

IMPLEMENTATION & PERFORMANCE ANALYSIS OF SECURITY IN HUMAN SENSOR NETWORKS

Radhika Rani Chintala¹, Narasinga Rao M R², Somu Venkateswarlu³

^{1,2,3}Department of CSE, KL University, Andhra Pradesh, India

¹radhikarani_cse@kluniversity.in

Abstract: The importance E-health care is increasing day by day very rapidly. In this system the parameters of a patient like BP, Pulse Rate, temperature, heart rate etc., are monitored by the doctors. For this to happen there should be wireless connectivity between biosensors implanted on patient body and storage space like cloud. Wireless Sensor Networks (WSN) plays an important role in this system. The data that is collected from a patient should be securely protected from outside attacks. For this powerful encoding/ decoding techniques like the one based on hash-chain and a special signature technique useful for tiny sensor nodes. The real time behavior of the proposed algorithm has been implemented and the experimental have been analyzed.

Keywords: Medical diagnosis, security, performance analysis of HSN.

1. Introduction

Now a day as the popularity of Human Sensor Networks (HSN) is increasing day by day[1]. Hence more research is going on in developing different portable & implantable biosensor devices. The wireless technology is used as a bridge, in filling gap for establishing communication link between sensors and storage devices. [1]. As a result the parameters of patient Personal Health Data (PHD) can be read and monitored by the doctors remotely. The medical data is shared by, for further analysis, healthcare staff, scientists, government organizations, insurance agencies, as well as patients relatives too. In this way the effectiveness of medicinal services can be expanded significantly.

The figure Fig. 1 shows simple HSN architecture [1]. Larger the size of the HSN, more the number of Patient Area Networks (PAN) covered by HSN. All the biosensors can be connected to a controller system.

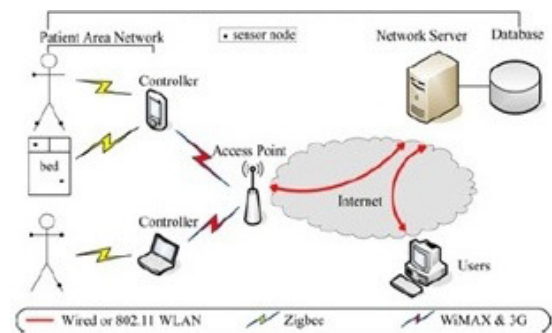


Figure 1. Architecture of HSN

The PHD, private information of a particular patient, play a crucial and viral role in diagnosing him and for further treatment to patient. Hence it is very important to secure the data stored in the server/controller. This gives medical security to the PHD.

2. Limitations of the HSN

- The biosensor nodes implanted in the patient, forming the HSN, have limited resources viz., memory capacity, processing speed and processor's processing capacity and power back up.
- Outside attackers may read and change the medical data of the patient. Even internal attackers too can access and misuse the data.

Keeping all the above safest and secure HSN has been proposed in [1]. These Human Sensor Networks used in medical field are different from

other such networks in the aspects like Data Transfer Rate, Mobility and Efficiency.

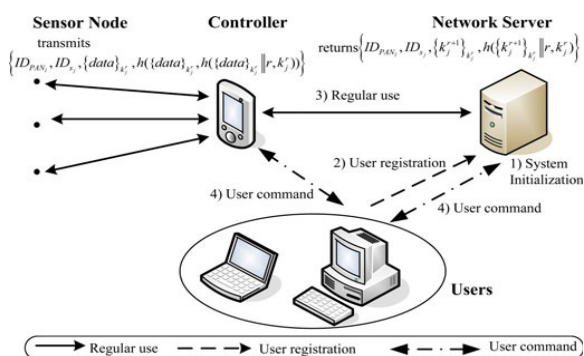


Figure 2. Security information flow in the proposed system.

