocol used is TCP with a data length of 660 bytes of which 594 bytes is ciphertext.

The Attacker can easily know health data that is sent directly without encryption. Fig. 12 shows that the attacker can only monitor data in the form of ciphertext from sniffing scenarios. The red box in Fig. 12 shows the IP address of the recipient, this is in accordance with the scheme used, namely identity-based encryption where the identity used is public so the attacker can find out the destination of the sent ciphertext. In this scenario, it can be seen the importance of a security algorithm on health data. An attacker can only monitor data in the form of ciphertext so that the original health data can be kept confidential.



```
WG192.16 8.0.2 16
0 4 117  69 -34 -
91 -118  -68 81 8
9 -116 - 84 -59 5
 67 -94  11 -23 1
16 -67 - 45 66 -2
5 27 -64  103 17
-37 -109  -19 -92
 -29 22  38 -33 -
35 -78 9 2 -76 -6
3 -100 2 6 -25 -6
0 29 -59  -8 -112
 93 52 - 11 28 -9
3 10 -55  -20 59
22 -61 4 1 6 -103
 -31 -99  -84 16
-123 -79  22 -41
-84 -89  -84 114
46 -12 - 117 -64
```

Fig. 12 Ciphertext obtained by the Attacker

2) *Chosen-Plaintext Attack:* The attacker acts as a fake sender using the IP spoofing method. IP Spoofing is a technique used to exploit security by changing the source of the IP Address. In this technique, the attacker does not get a message from the process that has been done whether successful or failed. The Attacker will send data with a fake IP Address, and the target computer will trust that the sender with a fake IP Address has sent the data. The illustration of the IP spoofing method is shown in Fig. 13.

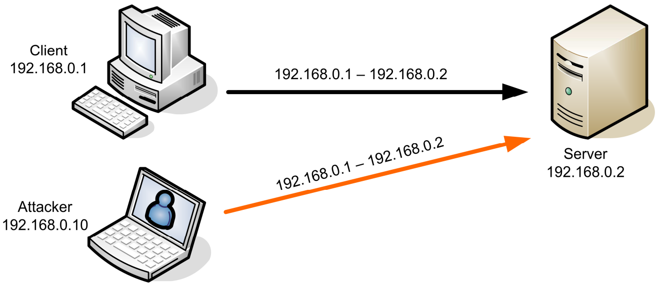
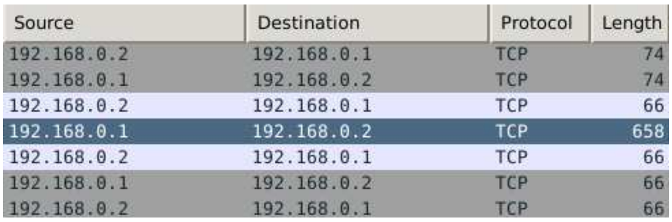


Fig. 13 Illustration IP Spoofing Method

It is assumed that the fake sender knows the identity and encryption algorithm used but does not know the elliptic curve used. The algorithm used is ECC for synchronization between sender and receiver. Data traffic between the sender and receiver in the Wireshark is shown in Fig. 14.



Source	Destination	Protocol	Length
192.168.0.2	192.168.0.1	TCP	74
192.168.0.1	192.168.0.2	TCP	74
192.168.0.2	192.168.0.1	TCP	66
192.168.0.1	192.168.0.2	TCP	658
192.168.0.2	192.168.0.1	TCP	66
192.168.0.1	192.168.0.2	TCP	66
192.168.0.2	192.168.0.1	TCP	66

Fig. 14 Traffic on Wireshark when an Attacker becomes a Fake Sender

Communication occurs between senders who have an IP address of 192.168.0.1 with a recipient who has an IP address 192.168.0.2. The attacker changes the IP address and falsifies it to 192.168.0.1 as the original sender's IP address. So the recipient considers the attacker's IP address to be the IP address of the original host not from outside the network.

The Attacker will encrypt using ECIES with the P-256 curve. Obtained results that the ciphertext sent by the attacker can be sent to the server but cannot be decrypted. This is because there are differences in the key pairs used. The health data generated by the attacker will be different from the ciphertext from the original sender.

IV. Conclusions

The security method proposed in this study uses the IBE scheme, which is integrated with ECC and ECIES. The ECC algorithm is used as a key pair generator, while the ECIES algorithm is used in the encryption and decryption process because it is hybrid. This research can be used in all aspects of health data, banking data, or other important data that requires high security. The advantage of this method is that the user's identity is embedded in the key pair so that it produces a unique key pair for each user. User identity is an advantage of the IBE scheme, which can be any arbitrary string that in this study we use IP addresses. This scheme we compare with other IBE hybrid schemes namely IBE-ECC-AES. In the setup and extract stages in the IBE scheme, the proposed method is superior with a computing time of 0.6 seconds. Whereas in encrypt and decrypt stage, the proposed method is 4 times slower than the comparison scheme, which is 0.2 seconds. In addition, the influence of hardware also affects the performance of the proposed method, the greater the ability of the CPU to be used, the more time it takes for encrypt and decrypt process to be constant and unaffected by a large amount of data sent. This is because the total computation time required by the system to send 10 to 150 data per shipment is worth fixed in 0.3 seconds for 100% CPU capability. The advantage of this method is that users can generate secret keys using the type of curve agreed upon by both parties; this minimizes the occurrence of key exchanges on the communication lines. In future work, this security method will be implemented on real hardware.

