SSR Method for Private Medical Data in HSN using Light Weight Encoding Method

Radhika Rani Chintala, Somu Venkateswarlu, M. Sri Lakshmi

Abstract— A human network (HSN) is a system of small, commercially available and relatively weak sensors implanted on patient's body for reading various parameters from patient body viz., heart rate, pulse rate etc.. As these readings are patient specific they must be properly encrypted and securely stored. The present paper proposes the development o of a Secure Storage and Retrieval (SSR) Method, is a method which is lightweight identity-based suitable for small sensors used in a HSN. Algorithms that are discussed are derived from SSR Method which balances privacy & security with availability. The experiments are conducted using the sensors which are cheaply available in the commercial market.

Keywords: Human sensor network, SSR method, privacy and security.

INTRODUCTION

The utilization of remote sensors for human medical services observing makes better approaches for giving high quality care for the patient health. A different cluster of particular sensors can be sent to screen for example an at-hazard patient noticeably having a past filled with heart attacks. By persistently gathering a patient's health information in a subtle way, the implanted sensors can give specialists with extra data to analyze the patient health in a better way. A human sensor network (HSN), is a critical part in this observing plan. HSN comprises of sensors put specifically implanted on the body of a patient or stitched on patient's garments, that goes along with that patient by gathering his/her health information [1].

The HSN must be continuously switched on for ceaselessly collecting the patient data and will be making extra security and protection requests. A patient appropriately need to restrict the get to and extent of the gathered information to various individuals. It was proposed that the patient can control the access to his medical data concurring to the day, month, time, date, and the individual’s identity who wishes to access the medical record. For instance, a patient might need to restrict a physical therapist's entrance to HSN information gathered on a specific date at 2 pm only. Practically speaking, more stringent get to conditions may be embraced, for example, access is limited to certain areas where the information was gathered.

Here in the present paper, HSN conveyed for therapeutic monitoring has been focused. The information gathered by the HSN, as is occupying large storage space, may be stored on a server computer or can be stored in a cloud. There are several agencies which are offering the cloud services at a reasonable cheaper rate. The stored data of the patient can be controlled using powerful encrypting techniques. The decrypting keys can be made accessible to the doctor by the patient. The cloud service provider will release the information of the patient on a specific day, date and time, to the concerned doctor at the request of the patient. It means the decrypting keys for the data collected from the patient on a specific day and date at 2pm is different from the data collected from the same patient on the same day and date at 2.30pm are different. Thusly, the patient can relegate the proper unscrambling key to various individuals to restrain access to data.

1.1 Symmetric Key Encryption (SKE)

In SKE method, the key that is used for encoding the data and decoding the data is same. In this way, for a persistent wearing HSN that screens him/her continuously on a 24X365 basis and just needs the essential specialist to get to that data, should store 24 X365X2 = 17, 520 symmetric keys in the HSN, if an alternate key is utilized for every half an hour. On the off chance that the patient wishes to control access to various people like other medical specialists, care takers in the hospital etc., then more keys should be appointed to the HSN.

1.1.1 Drawback with SKE

Since the solitary key is used for both encoding and decoding the data at the Cloud, an issue may happen once the HSN or the solitary sensor of HSN is stolen. Then the intruder will is capable enough to decode the entire information. To overcome this drawback, it is preferable to use separate keys for encoding and decoding. Hence for a HSN using 24 X365X2 = 17, 520 keys are required, for the above case, for encoding and same no. of keys are required for decoding the data. Or it is also possible to generate separate keys for separate sensors. Thus if 25 sensors are used then 25 X 24 X365X2 = 4,38,000 keys have to be generated for the above case. Whatever may be the solution used by, the key management is going to be very complicated process.

Revised Version Manuscript Received on March 08, 2019.

