



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Doctor of Philosophy

A Study on Secret Key Generation Using Biosignals



Interdisciplinary Program of Information Security

The Graduate School

Pukyong National University

February 2019

A Study on Secret Key Generation Using Biosignals

생체신호를 이용한 비밀키 생성에 관한 연구

Advisor: Prof. Sang-Uk Shin

by
Juyoung Kim



A thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in Interdisciplinary Program of Information Security,
The Graduate School,
Pukyong National University

February 2019

A Study on Secret Key Generation Using Biosignals

A dissertation

by

Juyoung Kim

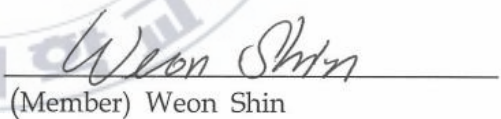
Approved by:



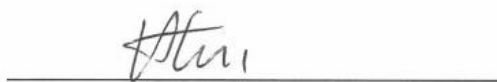
(Chairman) Kyung-Hyune Rhee



(Member) Chang-Soo Kim



(Member) Weon Shin



(Member) Taek-Young Youn



(Member) Sang-Uk Shin

February 22, 2019

Contents

List of Figures	iii
List of Tables	v
Abstract	vi
Chapter 1. Introduction	1
1.1 Motivation	1
1.2 Contributions and Organization of the Thesis	2
Chapter 2. Preliminaries	6
2.1 IPI features	6
2.2 Fuzzy Commitment	7
2.3 Fuzzy Vault	8
Chapter 3. Biometric Secret Key Generation using Seed Piece Pool	11
3.1 Vulnerability of Fuzzy Vault	11
3.1.1 Attacking Fuzzy Vault	11
3.1.2 Attack Scenario	12
3.1.3 Experimental Environment	14
3.1.4 Result	15
3.2 Vulnerability of Fuzzy Commitment	17
3.2.1 Peak Misdetection	18
3.3 Seed Piece Pool Based Key Generation Method	19
3.3.1 Seed Piece Error Correction	20
3.3.2 Peak Misdetection Recovery	21
3.3.3 IPI Rearrangement	22
3.3.4 IPI Selection	23
3.3.5 Seed Piece Pool	23
Chapter 4. BKG System Design and Implementation	26
4.1 System Overview	26
4.1.1 BKG System structure	27

4.1.2 Software Block	29
4.2 Biometric Secret Key Generator	32
4.2.1 Biometric Information Collection Block	33
4.2.2 Seed Piece Pool Management Block	33
4.2.3 Biometric Secret Key Generation Block	38
4.3 Security Protocol Simulator for Biometric Secret Key Verification	40
4.3.1 Security Protocol Establishment Block	41
4.3.2 User Interface Block	42
Chapter 5. Experiment	45
5.1 IPI Entropy Test by LED Wavelength of PPG Sensor ...	45
5.1.1 IPI Data Collection	45
5.1.2 AIS.31 Test	47
5.1.3 Result	48
5.2 Entropy Verification of Seed Piece Data	50
5.2.1 Experiment	50
5.2.2 Result	52
5.3 FAR/FRR Test	60
5.3.1 FAR Test	60
5.3.2 FRR Test	60
5.4 Biometric Secret Key Randomness Test	61
5.4.1 Entropy Test Results By Number of Seed Pieces	62
5.4.2 Correlation Between the Number of Seed Pieces and the Randomness	64
5.5 Biometric Secret Key Generation Time Test	70
5.5.1 Experimental Environment	70
5.5.2 Result	70
5.6 IPI Recovery Test	71
5.6.1 Entropy Test	71
5.6.2 Number of Recovery	73
Chapter 6. Conclusion	74
Reference	76

List of Figures

Figure 1.1 Principle of PPG sensor	2
Figure 2.1 PKSA protocol	9
Figure 3.1 Correlation attack scenario	13
Figure 3.2 Similar PPG signal generation results using Kalman filter algorithm	14
Figure 3.3 Vault data and predicted PPG signals	16
Figure 3.4 Fuzzy commitment key generation	17
Figure 3.5 IPI error due to peak misdetection	19
Figure 3.6 Error correction for synchronizing seed piece between BKGs	21
Figure 3.7 IPI segmentation and integration	22
Figure 3.8 IPI rearrangement	23
Figure 3.9 Key generation procedure using seed piece	24
Figure 4.1 BKG system overview	26
Figure 4.2 BKG system operation structure	28
Figure 4.3 Block of biometric secret key generator system	30
Figure 4.4 Identify IPI using Bloom Filter	36
Figure 4.5 Function of seed piece pool	37
Figure 4.6 Seed derivation function	39
Figure 4.7 Data encryption and integrity verification data generation and verification process	41
Figure 4.8 User interface	43
Figure 4.9 BKG and encryption process monitoring	44
Figure 5.1 0,1 distribution by bit	47
Figure 5.2 Biometric secret key generation simulator test	

configuration	51
Figure 5.3 Monobit test results	53
Figure 5.4 Poker test results	54
Figure 5.5 Run test fail result	55
Figure 5.6 Run test pass result	55
Figure 5.7 Autocorrelation test result	57
Figure 5.8 Biometric secret key randomness test results	69



List of Tables

Table 3.1 Correlation attack test result	16
Table 4.1 Block summary of BKG system	31
Table 4.2 Gray Code	33
Table 4.3 Seed length according to hash function	40
Table 5.1 Multi-wavelength LED sensor specifications	46
Table 5.2 AIS.31 Test	48
Table 5.3 Entropy test result by LED wavelength (P:Pass F:Fail)	49
Table 5.4 IPI of first and second derivative value AIS.31 test result	51
Table 5.5 Long Run test result	56
Table 5.6 Ubpulse 340 sensor specifications	58
Table 5.7 Seed piece entropy test results	59
Table 5.8 FAR Test result	60
Table 5.9 FRR Test result	61
Table 5.10 Entropy test results by number of seed pieces	63
Table 5.11 IPI generation time test result	71
Table 5.12 Result of entropy test of corrected seed piece	72
Table 5.13 Number of IPI corrections	73

생체신호를 이용한 비밀키 생성에 관한 연구

김 주 영

부경대학교 대학원 정보보호학협동과정

요 약

헬스케어 시장이 확대됨에 따라 개인 생체정보의 중요성이 대두되고 있다. 특히 인슐린 펌프와 같은 임플란트 디바이스가 외부에서 공격당할 경우 사용자의 생명에 치명적일 수 있다. 따라서 임플란트 디바이스가 안전하게 외부와 통신하기 위해 비밀키를 생성할 수 있는 방법이 필요하다. 사전에 키 공유 없이 임플란트 디바이스와 통신하기 위해 생체 신호를 이용할 수 있다. 생체 신호 중 PPG 신호는 신체 내외부에서 측정할 수 있고 개개인마다 고유한 특성을 가지고 있다. PPG 신호를 이용해 비밀키를 생성하는 대표적인 방법으로는 fuzzy vault와 fuzzy commitment가 있다. 그러나 fuzzy vault는 상관관계 공격에 취약하고 fuzzy commitment는 항상 PPG 신호를 측정하고 있어야 비밀키를 생성할 수 있다. 본 논문에서는 fuzzy commitment의 시드 조각폴을 도입하여 기존에 측정된 PPG 신호를 이용해 비밀키를 생성할 수 있는 방법을 제안한다. 제안한 방법을 토대로, 제안한 방법을 토대로 생성된 생체 비밀키의 사용 가능성을 확인하기 위해 엔트로피 테스트 및 생성 시간 테스트 등의 실험을 진행하였다.

Chapter 1. Introduction

1.1 Motivation

As the market for healthcare and medical devices expands the importance of personal biometric information security increases. At the 2012 RSA conference, McAfee hacker Barnaby Jack demonstrated hacking of an insulin pump, which is an implant device. The demonstration involved the remote control of equipment inside the body, which can have lethal consequences for patients with an implanted insulin pump. Moreover, biometric information is related to the life of users, which may be very sensitive to leaked personal information. Therefore, the equipment that stores biological information requires a highly stable security technique. An implant device is a device inserted inside the body that can communicate with external systems when an update is required for enhancements, or to address software vulnerabilities. Thus, secure communication between implant devices is required, as they can be targeted by malicious hackers. For secure communication with the implant, it is necessary to share the secret key between the devices. However, when a secret key is inserted into an implant device in advance, the secret key may be leaked. Therefore, a method for secure communication

without prior secret sharing is necessary; in addition to the generation of a one-time secret key using biosignals that can be measured both inside and outside of the body with similar measurement values, regardless of the measurement positions. Moreover, the measured biosignals should be unique to each individual and exhibit randomness. Implantable devices therefore require a method for generating one-time secret keys using biosignals, which satisfies the above-mentioned criteria for secure communication.

1.2 Contributions and Organization of the Thesis

A photoplethysmogram (PPG) signal is obtained by measuring the blood flow of the blood vessel using the light source (a light-emitting diode (LED)) and the light detector charge-coupled device (CCD), as shown in Figure 1.1.

