International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:5, 2012

Implementation of Security Algorithms for u-Health Monitoring System

Jiho Park, Yong-Gyu Lee and Gilwon Yoon*

***Abstract*—**Data security in u-Health system can be an important issue because wireless network is vulnerable to hacking. However, it is not easy to implement a proper security algorithm in an embedded u-health monitoring because of hardware constraints such as low performance, power consumption and limited memory size and etc. To secure data that contain personal and biosignal information, we implemented several security algorithms such as Blowfish, data encryption standard (DES), advanced encryption standard (AES) and Rivest Cipher 4 (RC4) for our u-Health monitoring system and the results were successful. Under the same experimental conditions, we compared these algorithms. RC4 had the fastest execution time. Memory usage was the most efficient for DES. However, considering performance and safety capability, however, we concluded that AES was the most appropriate algorithm for a personal u-Health monitoring system.

***Keywords*—**biosignal, data encryption, security measures, u-health

I. Introduction

IMPROVED medical services as fusion technology combined with bio-technology (BT), nano-technology (NT), information technology (IT) provide with upgraded and high-quality service to patients than before [1]. Mostly ubiquitous healthcare systems aim to offer non-constrained and even non-invasive measurement so that patients can move around and perform daily activities during measurement [2]. This is why u-Healthcare draws spotlight. In order for u-healthcare monitoring system to be successful, there are technical requirements to be fulfilled. They are user-friendly measurement, signal processing, communication, network management and traffic control, miniaturization and low-power technology and etc [3]. In addition to these, the security measures have been regarded as an important issue and have been studied in communication fields rather extensively. Interestingly enough, there have been few studies on security measures in u-health monitoring system.

A u-healthcare system monitors medical information, personal data and location information etc of a client or clients (end point) that receive medical service. This information is being transmitted to a remote healthcare server through wireless network or Internet [4]. The server can monitor health status, activity log and even a lifestyle. Physician can utilize this information and diagnose and take appropriate measures to a particular client. Above all, in an emergency case of accident, vital signs are transmitted to the remote healthcare server and a prompt clinical service can be provided with, which may save one's life.

J. H. Park (e-mail: loki361@seoultech.ac.kr), Y-G Lee (e-mail: yglee@seoultech.ac.kr) and G. Yoon are with Seoul National University of Science and Technology, Seoul, KOREA (phone: 82-2-970-6419; fax: 82-2-979-7903; e-mail: gyoon@seoultech.ac.kr). *Correspondence to G. Yoon.

However, this personal information can be relatively easy to be accessed due to the nature of wireless or internet based communication. We may not exclude a possibility of being abused and the personal information may be put to a bad use [5]. Therefore, it will be necessary to apply security technology through which only the server and end point user can read data correctly in the u-health system.

U-health monitors should be very compact and consume power as less as possible. Unfortunately, the hardware constrains makes implementing security measures to be difficult since computational speed, capabilities of arithmetic operations and memory size tend to be very limited.

In this study, we proposed several encryption algorithms suitable for our u-health monitoring system and tested them successfully.

II. U-health Monitoring System

We have reported on our development of u-health monitoring system [6-8] previously. Our system consists of three parts of the biosignal measurement unit (BMU), wireless network system based either on a gateway or smart phone and server unit.

The BMU measures vital signs of a user and sends them to the gateway [7] or smart phone [8]. ECG, acceleration, temperature, photoplethysmogram (PPG) and SpO_2 were measured. The on-board microprocessor of the BMU was a 32-bit microcontroller (STM32L151R8, STMicroelectronics) and an extra 2 GB SD memory was used.

Fig. 2 illustrates network systems that interface between the BMU and a remote server. In case that the BMU is connected to a smartphone, the BMU data are sent to the smartphone through the Bluetooth wireless network. Then the smartphone transmits data to the remote server through Wi-Fi or 3G. When we do not use the smartphone, the BME sends data to a ZigBee- WLAN (wireless local area network) gateway through ZigBee wireless network. Then the gateway relays data to the remote healthcare server through WLAN.