Radhika Rani Chintala, Department of CSE, KLEF, Vaddeswaram, Guntur, Andhra Pradesh, India

Somu Venkateswarlu, Department of CSE, KLEF, Vaddeswaram, Guntur, Andhra Pradesh, India

M. Sri Lakshmi, Department of CSE, KLEF, Vaddeswaram, Guntur, Andhra Pradesh, India

Consequently, numerous traditional protocols such as SSL utilize symmetric keys in order to encrypt the data, yet utilize public keys in order to encode the symmetric key before transmission.

1.2 Conventional Public Key Encryption(PKE)

In traditional PKE like RSA algorithm, 2 separate keys are utilized, encoding key and decoding key. Encoding key is saved on HSN, and decoding key is saved securely. If the intruder, somehow, came to know the encoding key, he can only be able to know the encoding process and cannot decode the data, as the decoding key is securely saved in the cloud. However the total encoding process becomes vulnerable [2]. This represents an issue when brief access to the HSN information is required.

1.3 Identity Based Encryption (IBE)

IBE is a type of an asymmetric cryptography such as RSA. Distinct from RSA in which both private key and public key are to be created simultaneously, IBE permits public key to be produced from a discretionary string [4]. The subsequent private key will be created independently later on. For example, a patient might ask the Cloud Manager to discharge keys to any specialist doctor (ER). A new public key will be generated by the patient's HSN each day, by utilizing the string str = {day | time | date | ER}. The Cloud Manager need not make the comparing private key. When a specialist doctor (ER) needs to acquire information for a specific date at 2 pm, he has to first validate himself to Cloud Manager. The Cloud Manager will then generate the decoding key utilizing the identical string str = {day | time | date | ER}. This can just decode the information collected from the patient on that date, day & time.

Because of the asymmetric property of the IBE, the intruder who got the access to the encoding key, can't decode the data as the decoding key, which is based on the structure, will be generated by the Cloud Manager only at the permission of the patient.

In the present paper protocols are designed keeping in view of IBE that give security & privacy assurances while permitting adaptable access to saved information. IBE is effectively examined and generally used in cryptography research, traditional IBE can't be proficiently executed on HSN sensors despite its demand. We created SSR Method, which is a lightweight IBE that is appropriate for HSN. By implementing the SSR Method on the sensors that are commercially accessible, reasonable performance improvement may be observed.

2 SSR METHOD SOLUTION

SSR Method is based on ECC- Elliptic Curve Cryptography, which is a public key encryption technique appropriate for HSN [3,4].

To setup an ECC, first we have to identify a private key x, and the public parameters such as (y, P, p, q, h(.)). The following Table 1 demonstrates the extent of mentioned parameters in bytes/bits.

Table 1: Size of Basic ECC Primitives

Private Key	X	160 bits
Public Parameters	(y, P, p, q, h(.))	1120 bits

y & P are 320 bits long,
p,q, & h(.) are 160 bits.

2.1 SSR Method

From fundamental primitives of ECC, the accompanying SSR Method primitives can be inferred. The instinct behind utilizing SSR Method is to allow the sensor to freely create a public key on-the-fly by utilizing the discretionary string. For instance, a sensor that gathers EMG readings on February 2nd Tuesday at 2pm first will make a string str = (Tuesday | 2pm | 01022017 | EMG).By utilizing this string, a public key ystr is generated by the sensor in order to encode the information and stores it in the cloud. Corresponding private key need not be made. Actually, the sensor can't generate the private key that is expected to decode the message.

If the Cloud Manager want to give this data to a medical doctor, he will infer the comparing private key xstr by utilizing the similar string str = (Tuesday | 2pm | 01022017 | EMG) that permits the specialist doctor to decode the data. As the string used to depict the event should be same, the key management is simplified. Our primitives are as per the following:

2.1.1 Setup

The selected patient chooses E which is an elliptic curve over GF(p),such that p is a large prime number. P is also additionally indicated as the base point of E and order of P is q, which is likewise another large prime number. The selected patient creates n private keys viz., $x_1, \dots, x_n \in$ GF(q)for creating the master private key.

$$X = (x_1, \dots, x_n). \qquad \text{Eq.(1)}$$

The n public keys are then generated to make up the master public key.

$$Y = (y_1, \dots, y_n) \qquad \text{Eq.(2)}$$

where $y_i = x_i P$, $1 \leq i < n$. Finally, the patient selects a collision resistant one-way hash function h: $\{0, 1\}* \rightarrow \{0, 1\}n$. The parameters

$$< Y, P, \ p, \ q, \ h(.) > \qquad \text{Eq.(3)}$$

are released as the system public parameters.