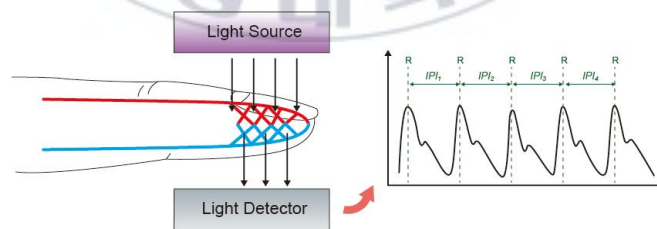


Figure 1.1 Principle of PPG sensor

When PPG signals are measured from inside and outside the body, the measured values of the inter-pulse intervals

(IPI) are similar. The IPI is also unique to each individual, and it is suitable for the generation of secret keys [1]. In addition, if the PPG signal is leaked, it can be replaced by a new PPG signal, which requires an additional measurement. This property of biosignals is suitable for the generation of keys for external communication with implants. To generate a key using the PPG signal, the following must be considered:

- (1) The entropy of the IPI: the entropy criterion of the IPI must be satisfied to generate a key. If the entropy is excessively high, the false rejection rate (FRR) may be high, and the key generation may be difficult to achieve. Therefore, the IPI interval with appropriate entropy should be determined
- (2) Key generation time: a significant amount of time is not required for the generation of a secret key. It should be possible to minimize the time required for the key generation by correcting the IPI error measurement and selecting an appropriate number of IPIs for the key generation.
- (3) The randomness of the generated keys: the keys generated using IPIs should guarantee randomness.

The IPIs are classified into high entropy segments and similar segments. A high similarity simplifies the process of generating secret keys, whereas a lower security strength and higher entropy increases the security strength of the secret keys, which makes them difficult to generate. Therefore, it is necessary to extract segments with a certain similarity among IPIs, to create a large set of IPIs that satisfy the entropy criterion. Accordingly, an appropriate number and interval of IPIs should be defined for the generation of secret keys. Moreover, given that the number of IPIs used to generate a secret key is proportional to the time required for its generation, it is necessary to provide an alternative method for the efficient generation of secret keys.

This thesis introduces the characteristics of PPG signals and how they are used to generate secret keys. In addition, problems from previous studies are identified using experiments, and improved key generation methods using IPIs are proposed. Chapter 2 describes the related research on the generation of keys using signals, and Chapter 3 identifies problems related to key generation methods using existing biosignals, which are demonstrated using experiments. In addition, an improved method for key generation using seed pools is proposed. In Chapter 4, a discussion on the design and implementation of the proposed seed pool key generation simulator is presented. In Chapter 5, the IPI entropy

verification, false acceptance rate (FAR)/FRR measurements, validation of the randomness of the generated keys, and IPI recovery method test are discussed, in addition to the results. The conclusions are then presented in Chapter 6.



Chapter 2. Preliminaries

Several studies have been conducted on secure communication using biosignals. Among these, the use of heartbeats has been investigated; in addition to the generation of keys using electrocardiograms (ECGs) and PPG signals, which are heartbeat signals, for user authentication. The proposed key generation method can be divided into two categories. One is the fuzzy vault method that generates a key using the entire bio-signal, and the other is the fuzzy commitment method that extracts a feature point from a heartbeat signal.

This chapter describes the related studies on key generation using biosignals and IPI characteristics.

2.1 IPI features

The IPI is one of the feature points extracted from a heartbeat signal, and it has characteristics that are unique to each individual [1]. The heartbeat signal has 14 feature points in one cycle, and it is difficult to extract all 14 feature points due to their significant variation [2]. Zhang et al. [3] revealed that ECG signals, similar to PPG signals, exhibit randomness. However, unlike PPG, ECG requires electrodes, which makes measurements more difficult to perform. Rushanan et al. [4]

defined the security requirements for the communication between implant devices and human networks, and Mohana et al. [5] verified the randomness of IPIs. Baga et al. [6] reported that signals such as ECG or EEG signals can be leaked to simple contacts such as handshakes , and that the IPI of the other party can be estimated with a probability of approximately 30%.

2.2 Fuzzy Commitment

Fuzzy commitment was the first key generation protocol used in body sensor networks [7]. For a fuzzy commitment process using biological signals, nodes A and B share the error correction code parameters in advance. The nodes then measure the biosignals within a given time-period. Thereafter, node A computes the secret s to be delivered to node B and hashes it with the following hash function [4]:

$$Com = f(x, s) = (h(s), x \oplus s)$$

Node B performs a $x' \oplus (x \oplus s)$ operation using the measured bio-signal x' of its own received commit. Moreover x' and x are not equal, although they have similar values. Therefore, s' generated by $x' \oplus (x \oplus s)$ contains an error. s' corrects the error using a previously defined error correcting code parameter. The extracted s' and s are confirmed by the hash function as

equal. $h(s) = h(s')$ means s that is transmitted safely [7].

Several studies have been conducted on the application of fuzzy commitment. Poon et al. [8] described the wireless body sensor network (WBSN) for use in telemedicine services. They proposed a way to extract 128-bit binaries from ECG and PPG and use them for network communications. And they used the Hamming distance to correct IPI errors.

Rostami et al. [9] a method of ECG authentication for implant medical devices and external medical devices. They used the fuzzy commitment method and confirmed that the lower 4 bits of the IPI passed the NIST random test.

Cherukuri et al. [10] proposed a protocol for human body sensor network communication and proposed an alternative bio-signal that can be used in conjunction with heartbeat signals to enhance randomness. Pirbhulal et al. [11] proposed a key generation method using the averages of deviation after collecting N ECGs, which is different from the existing fuzzy commitment and fuzzy vault methods.

2.3 Fuzzy Vault

Fuzzy vault is a scheme for authenticating biometric data without storage. It is mainly used for fingerprint recognition. Venkatasubramanian et al. [12] proposed a physiological signal-based key agreement (PSKA) based on a fuzzy

vault-based key agreement scheme. The PKSA protocol is summarized in Figure 2.1 below.

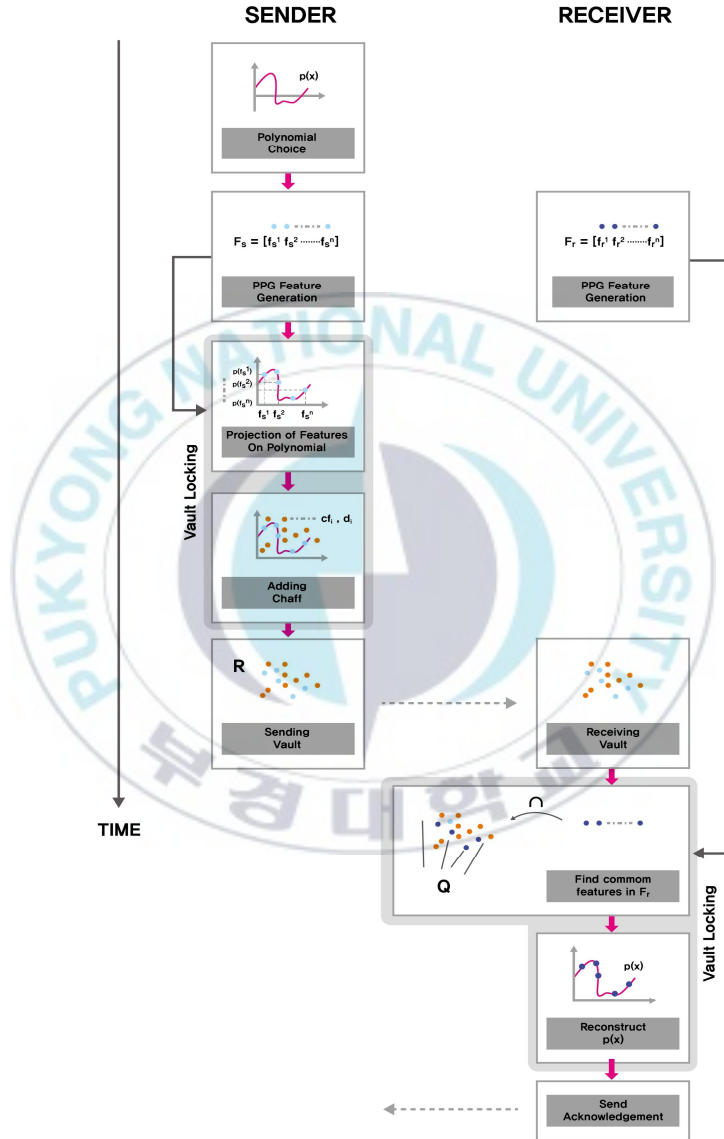


Figure 2.1 PKSA protocol

Sensor nodes share degrees of polynomials. The sender

and receiver then collect the PPG signal, and the sender extracts feature points from the collected PPG signal. Based on the extracted feature points, the sender generates polynomials and mixes feature points and chaff points to create vaults. The vault generated by the sender is passed to the receiver, and the receiver extracts the feature points from its PPG signal to determine the coefficients of the polynomial. Finally, the sender and the receiver check whether the polynomial matches using the MAC.

Chunqiang et al. [13] proposed an ordered physiological feature-based key agreement (OPKA) protocol, which is a Lagrangian interpolation-free protocol that PSKA uses to calculate secret sharing.

Kalai et al. [14] proposed a way to reduce communication costs using linear prediction coding. However, it is necessary to presume that the biosignals can be predicted using previous biosignals.

Chapter 3. Biometric Secret Key

Generation using Seed Piece Pool

There are several problems associated with the fuzzy commitment and fuzzy vault methods. In this chapter, a discussion on the problems associated with the two methods is presented, and a biometric secret key generation method using seed piece pools is proposed.

3.1 Vulnerability of Fuzzy Vault

The fuzzy vault scheme is vulnerable to correlation attacks using biometric data that is similar to the original biometric data. Therefore, the fuzzy vault system using PPG signals may be vulnerable to correlation attacks. Moreover, a correlation attack was conducted to fully identify the fuzzy vault vulnerabilities.

