

South Dakota State University

Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange

Electronic Theses and Dissertations

2021

Improved Secure and Low Computation Authentication Protocol for Wireless Body Area Network with ECC and 2d Hash Chain

Soohyeon Choi
South Dakota State University

Follow this and additional works at: <https://openprairie.sdstate.edu/etd>



Part of the [Computer Engineering Commons](#), [Electrical and Computer Engineering Commons](#), and the [OS and Networks Commons](#)

Recommended Citation

Choi, Soohyeon, "Improved Secure and Low Computation Authentication Protocol for Wireless Body Area Network with ECC and 2d Hash Chain" (2021). *Electronic Theses and Dissertations*. 5641.
<https://openprairie.sdstate.edu/etd/5641>

This Thesis - Open Access is brought to you for free and open access by Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Open PRAIRIE: Open Public Research Access Institutional Repository and Information Exchange. For more information, please contact michael.biondo@sdstate.edu.

IMPROVED SECURE AND LOW COMPUTATION AUTHENTICATION
PROTOCOL FOR WIRELESS BODY AREA NETWORK WITH ECC AND 2D HASH
CHAIN

BY
SOOHYEON CHOI

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2021

THESIS ACCEPTANCE PAGE

Soohyeon Choi

This thesis is approved as a creditable and independent investigation by a candidate for the master's degree and is acceptable for meeting the thesis requirements for this degree.

Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Kwanghee Won
Advisor

Date

Sid Suryanarayanan
Department Head

Date

Nicole Lounsbery, PhD
Director, Graduate School

Date

ACKNOWLEDGEMENTS

I would like to thank the South Dakota State University for giving me the opportunity to meet Dr. Kwanghee Won, my beloved thesis adviser. The completion of this study could not have been possible without his advice. I would also like to thank Dr. Sung Shin, my graduate coordinator for helping guide me a lot and taking time to read my thesis. Due to their effort and interest, the quality of my thesis has greatly improved. Aside from that, I cannot forget the time in which I spent with all of the CCT (Convergent Computing Technology) lab members.

I am sincerely thankful Mr. Adam Molstad and Mrs. Kristin Molstad for correcting error in my thesis and providing feedbacks. Their feedbacks were really valuable and helpful. I am also thankful to all of my committee members: Dr. Kwanghee Won, Dr. Sung Shin, Dr. Joshua Reineke, and Dr. Sid Suryanarayanan.

Last, I would like to thank my parent. They always believe me without a single doubt that I am capable of accomplishing goals in my life. Thanks to their effort, I could finish my master's program safely.

CONTENTS

ABBREVIATIONS.....	v
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
I. ABSTRACT.....	viii
II. INTRODUCTION.....	1
III. RELATED WORK.....	4
IV. PRELIMINARIES.....	7
1. Elliptic Curve Cryptography.....	7
2. 2D Hash Chain: SHA-256 and SHA-512.....	8
V. PROPOSED MODEL.....	12
1. Subtraction-Modular Random Key Selection.....	13
2. Initialization.....	15
3. Synchronization.....	16
4. Authentication.....	18
VI. ANALYSIS.....	21
1. Performance Analysis.....	21
2. Security Analysis.....	24
VII. CONCLUSION.....	27
VIII. LITERATURE CITED.....	29

ABBREVIATIONS

WBAN	Wireless Body Area Network
ECC	Elliptic Curve Cryptography
ECG	Electrocardiogram
KDC	Key Distribution Center
KCI	Key Compromise Impersonation

LIST OF FIGURES

Figure 1. The architecture of Wireless Body Area Network.....	2
Figure 2. $N \times N$ key pool created by 2D hash chain technique	9
Figure 3. Different key pools created by different keys.....	10
Figure 4. The worst case of random key selection algorithm.....	13
Figure 5. The bad cases of random key selection algorithm.....	14
Figure 6. Initialization.....	15
Figure 7. Synchronization and Authentication.....	17
Figure 8. Simple Authentication.....	19

LIST OF TABLES

Table 1. Notations.....	12
Table 2. Execution time for each operation.....	21
Table 3. Computation cost comparison.....	23
Table 4. Communication cost comparison.....	24
Table 5. Security feature comparison.....	25

ABSTRACT

IMPROVED SECURE AND LOW COMPUTATION AUTHENTICATION
PROTOCOL FOR WIRELESS BODY AREA NETWORK WITH ECC AND 2D HASH
CHAIN

SOOHYEON CHOI

2021

Since technologies have been developing rapidly, Wireless Body Area Network (WBAN) has emerged as a promising technique for healthcare systems. People can monitor patients' body condition and collect data remotely and continuously by using WBAN with small and compact wearable sensors. These sensors can be located in, on, and around the patient's body and measure the patient's health condition. Afterwards sensor nodes send the data via short-range wireless communication techniques to an intermediate node. The WBANs deal with critical health data, therefore, secure communication within the WBAN is important. There are important criteria in designing a security protocol for a WBAN. Sensor nodes in a WBAN have limited computation power, battery capacity, and limited memory. Therefore, there have been many efforts to develop lightweight but secure authentication protocols. In this thesis, a computationally efficient authentication protocol based on Elliptic Curves Cryptography (ECC) and 2D hash chain has been proposed. This protocol can provide high level security and require significantly low computation power on sensor nodes. In addition, a novel key selection algorithm has been proposed to improve efficiency of key usage and reduce computation cost. For this protocol, ECC is used for key exchange and key encryption. The scheme encrypts a key with ECC to create a pair of points and uses this pair of points as keys for an intermediate node and sensor nodes. 2D hash chain technique is used for generating

2D key pool for authentication procedure. This technique can generate many keys efficiently and effectively with hash functions. For security part, this protocol provides essential security features including mutual authentication, perfect forward security, session key establishment, and etc., while providing high level security. In experimental results, this protocol reduced sensor nodes' computation cost significantly by using combination of ECC and 2D hash chain. Moreover, the computation cost on the intermediate node has been reduced to 48.2% of the existing approach by the new key selection algorithm at an initial authentication. After the initial authentication, the intermediate node's computation cost is further reduced to 47.1% of the initial authentication by eliminating synchronization phase. In addition, communication cost which is the total packet size of all messages is 1280-bits, which is 5392-bits smaller than the existing approach, for entire authentication and after the initial authentication the cost is reduced to 768-bits.