2.1.1 Keygen

From the public key that is generated by using a string str, the corresponding secret key x_{str} is derived by the patient by executing Keygen(str)= x_{str}

$$x_{\text{str}} = \sum_{i=1}^{n} h_i(\text{str}) x_i \qquad \text{Eq.(4)}$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

where h_i(str) is the i^{th} bit of h(str).

2.1.2 Encode

To encode an information packet 'm' utilizing a public key that is computed using string str, sensor does Encode(m, str) for finding the encoded text *c*. Algorithm1 demonstrates the procedure. In Algorithm1, the line numbers 1 & 2 require to be executed only once to find public key y_{str}.

Algorithm 1: Encode(m, str)

1. Obtain the string str based upon the agreed syntax.
2. Public key y_{str} is generated using $y_{str} = \sum_{i=1}^{n} hi(str).yi$
3. Execute EccEncode(m,str) to obtain cipher text c.

2.1.3 Decode

The doctor runs Decode(c, x_{str}) to find the actual information packet '*m'* that has been encoded by utilizing a private key obtained from str. The method to follow is given through Algorithm2.

Algorithm 2: Decode(c,str)

1. Requests permission from cloud manager to obtain data described by str.
2. Cloud Manager executes Keygen(str) to derive $x_{str.}$
3. Doctor obtains m by executing EccDecode(c,x_{str}).

2.2 HSN Security Protocols

Protocols developed using SSR Method have been explained here. There are totally four phases.

Phase-1: (Initialization phase) - In this phase patient uses HSN for first time.

Phase-2: (Data collection phase) - This phase explains how encoding is done on the collected information by the sensor.

Phase-3: (Transfer of data phase) - This phase explains the process of transferring the data to cloud by HSN.

Phase-4: (Query phase) - This phase occurs when the specialist doctor needs to get the data back from the cloud.

str describes the agreed upon structure that is used to define public key. The data collected at different intervals will be as follows: (Tuesday | 2 p.m. | 01022017 | EMG) and (Tuesday | 2.30 p.m. | 01022017 | EMG). The cloud will never delete the data and is assumed to be trusted having enough security mechanism for protecting the data.

2.2.1 Phase-1 (Initialization phase)

Patient has to execute the setup procedure and then only he will be getting master private key $X = (x_1, . . . , x_n)$ & public parameters [Y,P, p, q, h(.)]. These parameters have to be loaded into all sensors that are available in HSN. Then the master private key has to be registered in the cloud.

2.2.2 Phase-2 (Data collection phase)

Sensor do collect the information d as per the structure defined in str. This information is encoded as per the procedure described in Algorithm 3.

Algorithm 3: Encoding of the data by the sensor

1. Derive the string str, and generate a random number n
2. Compute m_1 = (flag| n), where flag is a known bit string
3. Compute m_2 = (d | n)
4. Compute c_1 = Encode(str, m_1)
5. Compute c_2 = Encode(str, m_2)

The cipher text c1 & c2 is then stored in cloud.

2.2.3 Phase-3(Transfer of data phase)

The information periodically collected from all the sensors in HSN is transferred to the cloud.

2.2.4 Phase-4 (Query phase)

The specialist doctor will contact the cloud manager for getting the access to the information stored in the cloud. The procedure is explained in Algorithm 4.