3.1.1 Attacking Fuzzy Vault

The typical methods of fuzzy vault attacks are the brute force attack [15] and correlation attack [16]. For a brute force attack that is carried out on fuzzy vault using fingerprint data, the attack complexity is presented below [17].

$$Complexity = \frac{{}_r C_{k+1}}{{}_n C_{k+1}}$$

In the above equation, r is the number of chaff points, k is the degree of the polynomial, and n is the real minutiae, and $k + 1$ feature points are selected to regenerate the polynomial. The fuzzy vault, which was tested in this study, had 500 chaff points, a polynomial to the 35th degree, $n = 36$, and the complexity was approximately 1.07583×10^{55} . This complexity was very high for the release of the fuzzy vault due to a brute force attack. Correlation attacks refer to the method by which an attacker obtains hidden biometric information from two vaults generated from the same biometric information using different chaff points. In this section, the generation of a prediction signal from the original PPG signal is discussed, in addition to its application in the testing of the correlation attack that releases the vault.

3.1.2 Attack Scenario

Figure 3.1 presents the correlation attack performed in this study.

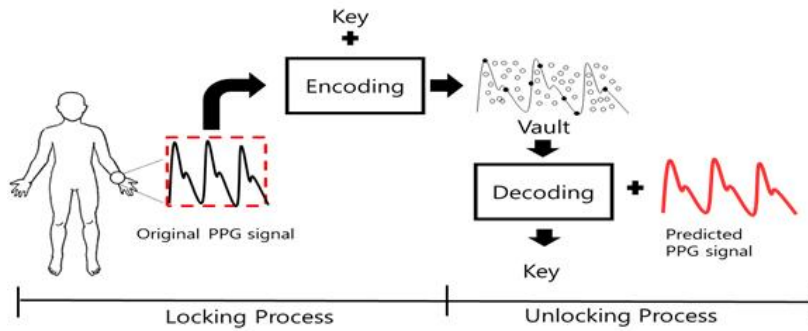


Figure 3.1 Correlation attack scenario

The PPG signal was collected by the ultra-wideband (UWB) or a similar frequency band, and a similar signal was then generated using the Kalman filter algorithm for the collected PPG signal. The Kalman filter algorithm uses the measured values and weights of the signal to produce predictions. To use the Kalman filter algorithm, the definition of the system model is required. In this study, a velocity-distance model that was similar to the PPG signal was applied. The prediction signal extracted from the Kalman filter is presented in Figure 3.2

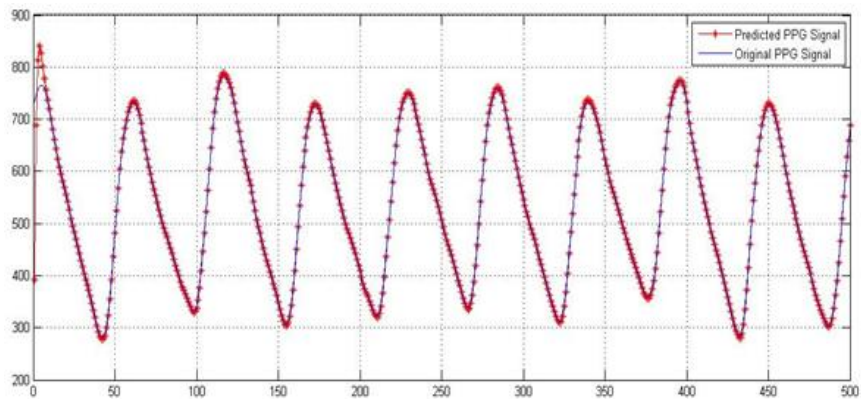


Figure 3.2 Similar PPG signal generation results using Kalman filter algorithm

The actual generated signal is very similar to the actual signal; however, it is not identical, as it contains the error. This is because the security service may not use a previously used PPG signal. The generated similar signal is then used to release the vault. The overall attack scenarios are summarized below.

- (1) Collect the original PPG signal.
- (2) Generate similar PPG signals using the Kalman filter algorithm of the original PPG signal.
- (3) Create a vault using the original signal.
- (4) Release the vault using a similar signal.

3.1.3 Experimental Environment

The data used in the experiments were sampled at 120 Hz using the MIT PsybioBank mimic2 dataset [18]. From this dataset, a PPG signal over a time-period of 1 h was extracted to generate 500 similar signals from 500 original points. As shown in Figure 3.2, a vault containing 500 chaff points was then created using 36 arbitrary points from the first signal to the 500th signal of the PPG signals. The next step was to attempt to release the vault using a simulated PPG signal with a set of 500 points. Finally, 36 points were randomly extracted from the 501st signal; thus, a total of 49500 iterations were repeated to determine the section where the vault was released. Experiments were conducted in two cases: CASE A, in which a correlation attack was performed using the PPG signal predicted from the PPG signal of the user; and CASE B, in which a correlation attack was carried out using the PPG signal of another user.

3.1.4 Result

As shown in Table 3.1, CASE A demonstrated a probability of 7.8% due to the release of the vault 35,292 times from 449,002 points; whereas in Case B, the vault was released 12 times from 449,002 points, yielding a 0.0026% probability.

Table 3.1 Correlation attack test result

	Total PPG Signal Point	Pass	Probability of Unlock
CASE A	449,002	35,292	7.8%
CASE B	449,002	12	0.0026%

The section where the vault was released was plotted as shown in Figure 3.3; where o is a vault, * is an original point, and \square is a similar point. In Figure 3.3, it can be seen that the original point matches the similar point.

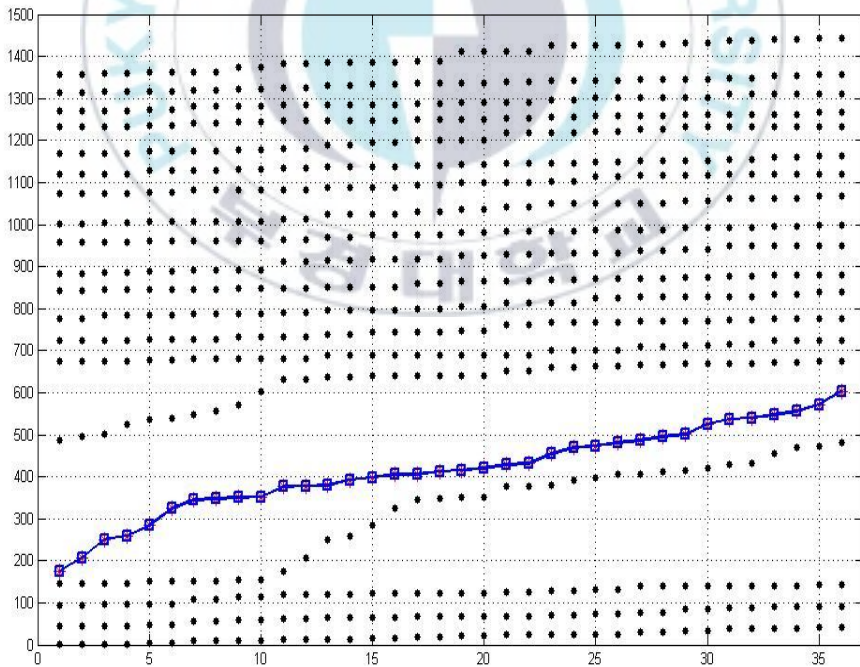


Figure 3.3 Vault data and predicted PPG signals

The fuzzy vault results reveal that if a PPG signal is leaked, it may be vulnerable to a correlation attack in some sections of the PPG signal.

3.2 Vulnerability of Fuzzy Commitment

The fuzzy commitment method involves the generation of the collected IPI through the Bose–Chaudhuri–Hocquenghem (BCH) encoding, as shown in Figure 3.4, and the sharing of the parity bit to match the IPI. Therefore, the PPG sensor cannot generate a key using the BCH code if the IPI value is outside the correctable error range due to peak misdetection.

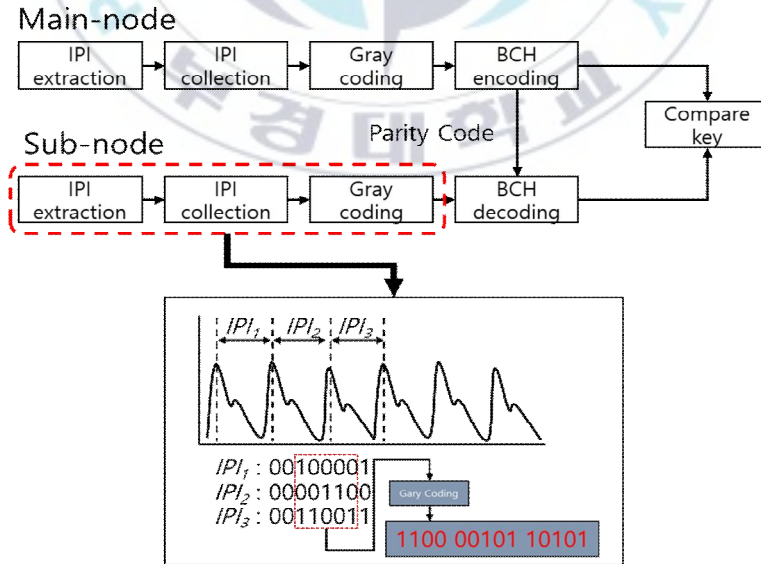


Figure 3.4 Fuzzy commitment key generation

If misdetection occurs, additional synchronization messages are required, given that the measurement point requires synchronization. Thus, the PPG sensor cannot generate a key using the BCH code if the IPI value is outside the correctable error range due to peak misdetection.

3.2.1 Peak Misdetection

Peak misdetection occurs due to human errors and other environmental factors. Figure 3.5 reveals that there is a problem associated with the synchronization of the IPI due to peak misdetection. This can affect the rate of continuous key generation.