INTRODUCTION

Interest in the healthcare area is growing since the population of elderly people in the world is increasing and they need healthcare systems for checking their health condition constantly [1]. Also, the development of technologies such as wireless communication, compact sensors, and low-power integrated circuits allows people make a new system called Wireless Body Area Network (WBAN) and this system has emerged as a promising technique for healthcare systems by providing remote and continuous patients' health condition monitoring system with small sensors [2-4]. WBAN consists of sensor nodes and an intermediate node (called as a server for convenience) which collects data from sensor nodes. In addition, there is Key Distribution Center to generate keys and system parameters and to distribute them. This thesis focuses on an authentication protocol between the sensors and the server node and tries to solve existing approaches' problems. The architecture of the WBAN is shown in Figure 1. The small sensors will be placed in, on, and around the patient's body and collect vital body function. Afterwards, the sensors send collected data to the server via short-range wireless communication. Since the sensors are attached to the patient's body for a long time to collect data, they need to be smaller and more compact. As a result, they cannot have a huge battery capacity and powerful computation ability. Therefore, to make the sensors have low battery consumption and low computation power is one of WBAN's challenges [5]. In addition, due to using a shared network for data transaction, it may have malicious attacks from unexpected adversaries [6]. Moreover, since human body function data such as electrocardiogram (ECG), blood pressure, blood sugar, etc., is sensitive and critical, the data should be checked before any transactions begin because if the data is modified

viciously, it may negatively affect the patients' health. Therefore, security of WBAN is becoming more and more important [7, 8].

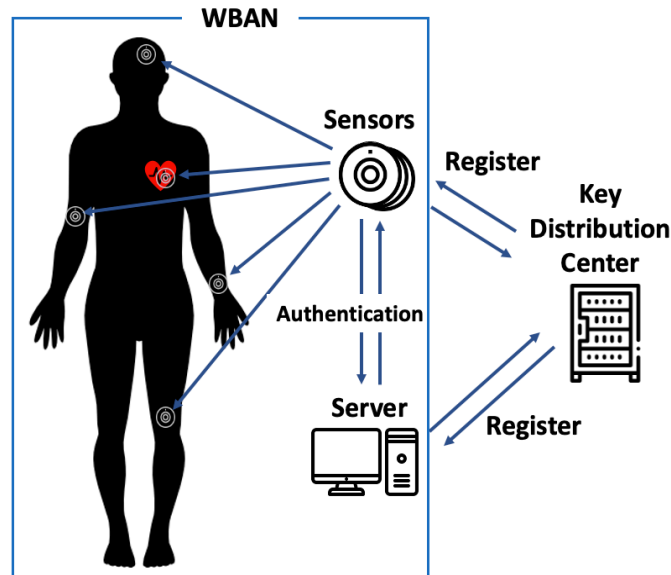


Figure 1. The architecture of Wireless Body Area Network

A secure and lightweight authentication protocol by applying Elliptic Curves Cryptography (ECC) with 2D hash chain is proposed to solve WBAN's problems [17]. However, this protocol has two problems. The first problem is that the server had to calculate too many hash functions to get keys. The second problem is that the key usage of a two-dimensional key pool is not efficient. Therefore, a key selection algorithm is modified to reduce the server's calculations and to increase the efficiency of key usage.

Elliptic Curves Cryptography (ECC) is a public-key cryptography tool based on an algebraic structure of elliptic curves over finite fields. ECC can use properties of an elliptic curve instead of big prime numbers like RSA, therefore, ECC offers the same security level as other widely deployed schemes with a smaller key [9, 10].

A hash chain technique is a successive application of a cryptographic hash function to a string key. A hash function is mathematical algorithm that maps arbitrary length data to fixed length data. In addition, it is designed as one-way function, therefore, it is practically infeasible to invert [11]. Thus, a hash function is a powerful and computationally efficient cryptography tool for WBAN [11].

As a result, ECC and hash functions are the most suitable cryptography schemes for resource constrained WBAN. Due to these characteristics, there have been many previous works. ECC is used as a key encryption and a key exchange technique, and the 2D hash chain technique is used as a key generation technique in this thesis. For 2D hash chain, SHA-256 is used as the main hash function and SHA-512 is used as the sub hash function. The combination of these techniques can make the authentication protocol more secure and computationally efficient, especially on the sensor side.

The rest of this paper is organized as follows. In Section III, the related work is briefly discussed. ECC and 2D hash chain are introduced in Section IV. The improved random key selection method and the proposed authentication protocol model's Initialization, Synchronization, and Authentication phases are described in Section V. Performance analysis and security analysis of our protocol are presented in Section VI. Finally, the conclusion of this paper is covered in Section VII.

RELATED WORK

In WBAN, it uses the wireless network and measured data from sensors is sensitive and life-relevant information, therefore a lot of work is focused on the security of data transaction [12, 13]. Moreover, they are trying to make lightweight authentication protocols since WBAN has restrictions about sensor's battery capacity and computation power [14, 15]. The sensor's size should be small-scale and minimized for the patient's convenience. Thus, the sensors do not have space for a big battery and some computation parts such as CPU, memory, etc. As a result, lightweight computation for authentication protocol is required. Several papers [12-15] about lightweight and secure authentication protocol for WBAN by applying ECC or/and hash functions are proposed which used the same techniques as this proposed protocol.

He, D *et al.* [12] proposed an anonymous authentication protocol for WBAN by applying an ECC based multiplication and hash function. Their protocol has reduced the computation burden on the client side which is a sensor side, and it has provable security. In [12], however, there are still several computations on sensors to verify that keys are correct or not. This means that the sensors are required to have some computation abilities and battery capacity because more computation leads to high battery consumption and requires sensors to have more powerful computation ability. Another anonymous authentication protocol proposed by K. Sowjanya *et al.* [13] is an enhanced lightweight ECC based end-to-end authentication protocol. This protocol removed the security vulnerabilities of the scheme of Li *et al.* [16] as well as reducing the overall complexity to reduce computation cost. Unfortunately, however, the sensors still need to compute two or more complicated calculations such as big number modular operation

and ECC based multiplications. M. Nikooghadam and H. Amintoosi *et al.* [14] proposed a secure and robust elliptic curve cryptography-based mutual authentication scheme. Their protocol was not for WBAN and the health care area, but they used ECC for the authentication protocol and tried to reduce computation costs on the user side, which is the same side as the sensors in a WBAN system. However, this protocol has the same problems as the other two protocols. They reduced some computations from entire protocol procedure, but the sensors still need to make several computations for authentication. In [15], Kumar *et al.* introduced a lightweight cloud-assisted authentication for WBAN. They used cloud technology to provide an identity-based anonymous authentication and key agreement (IBAAKA) protocol for WBAN. However, their protocol's computation cost is higher than [14]. A secure and lightweight authentication protocol has been proposed [17]. This protocol significantly reduced the sensors' computation cost and provided high level security by combining ECC and 2D hash chain techniques. The protocol has, nevertheless, two problems: inefficient key usage and too many hash calculations required on the server.

To solve these protocols' problems, the random key selection algorithm in the authentication protocol is improved while maintaining the security level. Through the improved random key selection algorithm, the efficient of key usage for the 2D key pool is increased and the number of hash function calculations on the server is reduced. In addition, this random selection algorithm does not require any complicated calculations on the sensors. The sensors only need to compare keys from the server with the 2D key pool and generate random numbers. As a result, this protocol can provide high level

security with ECC and 2D hash chain techniques, and it increases efficiency of key usage.