Data from the BMU were compressed in order to increase the amount of data for a given network capacity. Compressed data rates were 192 B/s for ECG, 192 B/s for PPG, 56 B/s for temperature, 80 B/s for accelerations. The rest of SpO_2, heart rate (HR) and pulse transit time (PTT) were 2 B/s each. In addition to health information, the location information was 8 B/s each for latitude and longitude. Other data, 70 B/s, included personal information such as name, address, ID number and phone number as well as the messages from a physician or medical specialist to a client or vice versa. The end-point was to send a 604 B/s of data.

The server unit was a personal (PC) or laptop computer based system. Even though PC is inexpensive and simple, all

the real time health information for multiple users were nicely and efficiently displayed so that one physician or healthcare personnel could handle multiple patients at the same time.

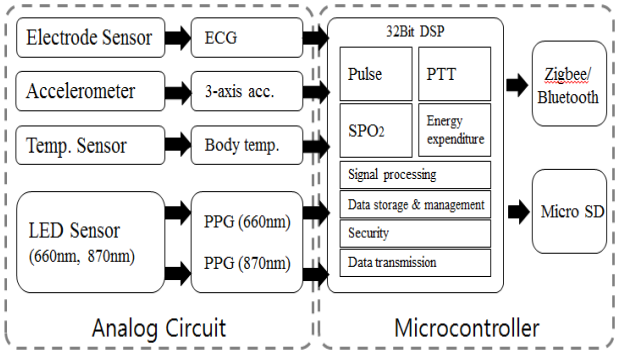

Fig. 1 Block diagram of Biosignal Monitoring Unit (BMU).

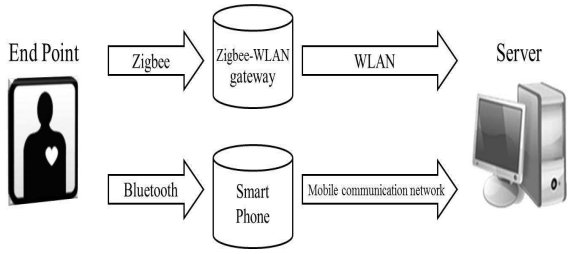

Fig. 2 Network connections between the client (endpoint) and the server

TABLE I
DATA RATES OF U-HEALTH MONITORING SYSTEM

	ECG	PPG	temp	accele ration	SpO_2
Full waveform monitor	192 B/s	192 B/s	56 B/s	80 B/s	2 B/s
	HR	PTT	location Info	personal Info	total
	2 B/s	2 B/s	8 B/s	70 B/s	604 B/s

III. SECURITY ALGORITHMS

It is necessary to encrypt transmission data being transmitted at the end point in order to increase the strength of information security. In encryption, there are two types of algorithms. One uses symmetric key and the other adapts public key. The strength of security algorithm is highly related with the key length.

In point of the standard on key publishing, there are two classes. First one is the symmetric-key algorithm. The feature is to use the same key during encryption and decryption. This class is faster than the public key cryptography because the symmetric-key algorithm is based on simple operations such as XOR and multiplications and uses smaller numbers (64 or 128 bits). But the symmetric-key algorithm requires key synchronization.

Public key algorithm, on the other hand, is generally slower than the symmetric key algorithm because it uses complex mathematical operations and requires two separate keys each for encryption and decryption. The BMU contains an embedded system that should fulfill the requirement of miniaturization and low-power consumption. A large number of operations are detrimental in meeting the demand of low-power consumption. Therefore, for u-health monitoring system, we proposed to apply several symmetric key algorithms that could be implemented by using simple arithmetic operations. Table II lists the keys to be applied.