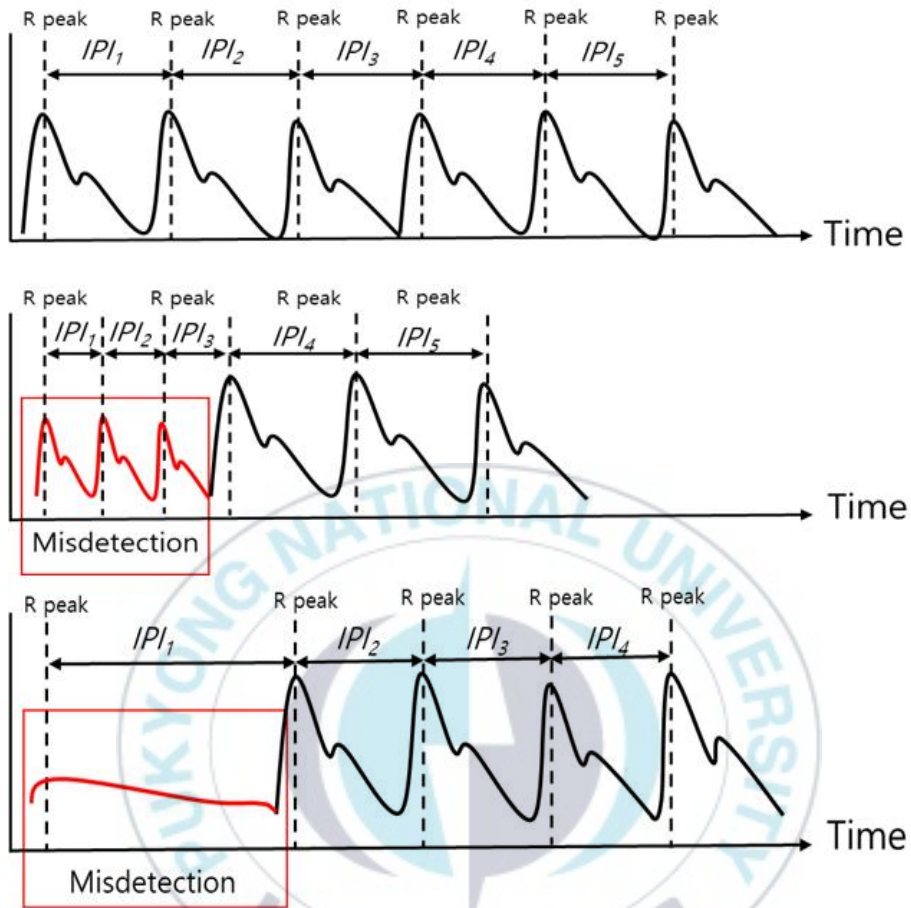


Figure 3.5 IPI error due to peak misdetection

3.3 Seed Piece Pool Based Key Generation Method

The aforementioned fuzzy vault and fuzzy commitment methods are vulnerable. In particular, fuzzy vault has more computation and memory requirements than fuzzy commitment, and its half total error rate (HTER), which is a measure of

biometric systems, is also higher than that of fuzzy commitment [19]. Therefore, the key generation method based on fuzzy commitment was used. However, fuzzy commitment can be used for IPI aggregation for a sufficient amount of time; and for the generation of a key only when the IPI can be measured (online). Moreover, there is a problem associated with the key generation efficiency due to peak misdetection.

In this section, a method is proposed to overcome the limitations of fuzzy commitment.

3.3.1 Seed Piece Error Correction

A seed is required to generate keys in the biometric key generator (BKG), and a seed is composed of multiple seed pieces. Several IPIs are required to generate one seed piece in the BKG, and BCH (Bose–Chaudhuri–Hocquenghem) codes are used as error correction codes to correct the discrepancy data between IPIs. Moreover, BKG₁ only transmits the parity code obtained after the BCH (n, k, t) encoding to BKG₂, as shown in Figure 3.6, to securely obtain the same IPI as BKG₂ without exposure. Furthermore, BKG₂ performs BCH decoding using the collected IPIs and received parity codes. If the number of mismatched bits is less than or equal to t, BKG₂ has the same IPI as the IPI collected by BKG₁.

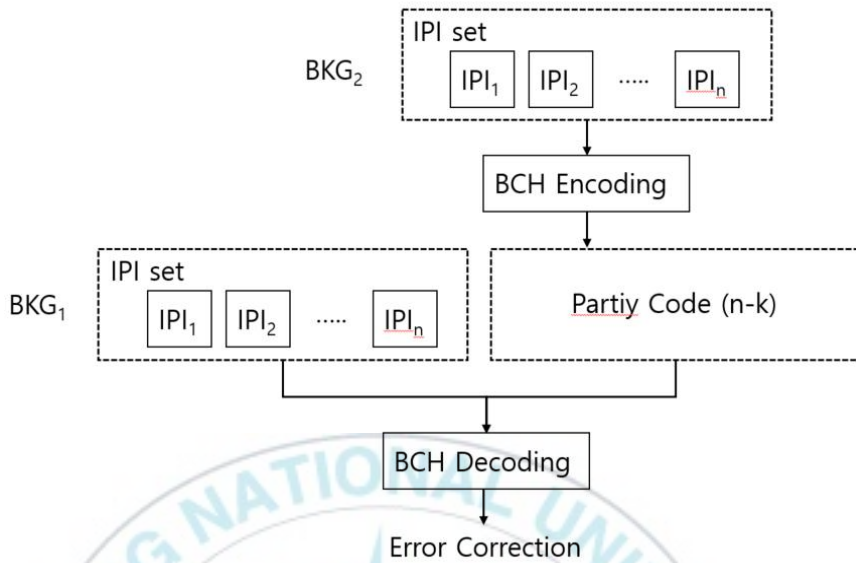


Figure 3.6 Error correction for synchronizing seed piece between BKGs

3.3.2 Peak Misdetection Recovery

It is often the case that the peak is erroneously detected due to the measurement noise, and the IPI value exceeds the normal range.

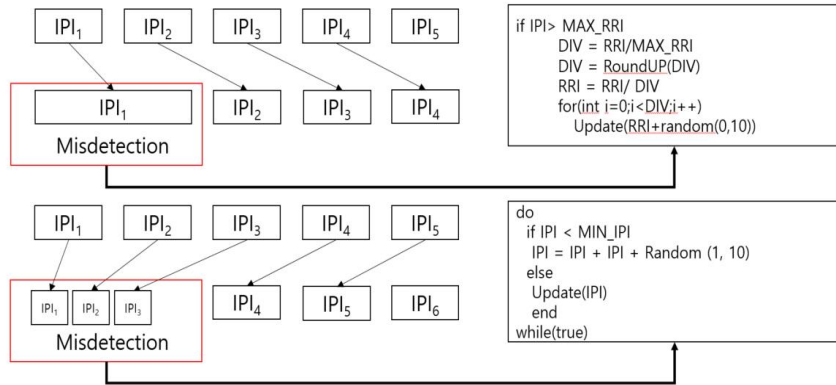
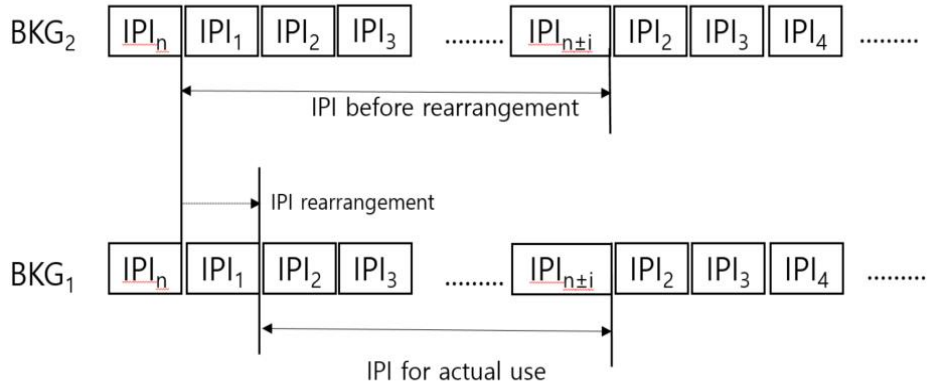


Figure 3.7 IPI segmentation and integration

To solve this problem, if the measured IPI is below the minimum threshold value as shown in Figure 3.7, the integration proceeds. If the IPI exceeds the maximum threshold value, the partitioning proceeds. When dividing and merging, random numbers are used to minimize the influence of entropy on the generated IPI.

3.3.3 IPI Rearrangement

The start times of the IPI acquisition for the same seed piece may be inconsistent due to the error between the transmission and reception points of two BKGs. At this instant, the seed piece value between BKGs may change. To mitigate the inconsistency of the collection start time, the IPI value is rearranged by delaying the used IPI interval by up to i times, as shown in Figure 3.8.



[Figure 3.8] IPI rearrangement

3.3.4 IPI Selection

It is very difficult to obtain an identical IPI due to the environment, posture, light leakage, and noise; even if the IPI is simultaneously obtained from two other parts of the same body. A method for the selection of matching information among IPIs measured by two BKGs was therefore required. Hence, a Bloom Filter, which uses a one-way hash function, was introduced to cryptographically secure the synchronization of seed pieces between two BKGs. The Bloom filter output (BFO) for each collected IPI transferred between the BKG and BFO was used to select only the same IPI collected by the two BKGs.

3.3.5 Seed Piece Pool

Fuzzy commitment could not generate keys when the IPI was offline (not measured). To improve this, a method was

proposed to generate seed pieces and update the seed piece pool by collecting error-corrected IPI values in real time in the online environment, and to generate seeds using a generated seed piece pool in the offline environment [20]. The key generation process using the seed piece pool is shown in Figure 3.9. First, metadata was created using a Bloom filter from an IPI set that was error-corrected and measured continuously. Next, the seed piece generation was completed by exchanging metadata with other BKGs, and filtering only the IPIs that matched with each other.