PRELIMINARIES

Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is one of several public key cryptography tools, which is a cryptography system that uses pairs of keys. This pair of keys consists of a public key and a private key [5, 18]. ECC is based on the algebraic structure of elliptic curves over a finite field, and it has been confirmed as an efficient and effective scheme [9, 10]. ECC can provide a 128-bit security level with a 256-bits key. This means that when ECC uses the same length of key as other schemes, it can provide a higher level of security. In addition, this technique uses properties of an elliptic curve instead of two very large prime numbers. As a result, ECC can also provide lower cost computation. Accordingly, it has an important advantage over the use of smaller keys to provide the same level of security [10]. Thus, ECC is a perfect tool for resource-constrained applications since it can provide high level security with low battery consumption and low computation power.

Elliptic curves are a plane algebraic curve, and they are symmetrical over the x-axis defined by an equation of the form (1):

$$y^2 = x^3 + ax + b \quad (1)$$

where x and y are the standard variables that define the function while a and b are the constant coefficients that define the curve under the condition $4a^3 + 27b^2 \neq 0$ [18].

The operations used on elliptical curves in cryptography are point addition and point doubling. The point addition “+” is defined as follows: Let the order of G be g , let

G be an additive cyclic group on an elliptic curve. Let $P, Q \in G$, l be a line containing P and Q , and let R be the third point of intersection of l with the elliptic curve. Let l' be a line connecting R and R' which is a point of x-axis symmetry of R . Then $P + Q$ is the point such that l' intersects the elliptic curve at R and R' . Scalar multiplication over the elliptic curve can be computed as follows: $k * P = P + P + \dots + P$ (k times) [18].

Encryption and decryption of ECC are defined as follows: Assume that a sender wants to send a message x to a receiver. For encryption, there are 3 steps. First, let x be any point on the elliptic curve. Second, the sender selects a random number k from $[1, g - 1]$.

Third, the encrypted message will be a pair of points m_1, m_2 where $\beta = pk * P$, $m_1 = k * P$, $m_2 = x + k * \beta$, P is a generator point on the curve and pk is a private key. For decryption, the receiver gets the m_1, m_2 from the server and decrypts them as follows: $x = m_2 - (pk * m_1)$ [18].

2D Hash Chain: SHA-256 and SHA-512

Hash functions, such as SHA-1 and SHA-2, are computationally efficient and powerful cryptography tools. A hash function is a mathematical algorithm that maps arbitrary length data to fixed length data. In addition, it is designed as one-way function, therefore, this function is practically infeasible to invert [11]. A hash chain method is a successive application of a cryptographic hash function to a string key. It can produce many one-time keys from a single key by computing a hash function in succession [19].

SHA-256 and SHA-512, which generate 256-bits and 512-bits of digested messages respectively, are used for 2D hash chain. SHA-256 as the main hash function

h_m and SHA-512 as the sub hash function h_s are applied to create an $N \times N$ hash chain key pool. At first, this algorithm performs h_s with a string key. After the first step, it applies h_m with the result of the first step $n - 1$ times to fill up the first row's columns of the key pool. To fill up the second row's columns, it computes h_s with the result of the first step and then performs h_m with itself $n - 1$ times. It does continue to compute h_s with the previous step's result $n - 1$ times and h_m with each result of h_s $n - 1$ times.

Figure 2 shows the completed key pool created by the 2D hash chain technique.

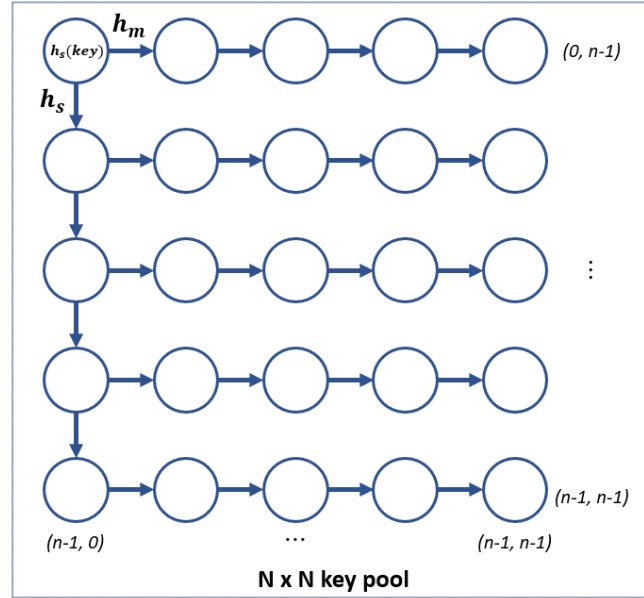


Figure 2. $N \times N$ key pool created by 2D hash chain technique

This key pool will be different when using different initial keys. Therefore, the 2D hash chain technique is able to create different key pools for each sensor with different keys as shown in Figure 3. When ECC encrypts a message x which is some point of the elliptic curve, it gets a pair of points m_1, m_2 as explained earlier. A server will have m_1 and the sensors will each have m_2 . Moreover, new m_2 which is interactive

with current m_1 can be produced through a reverse computation method. The reverse computation concept is defined as follows: $ECC_{e(x)} = m_1, m_2 \rightarrow ECC_{d(m_1, new\ m_2)} = new\ x$. Encrypt the x with ECC to get m_1, m_2 and decrypt m_1 and $new\ m_2$ to get new x . Thus, it can create one m_1 for the sever, and multiple m_{2-n} for each sensor.

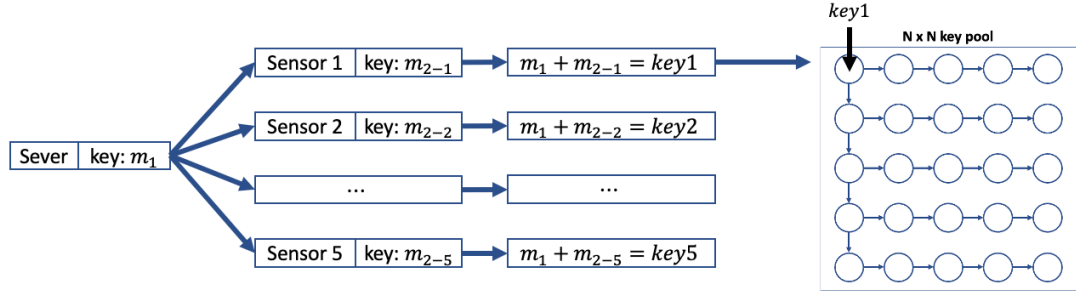


Figure 3. Different key pools created by different keys

Furthermore, to make this authentication scheme more secure, hash functions are modified to use a private key. Since SHA-256 and SHA-512 are open-source cryptography techniques, everyone can compute these methods exactly the same as you do if they know hash functions' key [20]. Thus, we need to secure hash function's key or modify hash function itself. By using the private key, key pools can contain different keys according to the private key. As a result, many key pools can be produced with one server's key, and attackers cannot produce the same key pool if they do not have the private key. The modified hash function concept is defined as follows: There is a round function in each hash function and this process is repeated 64 times for SHA-256 and 80 times for SHA-512. Each round takes a fixed size input, which is a combination of the most recent message block and the output of the last round, and the input is sliced into 8

pieces as a, b, c, ..., h [20]. The sequence of sliced pieces of the input is changed by using the selected private key to achieve a different result of each round function.