Algorithm 4: Doctor querying for data

```
for every ($c_1^i$, $c_2^i$), i ϵ k for patient
{
     Cloud sends $c_1^i$ to specialist doctor
     Specialist doctor executes Decode($c_1^i$, str)
     If initial bits of result matches the flag then
1.Specialist doctor requests for the corresponding $c_2^i$ from the cloud
2. Specialist doctor executes Decode($c_2^i$, str) and checks if n==$c_1^i$
3. Specialist doctor accepts d if both are correct
     end if
}
```

The information obtained from the patient is fully encoded before getting stored in the cloud. The doctor has to decode each (c_1, c_2) separately for knowing the patient information. Here size of c_1 is much smaller than size of c_2. This is the reason why the cloud returns c_1 first to the doctor.

c_2 implants a similar arbitrary number n in c_1 & c_2 as well. The specialist will just acknowledge the information in c_2 to be legitimate just as both arbitrary numbers are same. This arbitrary n is known just to the sensor scrambling the information.

2.3 Query Improvements

As the amount of data (proportional to number of sensors and duration of time interval) increases, more time is consumed by the doctor while accessing the information from the cloud. The cipher text (c_1,c_2) can't be read by the cloud and hence cloud cant index the content (c_1,c_2). This makes the doctor to consume more time while searching for a patient's data. The doctor has to go through every (c_1,c_2) before gaining the accessing the data of the patient under observation.

Performance of the search can be enhanced by slightly changing the encoding methodology which summarizes several data packs together. For instance, consider that the encoding is to be done on the data collected on the same day and date, but at different times. Conditions are more or less the same for both cases (same day and date), let there be a hint created like:

η = Encrypt(m, str) where

m = (flag|n|i_1^1|i_1^2) Eq.(5)

and str = {*day*, date, time, ER}. Here $i_1^1$and i_1^2refer to the indices pointing to$c_1^1$and c_1^2.

The specialist doctor can obtain an additional key for date from the cloud manager in order to decode the hint that enhances the performance, because doctor is asking for c_1 only from the hint that doctor decodes. The doctor needs exact key x_{str} to decode $c_1$that is not known to him yet from the hints. Hence the procedure described is secured one.

3 ANALYSIS OF SSR METHOD & ITS EVALUATION

3.1 Security

Security of the protocols proposed is analyzed in this section. Encoding& decoding is done utilizing the keys xstr and ystr based on the structure str. xstr & ystr don't break the discrete logarithm ppt. Wherever, given y = xP, it's not possible to calculate x given y and P, because both are made by adding points on a same curve.

Eavesdropping Attack: While the HSN is transferring the information packets to the cloud, the attacker may gain the control on the data. But the data that is transmitting towards the cloud is encoded and hence attacker learns nothing.

Tracking Attack: In this attack, the attacker attacks the patient’s privacy by monitoring multiple transmissions between HSN and the cloud. At the maximum the attacker may be able to that the information is generated by the same patient. As each cipher text is being encoded using a separate prime number, even the identical information generated by the same patient is also going to be encoded by differently structured str. Hence attacker learns nothing.

Compromised Sensor: Even though any sensor is compromised within the HSN, the attacker learns nothing out of it as each sensor is having its own set of parameters.

Matching Attack: The attacker initiates this attack by generating several public keys by using various structures str. The attacker then encodes all possible combinations utilizing various public keys to find out the matched cipher text. However, c1 and c2 have a random number n created by sensor. As the attacker, can't estimate the value of n, the matching attack fails.

Honest-But-Snooping Cloud: In the proposed protocol, entire information stored in the cloud is encoded using the structure str. The private keys are not stored there. Hence even if any attacker enters the cloud (may be through advertisement channel) he will be able to read the data. The information of the patient taken from various sensors in the HSN has to be stored in a different storage location also as the cloud may delete the data unintentionally sometimes.

Complexity Analysis: When n public keys Y = (y1, . . . , yn) have been considered then SSR Method will generate a public key ystr, using the structure str having a time complexity O(n). Time required to encode using ystr is O(1). In the same fashion time complexity for decoding is O(n) and for generating the private key xstr is O(1).