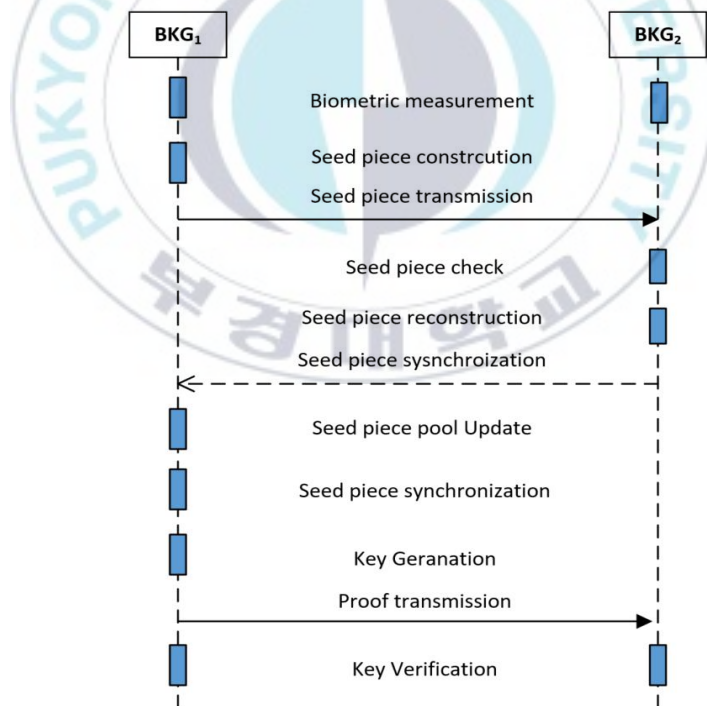
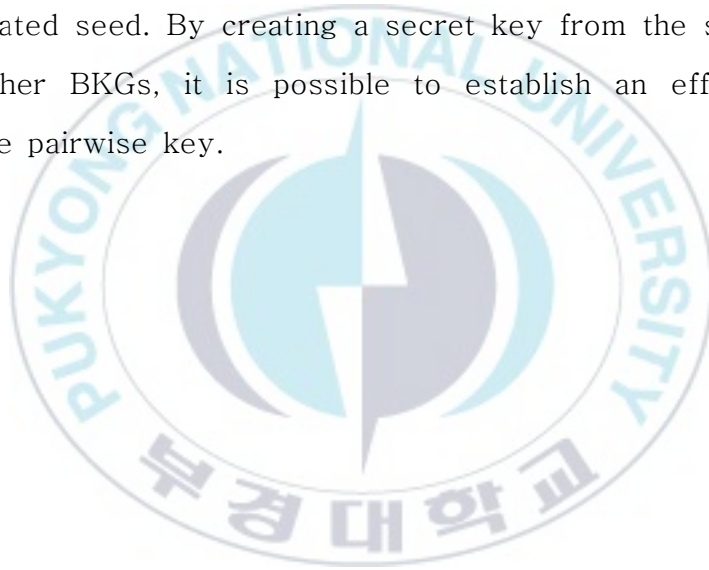


Figure 3.9 Key generation procedure using seed piece

Seed piece pools consist of a number of up-to-date seed pieces, and they can generate seeds in off-line situations wherein PPG sensors are not functional due to unexpected problems. When updating the seed piece pool, verify the seed pieces using the session key. The session key uses the hashed value of the seed piece in the current session.

A seed is generated using a seed piece obtained from a seed piece pool, and a session key is established using the generated seed. By creating a secret key from the same seed as other BKGs, it is possible to establish an efficient and secure pairwise key.



Chapter 4. BKG System Design and Implementation

In this study, a BKG system was designed and implemented to verify the efficiency and security of the biometric secret key generation method based on the seed piece pool. In this chapter, a description of the design of the BKG system is presented.

4.1 System Overview

The BKG system consists of a biometric secret key generation module and a security protocol simulator for biometric secret key verification, as shown in Figure 4.1.

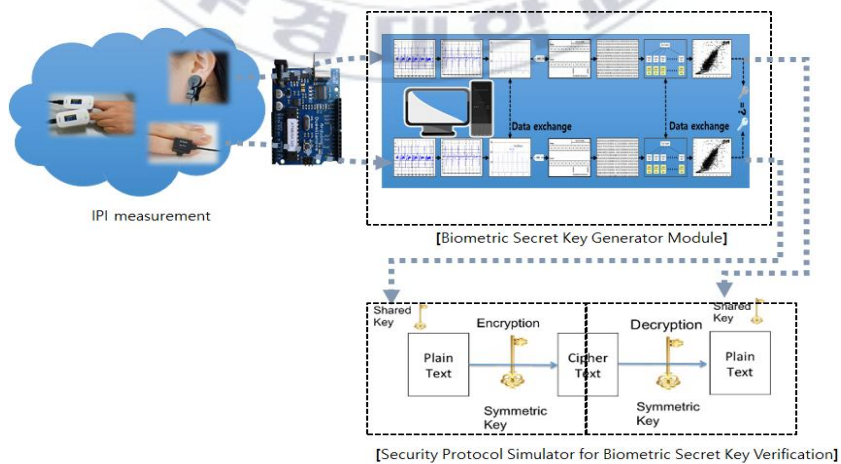


Figure 4.1 BKG system overview

The biometric secret key generation module has the following functions.

- (1) Measurement and collection of IPI data
- (2) Conversion of IPI data of a certain size into seed pieces
- (3) Update of seed piece to seed piece pool
- (4) Biometric secret key generation

The security protocol simulator verifies the biometric secret key generated from the biometric secret key generation module and proceeds with data encryption. This section describes the main functions and software configuration of the BKG system.

4.1.1 BKG System structure

The biometric secret key generated from the biometric secret key generation module is verified by the security protocol simulator for biometric secret key verification. The biometric secret key verification security protocol simulator performs validation of the generated biometric private key and data encryption.

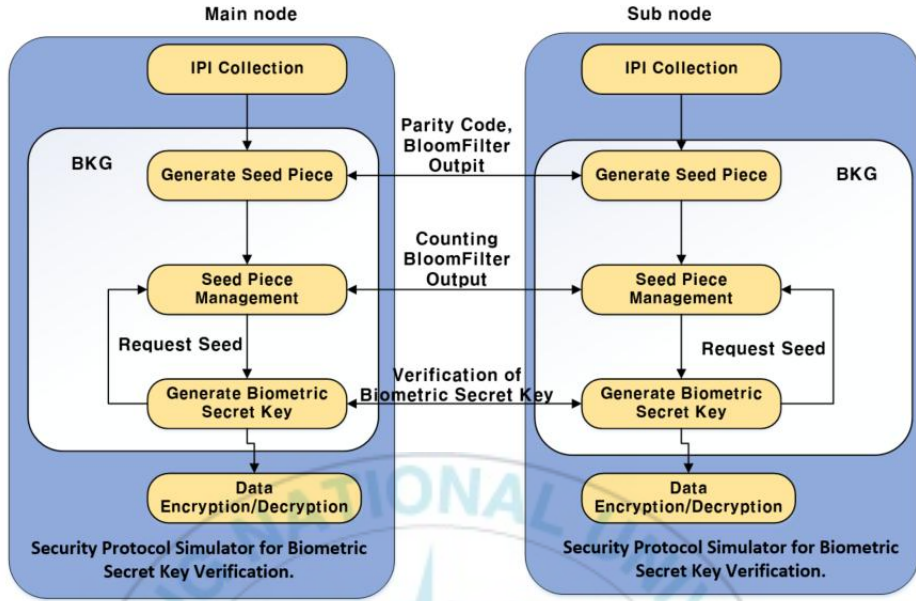


Figure 4.2 BKG system operation structure

Figure 4.2 presents the operational structure of the BKG system. The IPI measured by the PPG sensor is transmitted to each BKG. The biometric secret key generator is divided into a main and sub-nodes. The main node generates the BFO of the collected IPI and encodes it using the BCH error correction code. The BFO generated from the main node and the parity code is delivered to the sub-nodes. The sub-node then recovers the received parity code using the collected IPI, and checks the BFO to see if the recovered value matches the main node. Thereafter, the same IPI value is selected and updated to the seed piece pool, and the metadata of the seed piece pool is transmitted to the main node. The main node synchronizes the seed piece pool

by verifying the received seed piece pool metadata. The metadata of the seed piece pool is the counting BFO, which is discussed in the next section. When a biometric secret key generation request is received, the BKG generates a key using data from the seed piece pool, and transmits the current state of the seed piece pool to the corresponding nodes to synchronize the seed piece pool. A sub-node can be operated by multiple nodes. The biometric secret key generated from the biometric secret key generation module is verified by the security protocol simulator. The test data is encrypted by the simulator and compared with the original data to check whether a secure channel is formed.

4.1.2 Software Block

The BKG system consists of a biometric secret key generator module and a security protocol simulator for the verification of the biometric secret keys, as shown in Figure 4.3.