PROPOSED MODEL

In this section, the improved key selection algorithm and the proposed authentication model's 3 steps are described. For this protocol, several notations are used for ECC and hash function calculation. Table 1 shows notations used in this paper. E is an equation for ECC, G is a generator point on the elliptic curve, p is a prime number for modular operation, k is a random number for ECC, pk is a private key for ECC, n is the number of sensors, key is a point on the elliptic curve, m_1, m_2 are a pair of the encrypted key , h_m and h_s are the main and sub hash functions, pk_h is a private key for hash functions, kp is a two-dimensional key pool, r and c are the row and column number for key selection algorithm, key_s and key_D are keys for synchronization, and key_{sa} and key_{Da} are keys for authentication.

Table 1. Notations

Symbol	Description
E	An equation for ECC
G	A generator point on the elliptic curve
p	A prime number for modular operation
k	A random number for ECC
pk	A private key for ECC
n	The number of sensors
key	A point on the elliptic curve
m_1, m_2	A pair of the encrypted key
h_m	The main hash function
h_s	The sub hash function
pk_h	A private key for hash functions
kp	A two-dimensional key pool
r	The row number for key selection
c	The column number for key selection
key_s	Sensor's key for synchronization
key_D	Server's key for synchronization
key_{sa}	Sensor's key for authentication
key_{Da}	Server's key for authentication

Subtraction-Modular Random Key Selection

When the protocol produces the 2D hash key pool, there are keys as many as $N \times N$. Each sensor will have a different key pool and select one key for each authentication procedure. Thus, a key selection algorithm is required for the authentication. The Random Key Selection algorithm was proposed which generates a random number between 1 and 3, and if the number is 1 (case 1), then a new key will be above the current position. If the generated number is 3 (case 3), then a new key will be left of the current position. If the random number is 2 (case 2), then generate a new random number and restart at a diagonal from the current position [17]. However, this algorithm has problems. If the case 2 continues to be generated, then it could encounter the worst case. The worst case means that only 1 key out of 25 keys is used for a 5×5 key pool as shown in Figure 4. Moreover, the bad cases occur when the case 1 (or 3) continues to be generated. The bad case means that only 4 keys out of 25 keys are used

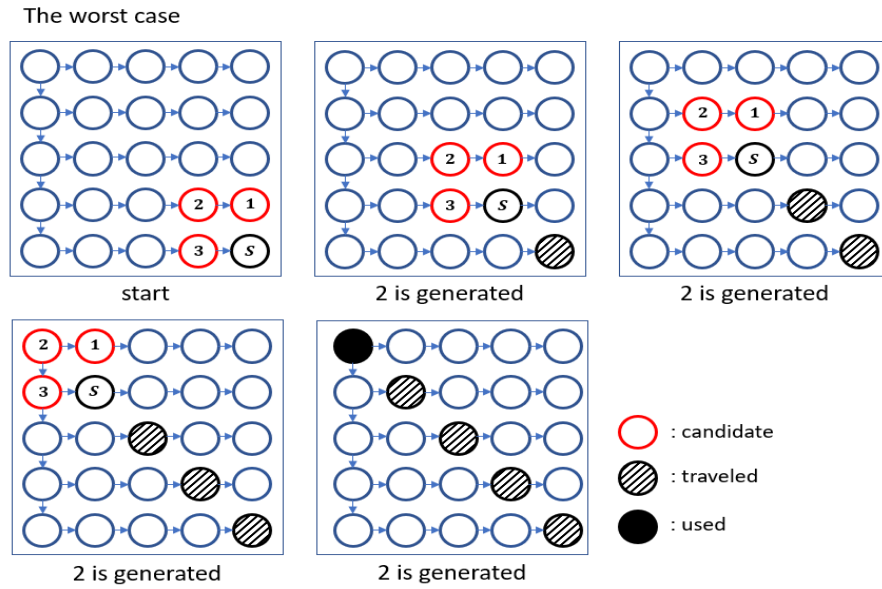


Figure 4. The worst case of random key selection algorithm

for a 5×5 key pool as shown in Figure 5. Therefore, this algorithm is needed to be redesigned to solve these problems.

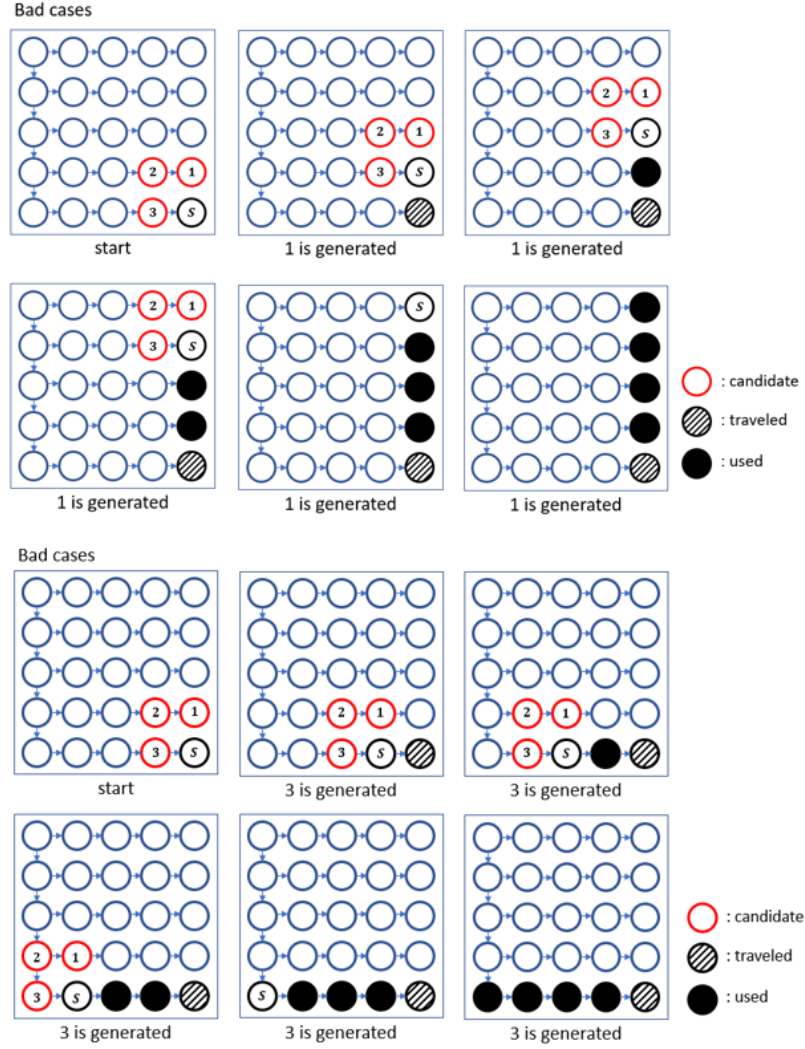


Figure 5. The bad cases of random key selection algorithm

The new algorithm randomly generates the column number and calculates the row number for the 2D key pool. The algorithm is defined as follows: When the sensor selects a key from the key pool, the column number (c) will be randomly generated by the sensor. After that it subtracts key_A from the previous key of key_A (p_key). Then it performs a modular operation with n to get the row number (r). As a result, $key_B(r, c)$