TABLE II
ENCRYPTION SCHEMES AND PARAMETERS

	Block cipher			Stream cipher
	Blowfish	DES	AES	RC4
Block(bits)	64	64	128	8
Key (bits)	56	56	128	128
Round	16	16	10	256

DES = data encryption standard, AES = advanced encryption standard
RC4 = Rivest Cipher 4

Block cipher types according to the key were Blowfish, data encryption standard (DES), advanced encryption standard (AES). Another algorithm to be implemented was Rivest Cipher 4 (RC4) based on Stream cipher type. These schemes have a characteristic of simple operational process.

Block cipher uses a fixed-length groups of bits called block and return cipher-text) n-bit block with respect to an n-bit of plain text block input. Stream cipher substitutes cipher text output following a pseudo-random cipher stream (key stream) for plain text input.

Blowfish is a symmetric block cipher algorithm [9]. This algorithm uses a variable key length between 32 and 488 bits and has a 64-bit of block size. 32-bit input is divided into four 8-bit inputs. Each 8-bit input is placed into S-boxes that accept 8-bit inputs and produce 32-bit outputs dependent on the key. The outputs are added and XORed to produce the final 32-bit output. In this work, we set the key length to be 56-bit which is the same as that of DES.

DES is a symmetric block cipher algorithm [10]. DES has a block size of 64 bits and uses a 56-bit key. Output of encrypted 64-bit block is produced through round operations of more than 3 times. In our work, we built as 16 round operations. Each round operation is implemented as follows. Before round operation, initial permutation (IP) was performed for 64-bit plain text and 64-bit block is divided into two 32-bit blocks (Lo and Ro). Ro is placed into F-function with the key and the F-function produced 32-bit blocks. The produced block is XORed with Lo(L1). Ro and L1 are criss-crossed. The F-function has four different stages called Expansion, Key mixing, Substitution and Permutation.

AES is also a symmetric cipher algorithm [11]. AES has a fixed 128-bit block size and its key sizes are 128, 192 and 256 bits. The number of round is 10, 12 and 14 depending on the key length. Encryption had four stages such as KeyExpansion, Initial Round, Rounds and Final Round. In our study, 10 rounds were executed using 128-bit key.

KeyExpansion is round keys creation process from the cipher key using the key schedule.

Initial Round (AddRoundKey) : Each byte of the state is XORed with the round key.

Rounds includes four different steps. Each step is called SubBytes, ShiftRows, MixColumns and AddRoundKey respectively.

SubBytes is a non-linear substitution step where each byte in matrix is updated using S-box.

ShiftRows is a transposition step where each row of the state is shifted cyclically with different byte number offsets.

MixColoumns is a mixing operation. Each column of the state is multiplied with a polynomial expression.

AddRoundKey is the same as Initial Round.

Final Round is the same as round operation except MixColounms.

RC4 is a symmetric stream cipher algorithm [12]. RC4 is consisted of two parts. They are permutation of 256 possible bytes and two 8-bit index pointers. The permutation is initialized by a variable length key using the key-scheduling algorithm (KSA). Once this is completed the stream of bits is generated using the pseudo-random generation algorithm (PRGA). In this study, we used a128-bit key. In KSA, 256-bit array S is filled from 0 to 255 and swap values of S with the key. PRGA changes the state and generates one-byte key stream output. In each loop, PRGA increments i and the value of S[i] is added to j. S[i] and S[j] are switched and output becomes the value of S[S[i]+S[j] mod 256]. Each element of array S is changed at least once.

IV. Results

Our Biosignal Monitoring Unit (BMU) measured biosignal from the subjects and received data as a sampling rate 300 per second. Data was preprocessed (for example, noise removal and band pass filtering) and then compressed. Afterwards, data were encrypted.

During one second period, 300 sampling was performed and 604 bytes should be transmitted. Therefore, more than two bytes per each sampling time (3.33ms) should be sent through Zigbee or Bluetooth queue every time. There was other restriction that all operations from sampling data to encryption should be finished within an interval of 3.33 ms because of the sampling rate.