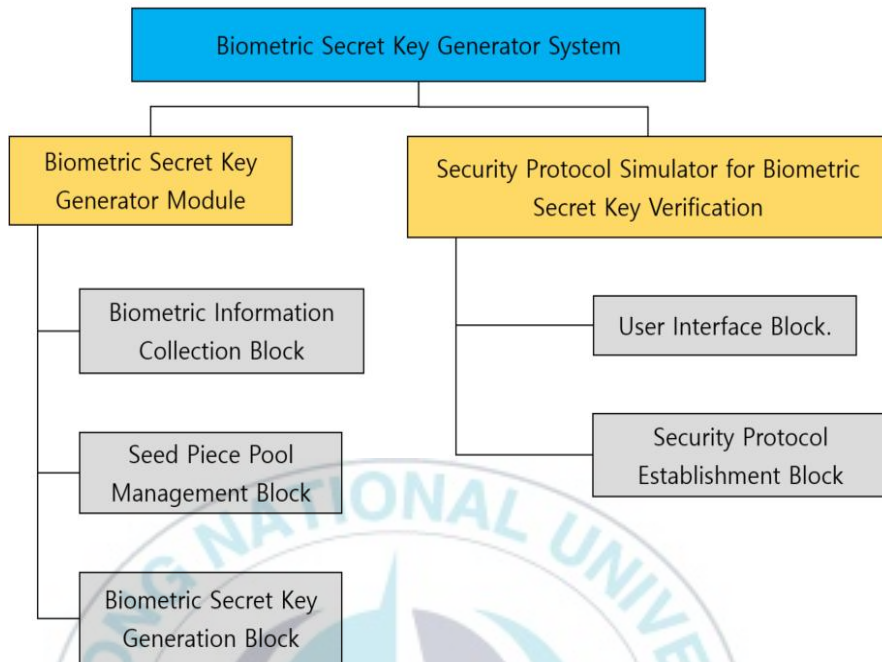


Figure 4.3 Block of biometric secret key generator system

The functions performed by each block are summarized in Table 4.1 below. The biometric information collection block supports two PPG sensors: a PPG sensor with multi-wavelength LEDs and a commercial PPG sensor (Ubpulse 340). The seed piece pool management block is driven based on the initial value set from the user interface block. The biometric secret key generation block provides an option to generate the biometric secret key automatically and manually, and it can encrypt the data using the biometric secret key generated via the security protocol establishment block. The seed piece pool update process, biometric secret

key generation process, and data encryption result can be confirmed via the user interface block.

Table 4.1 Block summary of BKG system

Block	Function	Description
Biometric information collection block	IPI Data Collection	Interworking with PPG sensors to collect IPI data and convert data to be used for seed piece generation
Seed piece pool management block	Synchronize seed piece pool information between each node	The main node performs bloom filter output and BCH encoding to generate the seed piece. The sub node performs BCH decoding and IPI sorting based on the information received from the main node, and then updates the seed piece pool and generates metadata. The main node receives and verifies the generated metadata and synchronizes the seed piece pool
Biometric secret key generation block	Seed generation function	Generate seed from seed piece pool based on preset seed piece count value
	Biometric	Generate the operating state value

	secret key renewal function	with the generated seed as input. Then the operating state value is updated using the new seed and additional inputs
	Biometric secret key generation function	Generates a biometric secret key using the preset biometric secret key setting value and the operation state value
User interface block.	GUI	Parameter input UI for simulator environment configuration Simulation state output such as seed pool update status / key agreement/data encryption
Security protocol establishment block	Data encryption/decryption function	Proceed with data encryption/decryption using the generated biometric secret key

4.2 Biometric Secret Key Generator

This section describes the detailed design of the biometric secret key generator block.

4.2.1 Biometric Information Collection Block

The biometric information collection block receives the IPI from the PPG sensor attached to the body. Based on the initial value set from the user interface block, only the part to be used for the seed piece is extracted from the received IPI value. The IPI values are output in binary code, which uses binary gray code to facilitate error correction. As shown in Table 4.2, the binary gray code is a code system that causes the number to be changed by one bit whenever the binary value is incremented by one, thereby reducing the bit change of the IPI and correcting a larger amount of IPI data than the binary data.

Table 4.2 Gray Code

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

4.2.2 Seed Piece Pool Management Block

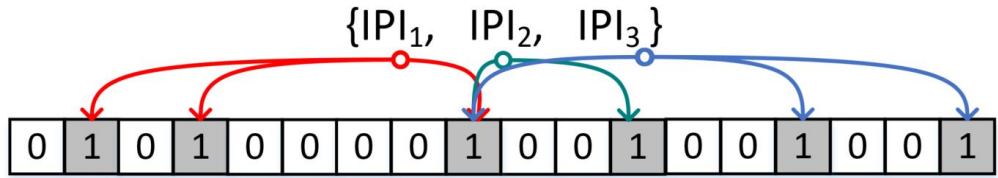
The main functions of the seed piece pool management

block are as follows:

- Encoding / decoding using BCH codes: encoding is performed on the collected IPI set for seed piece generation, and a parity code is generated. The parity code is used for BCH decoding on the sub-nodes, to match the IPI.
- Seed piece synchronization using a Bloom filter: the IPI data collected from the main node generates a BFO value. The sub-node uses the BFO information received from the main node in the decoded IPI set to select only the same IPI as the main node.
- Seed piece pool update: the maintenance of the seed piece pool by updating generated seed pieces to the counting Bloom filter, the deletion of the seed piece used in the key generation from the seed piece pool, and the synchronization of the seed piece pool between the nodes
- Seed piece output: based on the initial number of seed pieces for biometric secret key generation, a certain number of seed pieces in the seed pieces pool are transferred to the biometric secret key generation block.
- Seed piece pool metadata verification: the verification of the output value of the counting Bloom filter, which is the metadata of the seed piece pool

The BCH code, which is the error correction code used in

this system, is required to define n, k, t three setting values in advance; where n is the total data size generated through BCH encoding. In this system, 63, 127, and 255 are used; where k is the size of the original bit data, and t is the number of correctable errors. For example, if the BCH encoding is performed to correct the 87-bit IPI aggregated data by 26 bits, 255-bit data is generated, and the parity code value becomes 168 bits as the $n-k$ value. The main node transfers only the parity code value of 168 bits to the sub-node, and the sub-node corrects the error of the 87-bit data through the 87 bits of the collected IPI data and the 168-bit parity code received from the main node. If more than t errors occur, all 87 bits are not recovered. Even if it is decoded, some data may be restored differently to the original data. In this case, a Bloom filter can be used to accurately synchronize the IPI set. A Bloom filter is a hash-based filter that is used to check whether an element in a data set belongs to that set, as shown in Figure 4.4. The main node and the sub-node set the same Bloom filter parameter in advance. The main node transmits the parity code and the BFO value for the IPI set to the sub-nodes. The sub-node checks the individual IPIs in the decoded IPI set for the same IPI through the BFO.



Bloom Filter Ouput

Figure 4.4 Identify IPI using Bloom Filter

The metadata of the seed piece pool consists of the counting Bloom filter as shown in Figure 4.5. The counting Bloom filter is an extension of the Bloom filter, which adds element expansion and deletion functions. When the seed piece is synchronized between the two nodes, the sub node delivers the counting BFO value to the main node, which inputs the Bloom filter value and seed piece information. The main node checks the seed piece value using the BFO value received from the sub-node, and synchronizes the seed piece pool with the output value of the counting Bloom filter of the sub-node. When the synchronization is completed, the synchronization completion message is transmitted to the sub-node to complete the update. The size of the seed piece pool is defined in advance between the nodes. If more seed piece data is received by the seed piece pool than the size of the seed piece pool, the oldest seed piece data is deleted. In other words, the seed piece pool is maintained by a first in, first out (FIFO) structure.

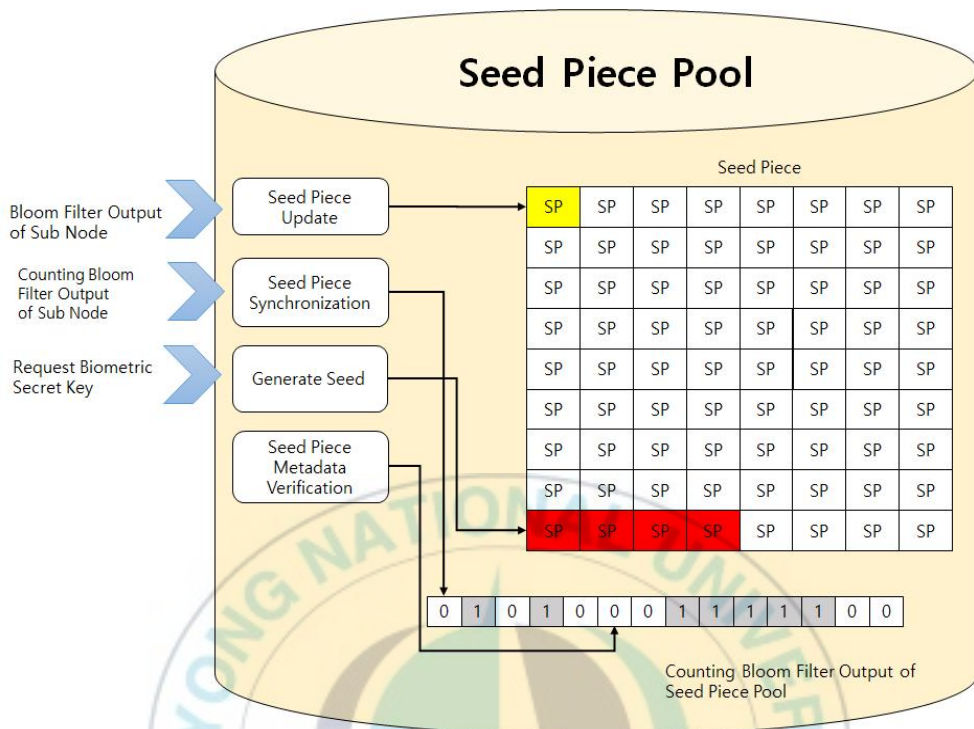


Figure 4.5 Function of seed piece pool

When a seed generation request for biometric secret key generation is received by the seed piece pool, n pieces of the seed pieces defined in advance are output. If n seed pieces are not collected in the seed piece pool, the system waits until more than n seed piece are filled. When more than n seed pieces are collected, the seeds are output from the seed piece pool. When the seed piece is output from the seed piece pool, the used seed piece information from the seed piece pool is deleted, and the element information is also deleted from the output value of the counting Bloom filter. The update information of the seed piece pool is then transmitted from

the sub-node to the main node, and the seed piece pool of the main node is inspected using the counting BFO value of the sub-node. The seed pieces of the main node that have not passed the test are deleted in the seed piece pool, and the counting BFO is generated and transmitted to the sub-nodes.