The protocols used by the authors in this proposal are based on asymmetric key encryption method. This method makes the system to store only the public keys in the sensors. This makes the schemes, proposed, secure against all the intruders in the HSN. But, the protocols proposed are at risk of attack if incase there are O(n) colluding attackers each with a separate and exclusive private key xstr. The attackers attacking the HSN together, can make use of their exclusive private keys in order to find the master private key X.

3.2 Limitations

A restriction of the proposed system is that individuals can just only discharge n private keys x1str,..., xnstr. Once if greater than n private keys are discharged, the master private key X (as given in 1) is at risk of compromise. While traditional IBE programs [3] don't use this restriction, those systems may not be applied on hardware of HSN. If sufficient no of private keys are released, then SSR Method also works more or less same as traditional IBE so long as a restricted quantity of private keys are released. This is believed as a sensible tradeoff because of following reasons.

- As long as the cloud manager is trust worthy, the private keys will not be shared with other user viz.. doctors.
- Every care has to be taken that n (number of private keys) should be large enough so as to ensure, the patient will have enough keys to encrypt the data securely.

Ultimately, HSN can be rekeyed by generating another set of n private keys, whilst the master private key, and keep the newest information in the sensors. A simple laptop can be used for rekeying by the HSN, and this data then kept in HSN and Cloud. To minimize the rekeying frequency, the cloud can be ordered to advice the patient when to rekey his HSN after a particular no of private keys are released. This avoids unwanted rekeying.

3.3 Performance

The performance of the proposed protocols can be evaluated by repeatedly conducting experiments on commercially available sensors.

Figure1 reveals the quantity of storage required for various encryption schemes. The proposed protocols are using the advantages of both symmetric and Asymmetric key encryption (RSA) schemes, even while using the storage space also, i.e. exhibiting the advantages of the asymmetric private key and making use of little more storage than normally demanded by SKE scheme.

4 EXPERIMENTAL RESULTS

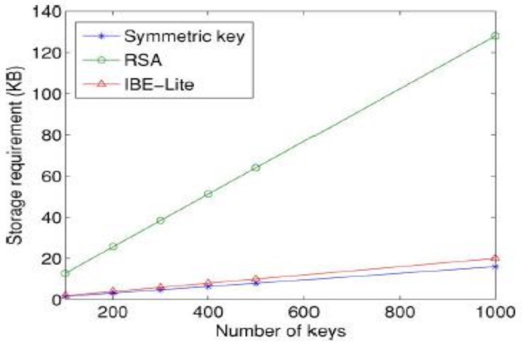

Figure1. Storage quantity required to store n keys for various encryption methods.

The below figure, Figure 2, clearly explains the overhead generated by various encryption schemes while transmitting the data. More time is required by SSR Method in order to derive a private key y_{str} by using string *str and n* no. of public keys viz.. y_1,..., y_n. This is the overhead of SSR Method. Public keys are computed in advance for storing them in sensors by symmetric key and RSA.

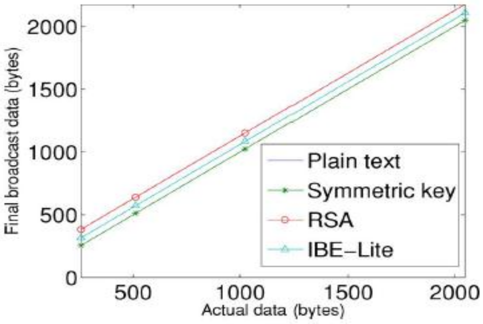

Figure2. Overhead of data transmission for various encryption schemes(in bytes)

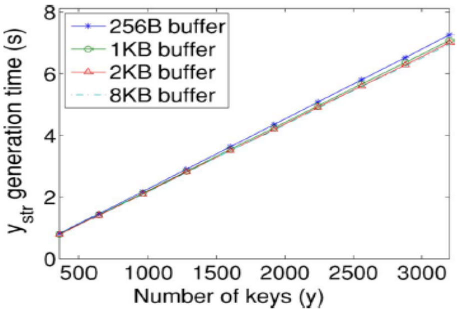

Figure 3. Time required for deriving one y_{str} with different n no. of public keys, y_1, . . . , y_n.