References

[1] A. Shamir, "Identity-based Cryptosystems and Signature Schemes," *Adv. Cryptol. - CRYPT0 '84, LNCS 196*, pp. 47–53, 1985.

[2] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Proc. Crypto 2001*, vol. 2139, pp. 213–229, 2001.

[3] L. B. Oliveira *et al.*, "TinyPBC : Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Comput. Commun.*, vol. 34, no. 3, pp. 485–493, 2011.

[4] L. B. Oliveira, R. Dahab, L. Julio, F. Daguano, and A. A. F. Loureiro, "Identity-Based Encryption for Sensor Networks," in *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, 2007.

[5] A. Sudarsono, M. Yuliana, and H. A. Darwito, "A secure data sharing using identity-based encryption scheme for e-healthcare system," in *International Conference on Science in Information Technology (ICSITech)*, 2018, pp. 429–434.

[6] G. Ateniese and P. Gasti, "Universally anonymous IBE based on the quadratic residuosity assumption," *Fischlin M. Top. Cryptol. – CT-RSA 2009*, vol. 5473, pp. 32–47, 2009.

[7] D. Boneh and C. Gentry, "Space-Efficient Identity Based Encryption Without Pairings," in *48th Annual IEEE Symposium on Foundations of Computer Science*, 2007, pp. 647–657.

[8] C. C. Tan, H. Wang, and S. Zhong, "IBE-Lite : A Lightweight Identity-Based Cryptography for Body Sensor Networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 926–932, 2009.

[9] Himanshu, P. Yadav, and A. Bisla, "Identity-based Encryption," Malaviya National Institute of Technology Jaipur, 2015.

[10] National Institute of Standards and Technology (NIST), *Recommendation for Key Management - Part 1 : General*, vol. SP 800-57. 2007.

[11] D. N. Purnamasari, A. Sudarsono, and P. Kristalina, "Secure Data Sharing Scheme using Identity-based Encryption for e-Health Record," in *2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, 2019, pp. 60–65.

[12] S. Al-Alak, Z. Ahmed, A. Abdullah, and S. Subramiam, "AES and ECC mixed for ZigBee wireless sensor security," *World Acad. Sci. Eng. Technol.*, vol. 81, no. 9, pp. 535–539, 2011.

[13] S. Sharma and V. Chopra, "Analysis of AES Encryption with ECC," *Int. Interdiscip. Conf. Eng. Sci. Manag.*, no. December, pp. 195–201, 2016.

[14] B. Ji, L. Wang, and Q. Yang, "New Version of AES-ECC Encryption System Based on FPGA in WSNs," *Journal of Software Engineering*, vol. 9, no. 1. pp. 87–95, 2014.

[15] S. A. Abbas and A. A. B. Maryoosh, "Data Security for Cloud Computing based on Elliptic Curve Integrated Encryption Scheme (ECIES) and Modified Identity-based Cryptography (MIBC)," *Int. J. Appl. Inf. Syst.*, vol. 10, no. 6, pp. 7–13, 2016.

[16] K.-L. Tsai, F.-Y. Leu, T.-H. Wu, S.-S. Chiou, Y. Liu, and H.-Y. Liu, "A Secure ECC-based Electronic Medical Record System," *J. Internet Serv. Inf. Secur.*, vol. 4, no. 1, pp. 47–57, 2014.

[17] M. Yuliana, G. Awaludinsyah, A. Pratiarso, and A. Sudarsono, "Design and Implementation of a Secured Personal Identity-based ECC and ECDSA : an Inpatient System," *Eur. Sci. J.*, vol. 11, no. 21, pp. 473–483, 2015.

[18] M. Rinaldi, *Kriptografi*. Institut Teknologi Bandung, 2006.

[19] C. Research, "Standards for efficient cryptography - SEC 2 : Recommended Elliptic Curve Domain Parameters," 2000.

[20] A. Sebastian, "Implementasi dan perbandingan performa algoritma hash SHA-1, SHA-256, dan SHA-512." Institut Teknologi Bandung, pp. 1–18, 2007.

[21] L. Zachariah, "Analysis and comparison of ECC & ECIES using IBE for securing patient's privacy," vol. 2, no. 6, pp. 43–47, 2012.