3. Structure of the System Proposed

The proposed system has been developed through four phases. The Phase-1: Initialization of the system: For each PAN, viz., PAN_i , the server securely distributes the unique primary key k_{0j} with each biosensor, say s_j , and stores the value $(IDPAN_i, ID_{sj}, k_{0j})$. Where ID_{sj} may be the identity of the node s_j while $IDPAN_i$ may be the identity of the PAN PAN_i . The Phase- 2: Users joining the system: The user U_i must enroll with the server before issuing a summons to MSN. In the wake of checking his enlistment data, the system server assigns a character, say ID_{Ui} , for him. The Phase-3: Using system Regularly: Now the HSN is prepared for customary utilization. The Phase-4: Issue of commands by users: After receiving proxy signature keys, if an individual, U_i , who is ready to execute an instruction, has to develop the instruction Que and after that it imprints a trademark sign on the Que.

3.1 Setting up experiment for implementation

To evaluate the likelihood of the framework that is proposed upon the biosensors, [2-8] for securing the data generated by HSNs, two sensors have been identified and are TelosB & MicaZ bits. The TelosB bit is outfitted with a

- MSP430 microcontroller: 8-MHz, 16-bit
- RAM: 10-kB,
- ROM: 48-kB,
- 1024-kB streak and
- 802.15.4/Zig-Bee radio.

Likewise, the MicaZ bit includes

- Atmel microcontroller: 8-MHz, 8-bit
- RAM: 4-kB and
- ROM: 128-kB.

In addition the system server, controller, organize client, and sensor hubs were also used. The conventions designed have been actualized in C language. Side projects of sensor node are written in nesC. TinyOS is used to run our motes[9] 2.x. All the programs have been repeated several times for obtaining better results.

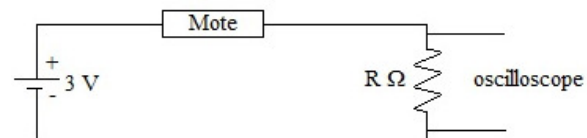


Figure 3. Setup for Energy consumption investigation

Circuit shown in the above figure has been built for calculating the energy consumed by various encoding methods in the sensor nodes [10]. At the point while a mote shown in the circuit diagram was executing an encoding method, the voltage drop across resistor V_r is recorded using an oscilloscope. Current I is calculated using Ohm's(Ω 's) Law and the Voltage V_m across mote is calculated using Kirchoff's Voltage Law (KVL). The power devoured through encoding method was computed by $V_m \times I$. Energy consumed can be calculated if the time of execution is noted down.

3.2 Evaluation Results

The system proposed can be evaluated by recording the metrics viz., memory overhead, execution time, and energy overhead. RAM and ROM used will give the memory overhead.

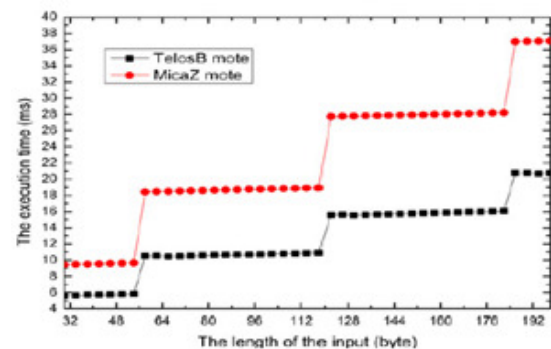
By changing the length of parameter modulus n and keeping message length as 8B, the recorded execution time periods of fundamental operations of system that is proposed have been noted in Table 1. It is observed that the operation of generation of server's public key and private key is only taking long time while all other operations are consuming the time in milliseconds only. The proposed protocols proves to be the good even on the smart phones also, as the minimum clock frequency of smart phones now a days is more than 1 GHz.