SSR Method do ask for extra time to generate the additional keys but will not use extra storage space as does the RSA. This property is exhibited through Figure 3.

Simulation has been done to evaluate the search improvements. The simulations performed have paid attention on the no. of messages exchanged between the cloud and the doctor who is using supposedly an advanced computer. Sensors will be taking the data from the patient body randomly for encryption, using the tuple having time period as one of its component.

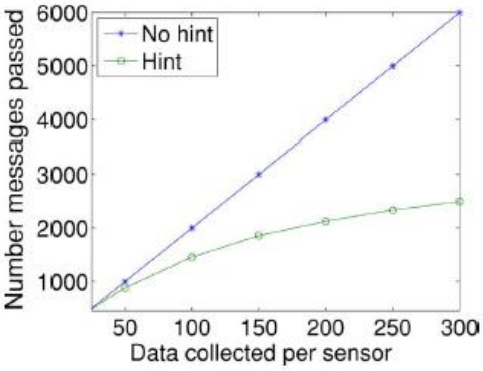

Figure 4. Number of passed messages.

The above figure, Fig. 4, shows the results for about 300 instances and also exhibits the improvement when hints are used.

5 PREVIOUS WORK

The advantage in using HSNs is in using relatively cheap and implantable sensors on the body of the patient, in order to monitor various health related parameters [5,6]. For the same purpose number of models have been proposed by researcher and they have even developed the working models too [7,8]. This present proposal though focusing on the same issue, focuses much on providing extra bit of security to the data being generated by the sensors.

A new type of asymmetric key encryption [AKE] method is IBE [4,9]. Zhang et al. [10] have produced algorithms using this IBE. But sensors used in those algorithms are relative larger ones. Whereas in the proposed SSR Method the sensors used are implantable and smaller ones. As a result the proposed algorithm does not depend on bilinear pairings like Weil or Tate pairings in primitives. Authors Wang et.al. [11,12,13] have proposed practically useful encryption algorithms. The problem with them is they paid the attention only on traditional algorithms which don't support the IBE characteristics listed in present paper.

Authors Mont et al. [14] have implemented IBE for a secured data transmission between the sensors. Authors Malasri and Wang [15] have used HSN for exchanging the security keys between the server and the device sensor used for the patient. The present algorithm proposed through this paper has paid the attention on using IBE on the small rather weak sensors and improving the performance of the system. The results given through this paper have been taken from the kit that is practically used.

Authors viz., Bao et al. [16] have used symmetric keys for providing security to the data generated by the HSN. The advantage with these keys is they use very small memory space per key and smaller encrypted text will be generated. Like RSA or IBE-Lite these will not have asymmetric property. The present algorithm performs faster than previous kits using sensors and other required hardware. Authors Baoet al. [17] in their subsequent paper have used continuously varying heart rate as a means for generating encrypting keys. Tan et al. [18] also consider usage of IBE over HSN. It suffered from the drawback of slow query performance while exploring large encrypted text. Radhika Rani Chintala et al. [19] also consider usage of Lightweight Encryption Algorithms for Wireless Body Area Networks.

5 CONCLUSION

A human sensor network (HSN) which user implantable sensors on the patient body for collecting various sensitive data from the concerned has been developed in the present paper that has used the SSR Method. The data generated by the sensors is highly encrypted for providing the privacy and security.

Trusted Cloud has been used for storing the generated data. The encryption mechanism has used a tuple for encoding where day, data and time are the parameters. Continuously varying time provides better security. Performance evaluation using experiments performed using the sensors which are commercially available in the market have shown better results.