will be selected as a new key. This algorithm increased the efficiency of key usage from 16% to 90%. In addition, it reduced the number of calculations on the server from 83 times to 35 times (42.1%).

Initialization

Key Distribution Center (KDC) defines keys and system parameters and generates the two-dimensional key pool. After that KDC distributes parameters, keys, and the 2D key pool to the server and sensors. Figure 6 shows the initialization phase, and the initialization phase is defined as follows:

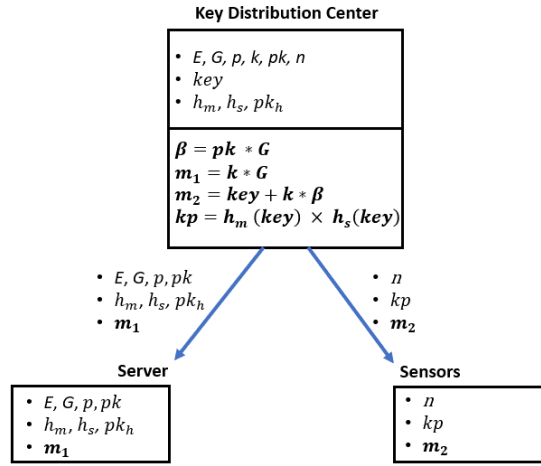


Figure 6. Initialization

- (1) KDC defines system parameters such as $E, G, p, k, pk, n, key, h_m, h_s, pk_h$.
- (2) E is an equation, G is a generator point on the elliptic curve, p is a prime number for modular operation, k is a random number, pk is a private key for ECC, n is

the number of sensors, key is a point on the elliptic curve, h_m and h_s are the main and sub hash functions, and pk_h is a private key for hash functions.

- (3) KDC creates two-dimensional key pool kp by calculating the main and sub hash functions with key as explained in IV.2 2D hash chain section. After that KDC encrypts key with ECC to get a pair of keys m_1 and m_2 for the server and sensor.
- (4) KDC distributes E, G, p, pk, pk_h , and m_1 to the server and n, kp , and m_2 to the sensor.

Synchronization

The server and sensor synchronize each other's key and 2D key pool to verify each other in the synchronization phase. Figure 7 shows the synchronization phase, and the synchronization phase is defined as follows:

- (1) At first, the server sends a request to the sensor to begin authentication.
- (2) When the sensor receives the request from the server, it selects the last key of the 2D key pool which is $kp(n - 1, n - 1)$ as key_s and sends m_2, n , and key_s back to the server.
- (3) The server decrypts m_2 received from the sensor with its m_1 by ECC to get key . After that, the server computes the main hash function h_m and the sub hash function h_s $n - 1$ times and $n - 2$ times respectively to get key_D as explained. 2D hash chain section. If key_s from the sensor is matched with key_D with 1 more computation of the main hash function, the server sends key_D to the sensor and saves its previous key as p_key_D . If key_s is not matched with key_D with 1 more

computation of the main hash function, consider the sensor to not belong to the system and reject this sensor.

- (4) The sensor compares key_D received from the server with the previous key of the key_S which is $kp(n-1, n-2)$. If they are matched, the sensor subtracts $kp(n-1, n-2)$ from p_key which is the previous key of $kp(n-1, n-2)$. After that, perform modular operation to the result with n to get the row number r . The sensor randomly generates the column number c . Afterwards, (r, c) will be selected as key_{Sa} . If they are not matched, reject it.