4.2.3 Biometric Secret Key Generation Block

The function of the seed piece pool management block is as follows:

- Seed generation: the generation of seed pieces using $n > 1$ seed pieces to increase the complexity of the seeds, and an increase in cryptographic safety using multiple seed pieces
- Biometric secret key renewal: the creation of an operation status value based on the seed generated for biometric secret key generation, and continuously updated operating values with additional inputs and new seeds
- Biometric secret key generation: the biometric secret key output based on the operating status value

The seed derivation functions in Figure 4.6 are used to increase the security strength of the seed. Each seed piece is used as an input to the hash function with counter and session

i values, in addition to the seed pieces. Each seed piece is transformed into a hashed seed piece through a hash function, and the n hashed seed pieces are then combined to generate an i session seed.

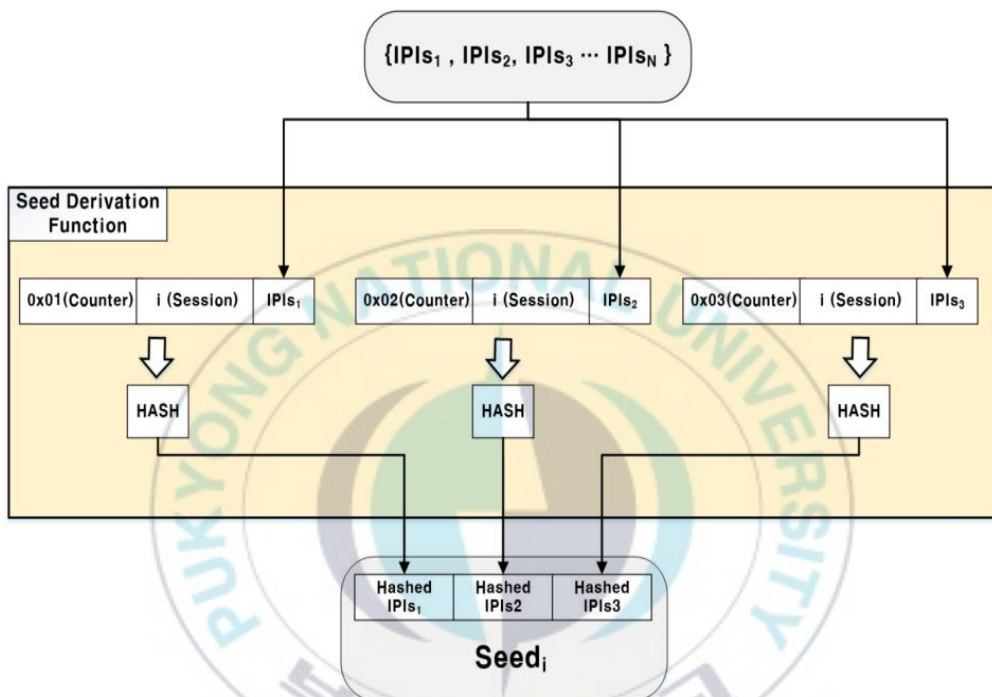


Figure 4.6 Seed derivation function

The seed length is determined by the seed function used in the seed derivation function. Table 4.3 shows the hash function and seed length used in the seed derivation function.

Table 4.3 Seed length according to hash function

Hash	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Seed length	440	440	440	888	888

The length of the seed piece is pre-defined as the initial set value. The lengths of the seed pieces are all set to be equal according to the pre-defined values. For example, if the length of the seed piece is 20 and the hash function of SHA-1, SHA-224, or SHA-256 is used, 22 seed pieces are seeded using the session value and the counter value (1-22). The seed is created by connecting the hashed seed pieces in a line. The generated seed is then used to generate the session key using the hash function. The session key of session i is the same as the biometric secret key length. This session key is used to verify the seed. The seed of each session is used as an input to a previously defined hash function for biometric secret key generation. The generated hash output is truncated from the left by a pre-defined biometric secret key length.

4.3 Security Protocol Simulator for Biometric Secret Key Verification

In this section, the design of the security protocol simulator

for biometric secret key verification is presented.

4.3.1 Security Protocol Establishment Block

The function of the security protocol establishment block is as follows.

- Data encryption: the encryption of data to be sent to other nodes, and the generation of message integrity data
- Data decryption: the decryption of encrypted data received from other nodes, and the verification of the message integrity

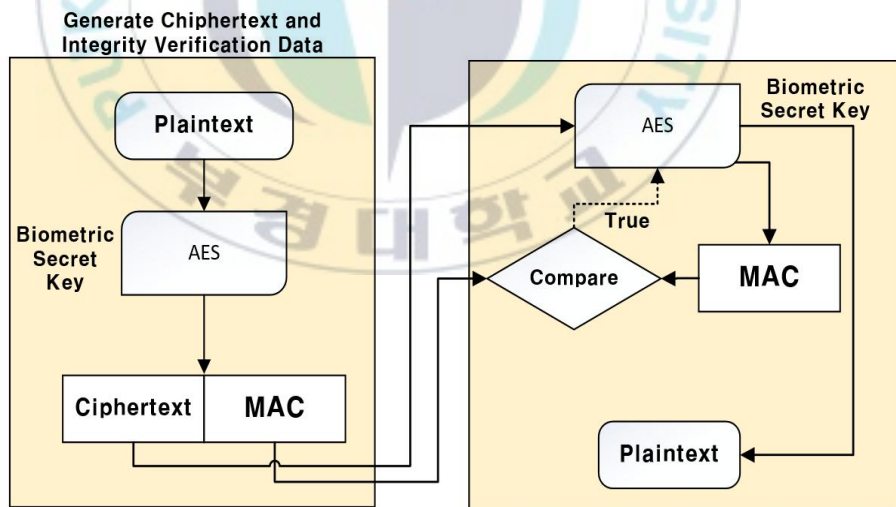


Figure 4.7 Data encryption and integrity verification data generation and verification process

As shown in Figure 4.7, the data to be transmitted is encrypted using the biometric secret key, and the data for integrity verification is generated. Moreover, the integrity of the encrypted data received from another node is checked. The encryption algorithm used for data encryption in this system is the Advanced Encryption Standard (AES).

4.3.2 User Interface Block

The user interface block is used for inputting the parameters required for biometric secret key generation. The main parameters are the BCH control parameter, seed piece length, seed piece pool size, IPI data size, and Bloom filter size, as shown in Figure 4.8.

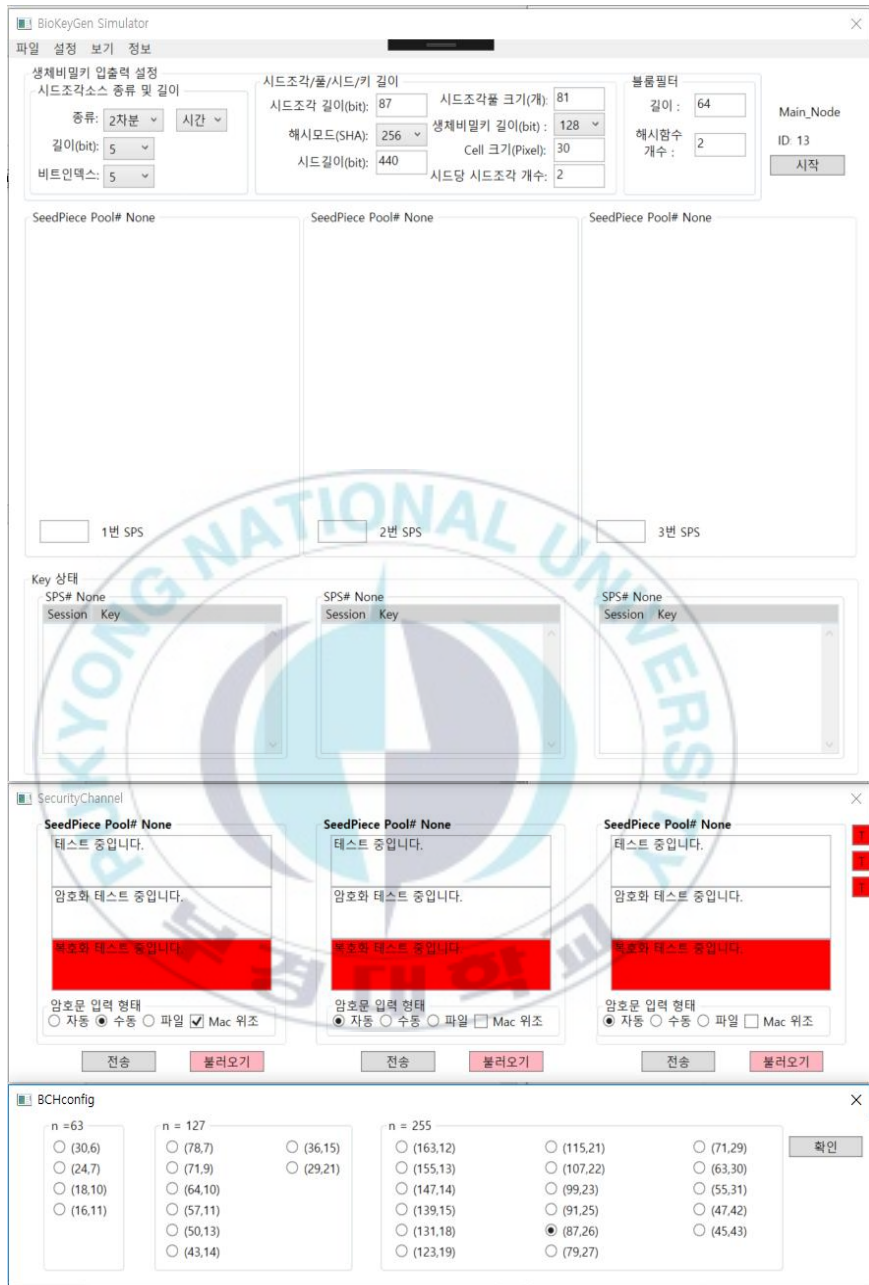


Figure 4.8 User interface

In addition to the parameter setting function, the seed

piece pool status, biometric secret key generation status, and data encryption process can be checked as shown in Figure 4.9.