In order to keep a low power mode, we did down-clock and operated the microcontroller at 8 MHz and limited data memory to be 10 Kbytes. All data processing except the security algorithm were the same. Each of our proposed four methods among the symmetric key algorithms was implement with pre-shared keys in our u-Health monitoring system. Each algorithm should be run under the operation performance of the microcontroller such as execution time and the limited memory size. Above all, the security strength was the top consideration.

Fig. 3 is an example of output waveform where ECG data was encrypted by Blowfish. Fig. 3(a) is the result of the correct decryption and shows correct waveform of ECG. Fig. 3(b) show ECG waveform when an incorrect key was used during decryption.

We observed similar figures for DES, RC4 and AED. We showed that confidentialness for data transmission was maintained by using the security algorithms.

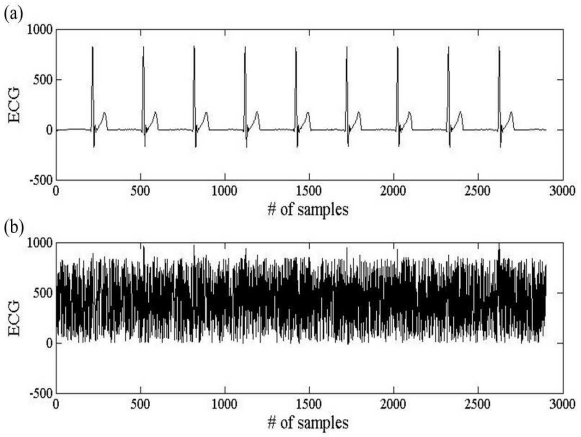

Fig. 3 Results of Implementing of the Blowfish algorithm to ECG signal. (a) is a correct decrypted data with the original key, (b) is an incorrect decrypted data with an invalid key.

For performance comparisons, we checked total clock cycles including encryption for each data sampling. We measured execution time and data memory usage including those assigned for security algorithm. We confirmed that all four algorithms worked properly under the time restriction of 3.33 ms and the memory size less than 10 Kbytes.

TABLE IIIII
Result of Security Algorithm Implementation

	Blowfish	DES	AES	RC4
Clock Cycles	15500	15000	20500	10250
Execution Time(ms)	1.94	1.85	2.56	1.28
Memory Usage(bytes)	6494	2525	3122	4313

Blowfish took about 15,500 clock cycles from data sampling to encryption and it was 1.94 ms at 8 MHz operation. It means that all process was done within the clock cycle. Blowfish needed a data memory of 6,494 bytes for data sampling, signal processing and encryption. DES required about 15,000 clock cycles and a memory usage of 2,525 bytes. During our experiment, operation speed was about the same. Normally it is known that Blowfish operates faster than DES. Blowfish has room for improvement by optimizing programming. AES spent 20,500 clock cycles that corresponded to 2.56 ms, still under 3.3 ms. AES allocated 3,122 bytes of memory. RC4 finished operation within 1.28 ms (about 10,250 clock cycles) and proved to be the fastest. RC4 used 4,313-byte memory. Table III summarized clock cycles, execution time and memory usage. RC4 was the fastest and AED required the smallest memory size.

V. Discussion and Summary

We studied the security algorithms that were necessary to secure transmission data in a biomedical telemetry of u-Health monitoring system. We investigated the security algorithms that could work properly in an embedded system with limited computation capabilities and small memory unit. We implemented several algorithms in our u-Health monitoring system and found that all operated by the 32-bit microcontroller (STM32L151R8, STMicroelectronics).

However, the strength of symmetric key algorithm is dependent on key length since key length determines the number of possible keys. There is brute force attack where every possible key is tried to intrude. Therefore, DES and Blowfish with a shorter key length of 56 bits can be more vulnerable to brute force attack than AES and RC4 (128 bits). Blowfish can use a longer key as that of AES, but Blowfish have a disadvantage of memory usage. Implementation of Blowfish required more than 4 Kbytes which was the largest memory usage. We suggest that AES is more suitable than Blowfish because of small memory usage considering a limited memory size, for example 10 Kbytes, that is a typical feature of low-end microprocessors employed in u-Health system.