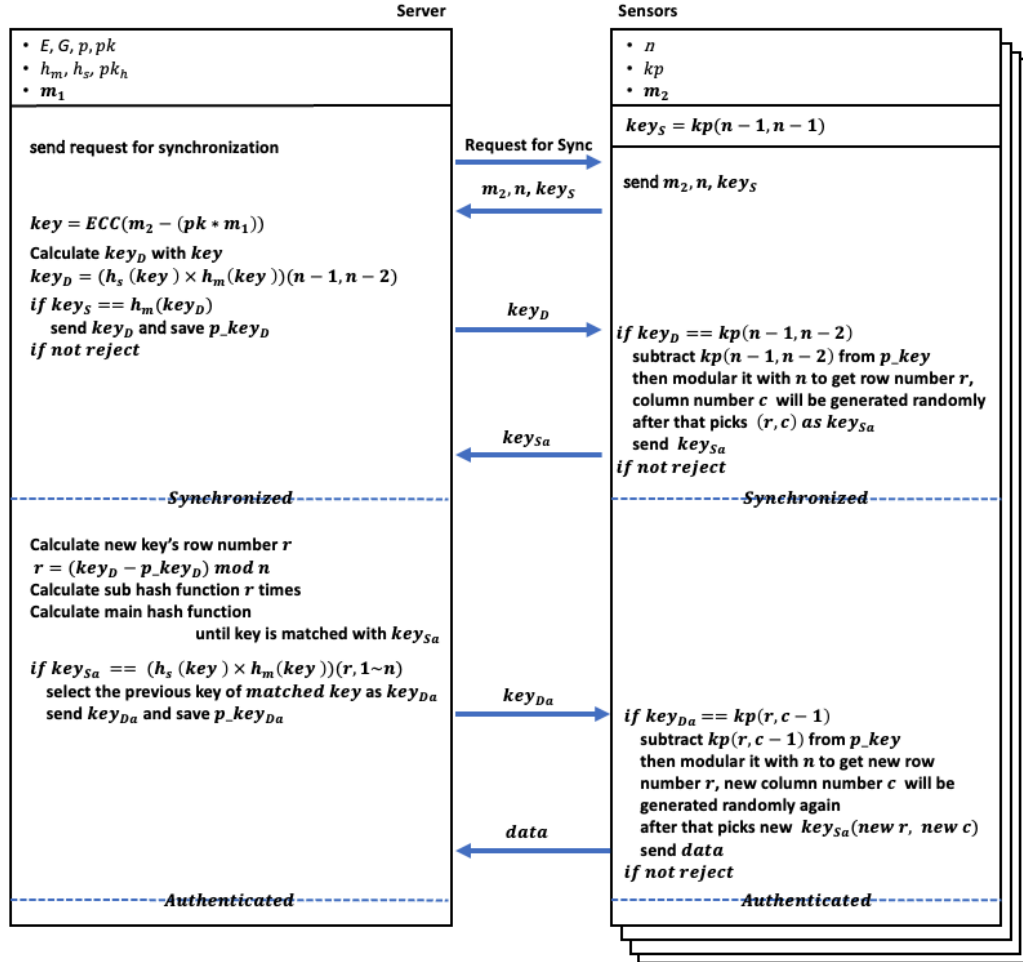


Figure 7. Synchronization and Authentication

After the step 3, the server can verify m_2 and key_s from the sensor is matched with its m_1 . After the step 4, the sensors can verify key_D from the server is matched with its key pool. As a result, both the sensor and server authenticated each other as valid devices and their keys and key pools are synchronized.

Authentication

In the authentication phase, the server and sensor authenticate each other through synchronized keys and key pools from the synchronization phase. After authentication, the sensor sends data to the server. If they authenticate each other successfully at first authentication, they do not need to synchronize their keys and key pool again. If they want to communicate again, they just need a simple authentication until they run out of all keys in the 2D key pool. The simple authentication phase is shown in Figure 8.

Authentication procedures are defined as follows:

- (1) After receiving key_{sa} from the sensor, the server calculates the new key's row number r as $r = (key_D - p_key_D) \bmod n$, where key_D is the server's key, p_key_D is the previous key of key_D . The subtraction with key_D and p_key_D is calculated, and then a modular operation with n is performed to get the row number r .
- (2) Afterwards, the server calculates the sub hash function h_s r times and then calculates the main hash function h_m until the calculated key is matched with key_{sa} .

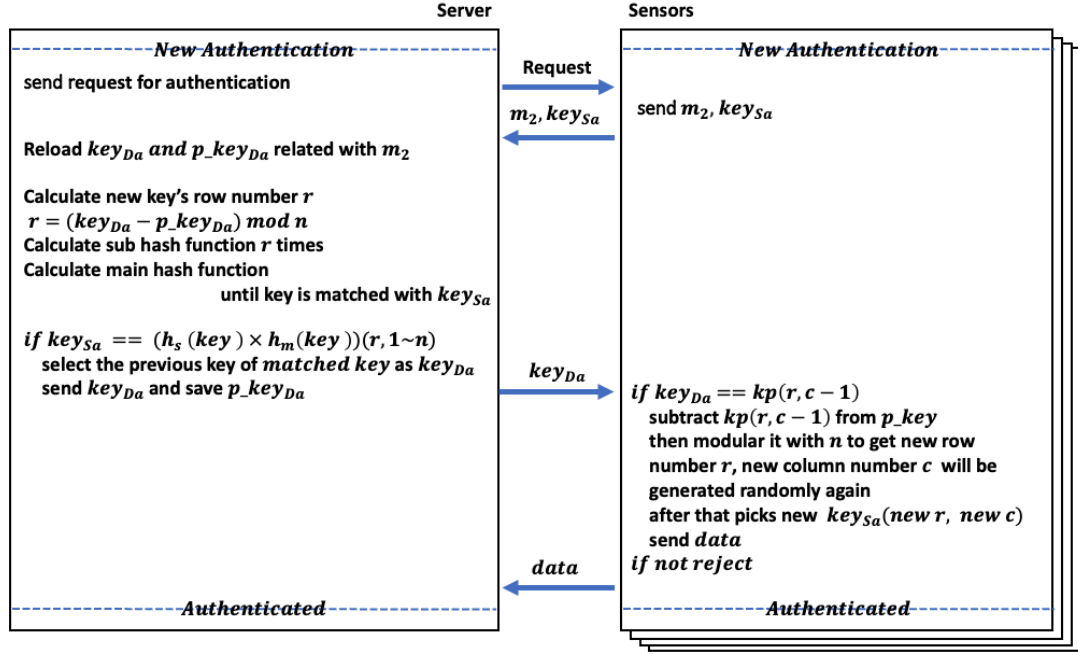


Figure 8. Simple Authentication

- (3) If key_{Sa} is matched with the calculated key which is $h_s(key) \times h_m(key)$ ($r, 1 \sim n$), the server selects the previous key of the matched key as key_{Da} and sends it to the sensor. In addition, the server saves p_key_{Da} , which is the previous key of key_{Da} , for the next authentication. If they are not matched, the server rejects this request.
- (4) The sensor receives key_{Da} from the server and compares key_{Da} with $kp(r, c - 1)$ which is the previous key of key_{Sa} . If they are matched, the sensor subtracts $kp(r, c - 1)$ from p_key which is the previous key of $kp(r, c - 1)$. After that, the sensor performs a modular operation with n to get the new row number r . The new column number c is generated randomly by the sensor. Thus, a new key_{Sa} is

selected as $(new\ r, new\ c)$. The sensor saves the new key key_{Sa} for the next authentication and sends data to the server.

- (5) For the new authentication, the server sends a new request to the sensor. The sensor sends m_2, key_{Sa} without n to the server. The server reloads key_{Da} and p_key_{Da} related to the sensor's m_2 . Afterwards, the sever performs the same procedure as the original authentication procedure.

ANALYSIS

In the analysis section, performance analysis and security analysis are described.

The proposed protocol is compared with other related schemes [12-15, 17].

Performance Analysis

The proposed protocol's performance analysis result is compared with other related protocols and the result is described in this section.

Table 2. Execution time for each operation [15]

Operations	Notation	Execution time (ms)
Modular Multiplication	T_M	0.027
ECC-based Multiplication	T_{SM}	0.304
ECC-based Addition	T_A	0.001
Exponentiation	T_E	0.297
Inversion	T_I	0.008
Hash Function	T_H	0.319
Bilinear Pairing	T_P	2.373
Random Number Generator	T_{RNG}	0

This authentication protocol is tested on MacBook Air, macOS Big Sur version 11.2.3, Apple M1 chip, and Memory 8GB. Sensor nodes are operated in Arduino ESP8266 Wi-Fi module. In the initial authentication phase, the server's total computation cost is 80ms. After the initial authentication phase, the server's total computation cost is reduced to 11ms.

Computation cost for the sensor and server and communication cost for the entire authentication are compared. The computation cost comparison is shown in Table 3.