Table 1. Execution time for the fundamental operations for our system

Exp. 1			
Processor Speed	1.6GHz		
The Length of n	2Kb	1Kb	0.75Kb
System Initialization	37721ms	2252ms	760 ms
Proxy Signature key (PSK) generation	34ms	5ms	2ms
PSK verification	4ms	1ms	1ms
Proxy signing	6ms	2ms	1ms
Signature Verification	6ms	2ms	1ms
Exp. 2			
Processor Speed	2.2GHz		
The Length of n	2Kb	1Kb	0.75Kb
System Initialization	27224ms	1603ms	555 ms
Proxy Signature key (PSK) generation	26ms	3ms	2ms
PSK verification	3ms	1ms	0.4ms
Proxy signing	4ms	1ms	1ms
Signature Verification	5ms	1ms	1ms
Exp. 3			
Processor Speed	2.6GHz		
The Length of n	2Kb	1Kb	0.75Kb
System Initialization	23098ms	1387ms	472 ms
Proxy Signature key (PSK) generation	21ms	3ms	1ms
PSK verification	2ms	1ms	0.4ms
Proxy signing	4ms	1ms	1ms
Signature Verification	4ms	1ms	1ms
Exp. 4			
Processor Speed	3.1GHz		
The Length of n	2Kb	1Kb	0.75Kb
System Initialization	20236ms	1146ms	388 ms
Proxy Signature key (PSK) generation	18ms	2ms	1ms
PSK verification	2ms	1ms	0.3ms

Proxy signing	3ms	1ms	0.5ms
Signature Verification	3ms	1ms	0.4ms

Figure 4 has demonstrated the execution time of SHA-1 hash function (extracted from TinyECC 2.0 [11]) on MicaZ and TelosB motes. The contributions to the hash capacity are arbitrarily produced numbers with length differing from 30 to 198 bytes in additions of 4 bytes. The same practical has been played several times and averaged. For instance, the execution time on a MicaZ bit for contributions of 30, 58, 122, and 186 bytes are 9.4609, 18.4498, 27.7263, and 37.0021 ms, individually. Likewise, the execution time on a TelosB bit for contributions of 30 and 58 bytes are 5.6012 and 10.50473 ms, individually.

**Figure 4.** Execution of SHA-1 hash function on MicaZ and TelosB motes

From Figure 4 it can be observed that the more the length of the PHD in the proposed framework in each round, the greater the time utilization of hash operation. The length of the PHD in each round ought to be picked to accomplish a harmony between the bundle length and registering many-sided quality.

The energy consumed by SHA-1 hash function is explored. When a MicaZ mote is used in the circuit, $V_r = 144$ mV, $I = 7.0727$ mA, $V_m = 2.8160$ V, and $P = 19.9167$ mW. When a TelosB mote is used, $V_r = 40$ mV, $I = 1.9646$ mA, $V_m = 2.9200$ V, and $P = 5.7366$ mW. Total energy consumed can be calculated from power and execution time from figure. 4. Even the energy consumed by SHA-1 on

motes can also be calculated from the values recorded.

Advanced Encoding Standard (AES) is considered to be one of the efficient encoding and decoding algorithm. And this supports most of the hardware for the HSNs. Thus this algorithm has been used for implementation of the proposed system on the CC2420 radio hardware on MicaZ and TelosB motes.

From Table 2 we can find the execution time of AES encoding/decoding for specified motes with the plaintext of variable length. The Table 3 demonstrates the execution time of remain solitary equipment AES encoding for the same motes and with similar conditions. It is noticed that hardware AES operation is much quicker than software AES operation for a similar level of security.

Table 2. Execution Time of Software AES for MicaZ and TelosB Motes with the Plaintext of Different lengths

MicaZ						
Plaintext length	48 B	96B	112 B	128 B	144 B	160 B
Encryption time	4ms	7ms	8ms	9ms	11ms	12ms
Decryption time	4ms	8ms	10ms	11ms	12ms	14ms
TelosB						
Plaintext length	48 B	96B	112 B	128 B	144 B	160 B
Encryption time	6ms	11ms	13ms	15ms	17ms	18ms
Decryption time	7ms	13ms	15ms	18ms	20ms	22ms

Table 3. Execution Time of Stand-alone Hardware AES for MicaZ and TelosB Motes with the Plaintext of different lengths

MicaZ						
Plaintext length	48B	96B	112 B	128 B	144 B	160 B
Encryption time	0.1ms	0.21ms	0.25ms	0.28ms	0.3ms	0.35ms
TelosB						
Plaintext	48B	96B	112	128	144	160

xt length			B	B	B	B
Encryption time	0.28 ms	0.54 ms	0.6ms s	0.7ms s	0.8 ms	0.84 ms

It is found that the energy utilization of hardware AES is much lower than that of software AES for a similar level of security. As indicated by the above investigation, the proposed algorithms are a great deal more productive than traditional public key algorithms [6-8,12] as far as calculation complexity and energy utilization.