RC4 uses the same key streams generated by the key scheduling algorithm when a long-term or pre-shared key is adapted. This is a well-known weakness of RC4 [13]. Key stream and plaintext can be discovered by collecting many ciphertexts when the operation of ciphertext = plaintext⊕key stream makes for plaintext to be XORed with the same key stream.

On the other hand, AES does not have a well-known weakness. At the same time, AES has a 128-bit long key that ensures safety and maintains lower memory usage than Blowfish and RC4. We concluded that AES was the most appropriate algorithm for our system.

Acknowledgment

This study was financially supported by Seoul National University of Science & Technology.

References

[1] P. J. Feldstein, *Health care Economics*, New York: Delmar, 2011, ch. 10.
[2] K. Yamakoshi, "Unconstrained physiological monitoring in daily living for health care," *Frontiers Med. Biol. Eng.*, vol. 10, no. 3, pp. 139–159, Sep. 2000.
[3] K. H. Lee, "Application of u-Health to Emergency Medical Service System," *J. Korean Med. Assoc. Eng.*, vol. 52, no. 12, pp. 1148–1153, Dec. 2009.
[4] K. Jeong, E-Y Jung, and D. K. Park, "Trend of wireless u-Health," in *9th symposium on Communications and Information Technology 2009 (ISCIT 2009)*, Incheon, Korea, 2009, pp. 829–833.
[5] H. S. Ng, M. L. Sim, and C. M. Tan, "Security issues of wireless sensor networks in healthcare applications," *BT Technology Journal*, vol. 24, no. 2, pp. 138–144, April 2006.
[6] W. Shin, Y. D. Cha, and G. Yoon, "ECG/PPG integer signal processing for a ubiquitous health monitoring system," *Journal of medical systems*, vol. 34, no.5, pp. 891-898, May. 2009.
[7] Y. D. Cha, and G. Yoon, "Ubiquitous health monitoring system using Zigbee and wireless LAN dual-network," *Telemedicine and e-health*, vol. 15, no.9, pp. 891-897, Nov. 2009.
[8] Y-G Lee, and G. Yoon, "Smartphone-based Mobile Health Monitoring System for Multiple Users", accepted, *Telemedicine and e-health*.
[9] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," in *Fast Software Encryption (FSE'93)*. vol. 809, LNCS 809, New York: Springer-Verlag, 1994, pp. 191–204.
[10] F.I.P. Standard, "Data Encryption Standard (DES)," *National Institute of Standards and Technology (NIST)*, FIPS 46-3, Oct 1999
[11] K. Wu, R. Karri, G. Kuznetsov, and M. Goessel, "Low cost concurrent error detection for the advanced encryption standard," in *International Test Conf. (ITC2004)*, pp. 1242–1248.
[12] P. Prasithsangaree, and P. Krishnamurthy, "Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs," in *IEEE Global Telecommunications Conf. (GLOBECOM ' 03)*, pp. 1445–1449.
[13] Scott R. Fluhrer, Itsik Mantin and Adi Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Selected Areas in Cryptography 2001*, pp. 1 – 24.

Jiho Park was born in Daegu, South Korea (1984) and received his B.S. degree in electronics and information engineering from Seoul National University of Science and Technology, Seoul, South Korea (2012).

Yong-Gyu Lee was born in Seongnam, South Korea (1984) and received his B.S. degree in electronics and information engineering from Seoul National University of Science and Technology, Seoul, South Korea (2010). He is currently a M.S. student at Seoul National University of Science and Technology and engages in biomedical engineering researches.

Gilwon Yoon received his B.S. in Electrical Engineering from Seoul National University, Seoul, Korea in 1977 and M.S. and Ph.D. in Electrical and Computer Engineering from the University of Texas at Austin, U.S.A. in 1982 and 1988 respectively. He worked at Samsung Advanced Institute of Technology, Korea between 1992 and 2003. He is with Seoul National University of Science and Technology since 2003.