However, due to hardware differences on each scheme and for accurate performance

comparison, Kumar et al.'s execution time list [15] is referred for each operation's execution time and this list is shown in Table 2. In [15], they indicate $T_M, T_{SM}, T_A, T_E, T_I, T_H, T_P$, and T_{RNG} are Modular multiplication, ECC-based multiplication, ECC-based addition, Exponentiation, Inversion, Hash function, Bilinear pairing, and Random number generator, respectively. Their respective computation times are 0.027ms, 0.304ms, 0.001ms, 0.297ms, 0.008ms, 0.319ms, 2.373ms, and 0ms. He, D *et al.* [12]'s protocol takes ECC-based multiplication, Bilinear pairing, and Hash function for authentication protocol. In their authentication, the sensor's computation takes 4 ECC-based multiplication and 1 hash function ($4T_{SM} + 1T_H$) which is approximately 1.535ms, and the server takes 4 ECC-based multiplication, 2 bilinear pairing, and 1 hash function ($4T_{SM} + 2T_P + 1T_H$), which is approximately 6.281ms. K. Sowjanya *et al.* [13]'s protocol takes ECC-based multiplication and Modular multiplication. Its computation takes 3 ECC-based multiplications and 1 modular operation ($3T_{SM} + 1T_M$) which is 0.939ms on the sensor and 6 ECC-based multiplication and 1 modular operation ($6T_{SM} + 1T_M$) which is approximately 1.851ms on the server. In M. Nikooghadam *et al.* [14]'s protocol, ECC-based multiplication, ECC-based addition, and Modular multiplication are used. This protocol's sensor takes 3 ECC-based multiplication and 2 modular operation ($3T_{SM} + 2T_M$) which is approximately 0.966ms, and the server takes 4 ECC-based multiplication and 2 ECC-based addition ($4T_{SM} + 2T_A$) which is 1.218ms. Kumar *et al.* [15]'s protocol takes ECC-based multiplication and Modular multiplication for their authentication. In [15], the sensor takes 3 ECC-based multiplication and 2 modular operation ($3T_{SM} + 2T_M$) which is 0.966ms and the server takes 5 ECC-based multiplication ($5T_{SM}$) which is approximately 1.851ms. In [17], the sensor only computes

Random number generator, and it is a significantly light and small computation.

Therefore, its computation time is considered to be 0ms. The server in this protocol takes ECC-based multiplication, ECC-based addition, and Hash function, and its computation time is approximately 26.782ms. The proposed protocol takes the same operations as [17] but it reduced the server's hash function computation time. As a result, total computation time on the server is reduced from 26.782ms to 11.47ms (48.2%). After the initial authentication, the server and sensor perform the simple authentication. Therefore, the server's hash function calculations are reduced more from 35 times to 16 times which means that the computation cost will be reduced from 11.47ms to 5.409ms (47.1%).

Table 3. Computation cost comparison

Scheme	Computation Cost (ms)	
	Sensor	Server
He, D <i>et al.</i> [12]	$4T_{SM} + 1T_H$ (1.535)	$4T_{SM} + 2T_P + 1T_H$ (6.281)
K. S <i>et al.</i> [13]	$3T_{SM} + 1T_M$ (0.939)	$6T_{SM} + 1T_M$ (1.851)
M. N <i>et al.</i> [14]	$3T_{SM} + 2T_M$ (0.966)	$4T_{SM} + 2T_A$ (1.218)
Kumar <i>et al.</i> [15]	$3T_{SM} + 2T_M$ (0.966)	$5T_{SM}$ (1.52)
Choi, S <i>et al.</i> [17]	$1T_{RNG}$ (0)	$1T_{SM} + 1T_A + 83T_H$ (26.782)
Proposed protocol	$1T_{RNG}$ (0)	$1T_{SM} + 1T_A + 35T_H$ (11.47)
After initial authentication	$1T_{RNG}$ (0)	$1T_{SM} + 1T_A + 16T_H$ (5.409)

Based on the result, this proposed protocol can provide a significantly lower computation burden for the sensor. Therefore, the sensors are not required to have high computation power and a big battery capacity. Moreover, in WBAN, we do not care about the server's computation cost because they assume the server will be given a powerful computation ability for authentication. Therefore, it does not matter even though 11.47ms and 5.409ms are quite a big number compared to other protocols.

For communication cost comparison, the proposed model is compared with other protocols [12-15] except [17] because [17]'s total message length (in bits) is the same as the proposed model. The result is shown in Table 4. In the result, the proposed model uses only 1280-bits length for the entire authentication procedure. However, K. Sowjanya *et al.* [13] uses 6672-bits length which means almost 5 times more than the proposed model. M. Nikooghadam *et al.* [14]'s protocol is the second lowest model, but it still uses 2.5 times more than the proposed model. After the initial authentication, communication cost is reduced from 1280-bits to 768-bits through the simple authentication. It is one eighth of K. Sowjanya *et al.* [13] and quarter of M. Nikooghadam *et al.* [14]'s protocol

Thus, this proposed model uses the lowest length of bits for the authentication protocol. As a result, the proposed model is the lightest-weight model communication cost wise as well as computation cost wise, especially on the sensor side.

Table 4. Communication cost comparison

Scheme	Length (in bits)
He, D <i>et al.</i> [12]	4288
K. S <i>et al.</i> [13]	6672
M. N <i>et al.</i> [14]	3264
Kumar <i>et al.</i> [15]	3440
Proposed protocol	1280
After initial authentication	768

Security Analysis

For [17]'s authentication protocol, ECC and two hash functions SHA-256 and SHA-512 are used for the WBAN authentication protocol. For this proposed protocol, the same authentication techniques are used as [17]'s scheme. Therefore, security analysis

results will be the same as the [17]’s work. A 256-bits key is used for ECC to provide a 128-bit security level which is the same as a 3072-bit key with RSA [21]. Hash functions SHA-256 and SHA-512 can provide 128-bit and 256-bit security level, respectively. Moreover, they are designed as non-reversible techniques; they are unbreakable if their keys are secure. For this protocol, the confidentiality of the hash functions’ keys can be guaranteed through ECC at a 128-bit security level. In addition, the hash functions’ round functions are modified to use a private key to be more secure. As a result, even if hash functions’ keys are exposed, the security level can be protected unless the private key is exposed.

Table 5. Security feature comparison

Security Feature	He, D <i>et al.</i> [12]	K. S <i>et al.</i> [13]	M. N <i>et al.</i> [14]	Kumar <i>et al.</i> [15]	Proposed protocol
Mutual authentication	X	O	O	O	O
Perfect forward security	O	O	X	O	O
Session key establishment	O	O	O	O	O
Anonymity	X	O	O	O	O
Non-repudiation	X	X	X	X	O
User impersonation attack	X	O	O	O	O
Server impersonation attack	X	O	O	O	O
Dos attack	X	O	O	X	O
KCI attack	X	X	O	O	O

The proposed protocol’s security features are compared with other related protocols [12-15]. Table 4 shows the result of the comparison. According to the result, He, D *et al.* [12]’s protocol provides only perfect forward security and session key establishment. K. Sowjanya *et al.* [13]’s protocol can provide mutual authentication, perfect forward security, session key establishment, anonymity, user impersonation attack immunity, server impersonation attack immunity, and Dos attack immunity. In M.