Next, the entire proposed system has been implemented. using CC2420 stand alone hardware AES algorithm for encoding the noted data from sensors. Every node uses AES decoding module to decode the reaction from server. Since the key size of the utilized AES calculations is 16 bytes while the yield of SHA-1 capacity is 20 bytes, just the initial 16-byte information are utilized as a part of the key overhaul operation.

Table 4. Code Sizes of the Whole system

	Memory	Controller	Node
MicaZ	ROM	18 bytes	15 bytes
	RAM	645 bytes	483 bytes
TelosB	ROM	17 bytes	16 bytes
	RAM	688 bytes	463 bytes

Table 4 demonstrates the size of code of entire framework. The TelosB mote's implementation has used nearly 17 KB of ROM and 450B of RAM. It corresponds to just 33% and 5% of the ROM and RAM limits of TelosB. The execution overhead of 33% of ROM is worthy. At last, total energy of both the cases has been compared. *Case1*: biosensor node transmitting data to the controller without any security protection and *Case2*: with the security scheme proposed in the paper. Expecting that data transfer rate of mote is 160 kb/s, the time of data transmission 20 bytes is 1ms, additional devoured energy is worthy.

4. Conclusion

In this paper, security challenges confronting a HSN for remote health checking are identified. The security examination has shown that proposed framework can accomplish the prerequisites of the

convention of this kind. The conventions are implemented in real time system using sensor nodes with restricted asset. Exploratory outcomes have demonstrated that our methodologies are possible for certifiable applications.

References

- [1] 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.
- [2] Shnayder V, Chen B.R, Lorincz K, Fulford-Jones T R F, and Welsh M, Sensor networks for medical care, Harvard Univ., Cambridge MA, USA, Tech. Rep. TR-08-05, 2005
- [3] Malasri K and Wang L, Addressing security in medical sensor networks, Proc. ACM HealthNet, 2007, pp 7-12
- [4] Rajasekaran R, Manjula V, Kishore V, and Sridhar T, Jayakumar C, An efficient and secure key agreement scheme using physiological signals in body area networks, Proc. Int. Conf. Advances Computer. Comm. Informat, 2012, pp 1143-1147
- [5] R.Kavitha and A.Suganthi, "A Study on Smart Clothing Technology Using Google Glass", International Innovative Research Journal of Engineering and Technology, Aug 2016, pp 76-81
- [6] Wang H, Fang H, Xing L, and Chen M, An integrated biometric based security framework using wavelet-domain HMM in wireless body area networks (WBAN), Proc. IEEE Int. Conf. Comm., Jun-11, pp 1-5
- [7] Malasri K and Wang L, Design and implementation of a secure wireless mote-based medical sensor network Sensors, vol. 9, no. 8, Aug-09, pp 6273-6297
- [8] Keoh S, Efficient group key management and authentication for body sensor networks Proc. IEEE Int. Conf. Comm., Jun. 2011, pp 1-6
- [9] Tan C.C, Wang H, Zhong S, and Li Q, IBE-lite: A lightweight identity based cryptography for body sensor networks, IEEE Trans. Inf. Technol. Biomed, vol. 13, no. 6, Nov-09, pp. 926-932
- [10] TinyOS: An open-source OS for the networked sensor regime, <http://www.tinyos.net>
- [11] Kapitanova K, and Son S, The price of security in wireless sensor networks, Computer Networks, vol. 54, no. 17, Dec. 2010, pp. 2967-2978
- [12] Liu A and Ning P, TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks, Proc. Inf. Process. Sensor Networks, 2008, pp. 245-256
- [13] Le X, Khalid M, Sankar R, and Lee S, An efficient mutual authentication and access control scheme for wireless sensor network in healthcare, J. Networks, vol. 6, no. 3, 2011, pp. 355-364