6 REFERENCES

1. Radhika Rani Chintala, Narasinga Rao M R and Somu Venkateswarlu, "A Review on Security issues in Human Sensor Networks for Healthcare Applications", International Journal of Engineering & Technology, vol. 7, No. 2.32, 2018, 269-274.
2. Radhika Rani Chintala, Narasinga Rao M R and Somu Venkateswarlu, "Design of a Secure System for reading patient's data using Medical Sensor Networks", JCPS, vol. 10, No. 1, 2016, 673-679.
3. Boneh D and Franklin M, Identity-based encryption from the Weil pairing, Proc. CRYPTO, 2001, 213–229.
4. Lo B and Yang G.Z, Key technical challenges and current implementations of body sensor networks, Proc. Body Sensor Networks (HSN), 2005, 1–5.
5. Chiu C. Tan, Haodong Wang, Sheng Zhong and Qun Li, IBE-Lite: A Lightweight Identity-Based Cryptography for Body Sensor Networks, IEEE Transactions, VOL. 13, NO. 6, 2009, 926 - 932".
6. Radhika Rani Chintala, Narasinga Rao M R and Somu Venkateswarlu, " Implementation & Performance Analysis of Security in Human Sensor Networks", IJPAM, vol. 116, No. 5, 2017, 193-198.
7. Zhong L, Sinclair M and Bittner R, A phone-centered body sensor network platform: Cost, energy efficiency and user interface, Proc.HSN, 2006, 179–182.
8. Malan D, Fulford-Jones T, Welsh M and Moulton S, Codeblue: Anad hoc sensor network infrastructure for emergency medical care, Proc. HSN, London, U.K., 2004.
9. Cocks C, An identity based encryption scheme based on quadratic residues, Proc. LNCS, vol. 2260, 2001, 360–363.
10. Zhang Y, Liu W, Lou W and Fang Y, Location-based compromise tolerant security mechanisms for wireless sensor networks, Proc. IEEEJ. Sel. Areas Comm., 2006, 247–260.Bao S.D, Zhang Y.T and Shen L.F, A new symmetric crypto system of body area sensor networks for telemedicine, Proc. Conf. Jpn. Soc. Med. Electron. Biol. Eng., 2005, 654.
11. Wang H and Li Q, Efficient implementation of public key crypto systems on mote sensors (short paper), Int. Conf. Inf. Comm. Security (ICICS), Raleigh, NC, 2006.
12. Wang H, Sheng B and Li Q, Elliptic curve cryptography based access control in sensor networks, Int. J. Security Netw., vol. 1, no.3/4, 2006, 127–137.
13. Wang H, Sheng B Tan C.C, and Li Q, Comparing symmetric-key and public-key schemes in sensor networks, Proc. IEEE ICDCS, 2008, 11–18.
14. Mont M, Bramhall P and Harrison K, A flexible role-based secure messaging service: Exploiting IBE technology for privacy in health care, Proc. Int. Workshop Database Expert Syst. Appl., 2003, 432–437.
15. Malasri K and Wang L, Addressing security in medical sensor networks, Proc. Health Net, 2007, 7–12.
16. Bao S.D, Zhang Y.T and Shen L.F, A new symmetric crypto system of Body Area Sensor Networks for Telemedicine, Proc. Conf. Jpn. Soc. Med. Electron. Biol. Eng., 2005, 654.
17. Bao S.D, Zhang Y.T and Shen L.F, Physiological signal based entity authentication for body area sensor networks and mobile health care systems, Proc. IEEE Eng. Med. Biol., 2005, 2455–2458.Koblitz N, Elliptic curve cryptosystems, Math. Comput, vol. 48, 1987, 203–209.
18. Tan C.C, Wang H, Zhong S and Li Q, Body Sensor Network security: An Identity-Based Cryptography approach, ACM Conf. Wireless Security(WiSec), 2008, 148-153.
19. Radhika Rani Chintala, Lakku Sai Jagan, Ch. Lakshmi Harika, V. V.Durga Ravali Amara, "Lightweight Encryption Algorithms for Wireless Body Area Networks", International Journal of Engineering & Technology, vol. 7, No. 2.20, 2018, 64-66.