Nikooghadam *et al.* [14]'s protocol, they can provide mutual authentication, session key establishment, anonymity, user and server impersonation attack immunity, Dos attack immunity, and key compromise impersonation (KCI) attack immunity. Kumar *et al.* [15]'s protocol provides mutual authentication, perfect forward security, session key establishment, anonymity, user and server impersonation attack immunity, and KCI attack immunity. The proposed model can provide mutual authentication, perfect forward security, session key establishment, anonymity, non-repudiation, user and server impersonation attack immunity, Dos attack immunity, and KCI attack immunity. As a result, the proposed model can support more security features than other related protocols with significantly lower computation cost on the sensors.

CONCLUSION

The study for security of WBAN is quite considerable and expanding because WBAN has emerged as a promising technique for healthcare systems. WBAN can provide a remote and continuous monitoring system for the health care field through tiny and compact sensors [1]. The sensors are located on, in, and around patient's body to collect body function data. However, since this system uses a wireless network and the measured data is critical and life-related, we need to check whether the data is from rightly and securely identified sensors or not before any transaction begins. Therefore, we need to care about the security of WBAN to protect the data from any malicious attacks [2, 3]. In addition, the sensors are attached to patient's body for a long time, therefore, to make sensors that have lightweight computation and low battery consumption is another challenge of WBAN [5]. This paper proposed an improved secure and lightweight authentication protocol for WBAN with ECC and 2D hash chain techniques. The key selection algorithm is modified to use subtraction-modular operation to solve inefficient key usage problem and too many hash functions on the server problem. As a result, the efficiency of key usage is increased and the number of hash function calculations on the server are reduced while providing the same security level as the previous scheme.

In future work, investigation of a way to further reduce the server's computations is needed. The new key selection algorithm reduced the server's computations significantly. However, even though the server has powerful computation ability, I think it would be best to reduce the number of calculations even more, because the more calculations there are, the higher the probability of error. Therefore, this investigation makes this authentication protocol more stable, secure, and lightweight protocol.

Moreover, we need to research about packet loss. When the server or sensor send a packet and lost it, then they need to send the packet again to recover transaction.

However, in this case, a key is exposed, and it could make a chance for Man-in-the-middle attack. Therefore, we need to investigate about a way to protect keys from this attack.

LITERATURE CITED

1. F. A. Khan, N. A. H. Haldar, A. Ali, M. Iftikhar, T. A. Zia and A. Y. Zomaya, "A Continuous Change Detection Mechanism to Identify Anomalies in ECG Signals for WBAN-Based Healthcare Environments," in *IEEE Access*, vol. 5, pp. 13531-13544, 2017, doi: 10.1109/ACCESS.2017.2714258
2. Jovanov E, Milenkovic A, Otto C, De Groen P, Johnson B, Warren S, Taibi G. A WBAN System for Ambulatory Monitoring of Physical Activity and Health Status: Applications and Challenges. *Conf Proc IEEE Eng Med Biol Soc.* 2005; 2005:3810-3. doi: 10.1109/IEMBS.2005.1615290. PMID: 17281060.
3. Raskovic, D., Martin, T., & Jovanov, E. (2004). Medical monitoring applications for wearable computing. *The computer journal*, 47(4), 495-504.
4. Shen, J., Tan, H., Moh, S., Chung, I., Liu, Q., & Sun, X. (2015). Enhanced secure sensor association and key management in wireless body area networks. *Journal of Communications and Networks*, 17(5), 453-462.
5. Khanna, A., Chaudhary, V., & Gupta, S. H. (2018). Design and analysis of energy efficient wireless body area network (WBAN) for health monitoring. In *Transactions on computational science XXXIII* (pp. 25-39). Springer, Berlin, Heidelberg.
6. Saleem, S., Ullah, S., & Kwak, K. S. (2011). A study of IEEE 802.15. 4 security framework for wireless body area networks. *Sensors*, 11(2), 1383-1395.
7. Pathania, S., & Bilandi, N. (2014). Security issues in wireless body area network. *Int J Comput Sci Mobile Comput*, 3(4), 1171-8.
8. Shankar, S. K., Tomar, A. S., & Tak, G. K. (2015). Secure medical data transmission by using ECC with mutual authentication in WSNs. *Procedia Computer Science*, 70, 455-461.

9. Miller, V. S. (1985, August). Use of elliptic curves in cryptography. In Conference on the theory and application of cryptographic techniques (pp. 417-426). Springer, Berlin, Heidelberg.
10. Zargar, A. J., Manzoor, M., & Mukhtar, T. (2017). ENCRYPTION/DECRYPTION USING ELLIPTICAL CURVE CRYPTOGRAPHY. *International Journal of Advanced Research in Computer Science*, 8(7).
11. Pub, F. I. P. S. (2012). Secure hash standard (shs). *Fips pub*, 180(4).
12. He, D., Zeadally, S., Kumar, N., & Lee, J. H. (2016). Anonymous authentication for wireless body area networks with provable security. *IEEE Systems Journal*, 11(4), 2590-2601.
13. K. Sowjanya, M. Dasgupta, and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 129–146, 2020.
14. M. Nikooghadam and H. Amintoosi, "A secure and robust elliptic curve cryptography-based mutual authentication scheme for session initiation protocol," *Secure. Privacy*, vol. 3, no. 1, pp. 165–178, 2020.
15. Kumar, Mahender, and Satish Chand. "A Lightweight Cloud-Assisted Identity-Based Anonymous Authentication and Key Agreement Protocol for Secure Wireless Body Area Network." *IEEE Systems Journal* (2020).
16. Li, X., Peng, J., Kumari, S., Wu, F., Karuppiyah, M., & Choo, K. K. R. (2017). An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. *Computers & Electrical Engineering*, 61, 238-249.

17. Choi, S., Shin, S., Jin, X., & Shin, S. (2020, October). Secure and low computation authentication protocol for Wireless Body Area Network with ECC and 2D hash chain. In Proceedings of the International Conference on Research in Adaptive and Convergent Systems (pp. 130-135).
18. Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). Guide to elliptic curve cryptography. Springer Science & Business Media.
19. Syamsuddin, I., Dillon, T., Chang, E., & Han, S. (2008, November). A survey of RFID authentication protocols based on hash-chain method. In 2008 Third International Conference on Convergence and Hybrid Information Technology (Vol. 2, pp. 559-564). IEEE.
20. Chaves, R., Kuzmanov, G., Sousa, L., & Vassiliadis, S. (2006, October). Improving SHA-2 hardware implementations. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 298-310). Springer, Berlin, Heidelberg.
21. Kerry Maletsky. Rsa vs ecc comparison for embedded systems. White Paper, Atmel, 5, 2015.