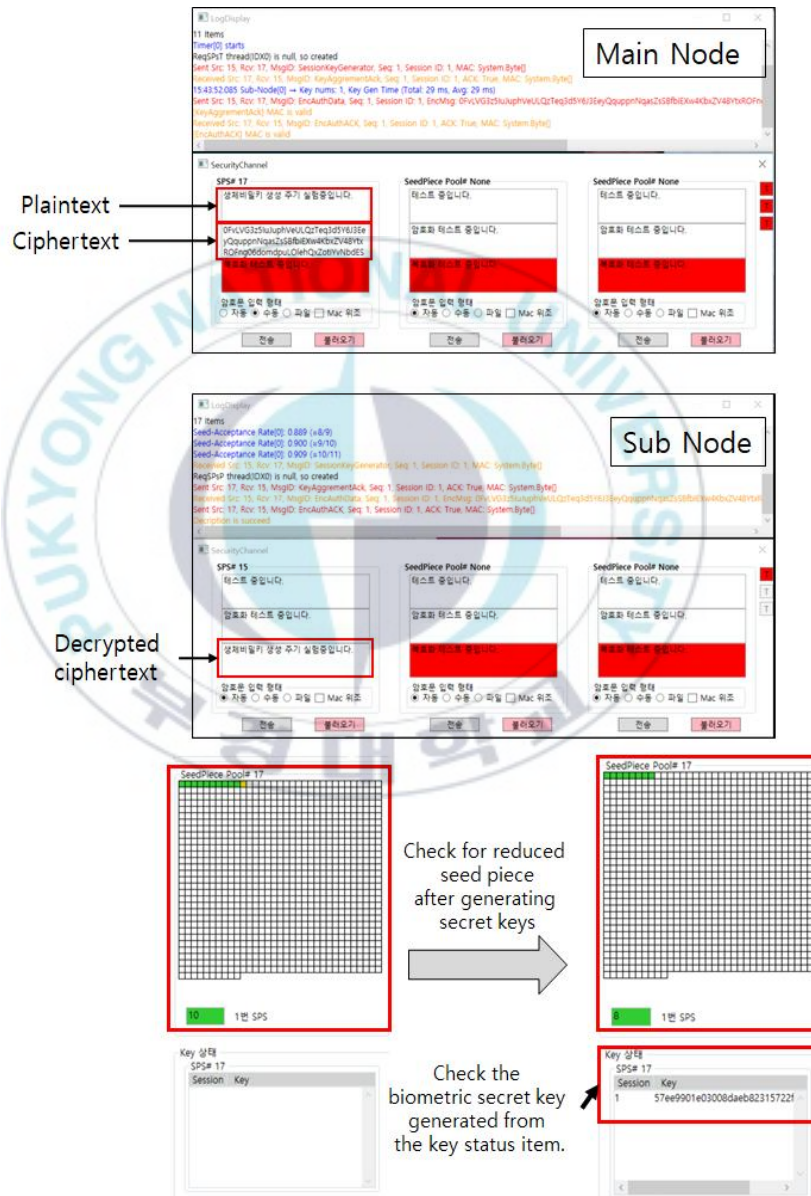


Figure 4.9 BKG and encryption process monitoring

Chapter 5. Experiment

This chapter describes the experiments and results of the verification of the most efficient parameters for biometric secret key generation by implementing the BKG system using IPI.

5.1 IPI Entropy Test by LED Wavelength of PPG Sensor

Prior to the tests conducted on the BKG system, experiments were conducted on the entropy characteristics of the PPG sensor using the LED wavelength. The IPI measurement results were different for each LED wavelength. Therefore, the following experiment was conducted to determine the wavelength with the highest entropy.

5.1.1. IPI Data Collection

The IPI data were collected using wavelengths of green, red, and infrared light that could be used to measure PPG signals. The data to be collected was the IPI value and the first and second derivative values of the LED for each wavelength. Moreover, the collected data was converted into a

16-bit binary number. The specifications of the multi-wavelength LED sensor used in the experiment are presented in Table 5.1 below.

Table 5.1 Multi-wavelength LED sensor specifications

Device	Specifications
Multi wavelength LED sensor	<ul style="list-style-type: none"> – Green, Red, Infrared LED – 6bit programmable LED current to 50mA – Dynamic Range 100dB – Programmable Transimpedance Gain $10k\Omega \sim 2M\Omega$ – 0~1000 amperes-Per-Second – Internal clock 4MHz – External clock 4~ 60 MHz

Approximately 150,000 bits were collected to identify the distribution of 0s and 1s per bit, as shown in Figure 5.1, with an uneven distribution of 0,1 from 16 to 13. Therefore, the high bits without randomness were removed, and entropy tests were conducted using 12 bits and 11 datasets.

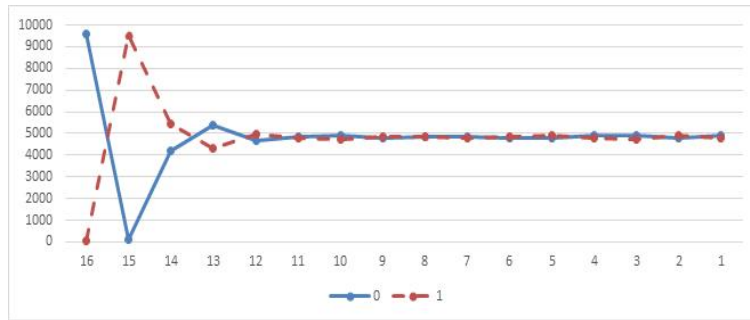


Figure 5.1 0,1 distribution by bit

5.1.2. AIS.31 Test

The AIS.31 standard was used for the entropy test. Moreover, AIS.31 is a standard established by the German Federal Office for Information Security, which is divided into the P1 class and P2 class, as shown in Table 5.2 below [21]. The entropy test was conducted from T1–T5 during the P1 test.

Table 5.2 AIS.31 Test

Class	Test	Means
P1	T0 (disjointness test)	Subsequent members are pairwise different
	T1 (monobit test)	Uniform test for bit sequence of length 20,000
	T2 (poker test)	Goodness of fit test for the number of 4 bits block
	T3 (runs test)	Test for the number of run which has l -length.
	T4 (long run test)	Check up the occurrence run of length ≥ 34 .
	T5 (autocorrelation test)	Autocorrelation value of $\sum (i\text{-th bit} \oplus i+5000\text{-th bit})$ which is approximately 2500
P2	T6 (uniform distribution test)	Uniform distribution test using ratio of 0's and 1's.
	T7 (comparative test)	Goodness of fit test for h blocks by comparison
	T8 (entropy test)	Estimate entropy as minimum distance of blocks.

5.1.3. Result

Table 5.3 summarizes the test results. The green LEDs did not pass the T5 test for the first and second derivative data, and the red LED did not pass the T5 test for the 12 bits of the second derivative. On the other hand, the infrared LED passed all the tests. The significance level of T5 was $2372 < T < 2674$ for the 20,000-bit test. The experimental results revealed that the entropy is the highest in the infrared LED.

Therefore, subsequent experiments were conducted with data collected using infrared LEDs.

Table 5.3 Entropy test result by LED wavelength
(P:Pass F:Fail)

TEST DATA(LED)	T1	T2	T3	T4	T5
GREEN 12bit	P	P	P	P	P
GREEN 11bit	P	P	P	P	P
RED 12bit	P	P	P	P	P
RED 11bit	P	P	P	P	P
INFRARED 11bit	P	P	P	P	P
INFRARED 12bit	P	P	P	P	P
GREEN 1deriv 12bit	P	P	P	P	F
GREEN 1deriv 11bit	P	P	P	P	P
RED 1deriv 12bit	P	P	P	P	P
RED 1deriv 11bit	P	P	P	P	P
INFRARED 1deriv 12bit	P	P	P	P	P
INFRARED 1deriv 11bit	P	P	P	P	P
GREEN 2deriv 12bit	P	P	P	P	F
GREEN 2deriv 11bit	P	P	P	P	P
RED 2deriv 12bit	P	P	P	P	F
RED 2deriv 11bit	P	P	P	P	P
INFRARED 2deriv 12bit	P	P	P	P	P
INFRARED 2deriv 11bit	P	P	P	P	P

5.2 Entropy Verification of Seed Piece Data

The seed piece collected from the BKG system must have sufficient entropy for cryptographic safety. In addition, if the size of one IPI is large, the time required to generate seed piece data is shortened, and the entropy decreases. On the other hand, if the size of one IPI is reduced, the use of multiple IPIs increases the entropy of the seed piece, and a significantly longer time is required to generate the seed data. In this section, the most efficient IPI bit size and bit index that can be used as a seed piece are obtained.

5.2.1. Experiment

Given that the entropy of the infrared LED was the highest in the entropy experiment, with respect to the wavelength of emitted light, the experiment was conducted with the PPG sensor using an infrared LED. The PPG sensor, BKG, and AIS.31 based entropy verification program were configured as shown in Figure 5.2. The AIS.31 test proceeded with the P1 class (T0 to T5). To confirm the entropy difference between the first and second derivative values, the first and second derivative values of approximately 1.8 Mb were collected from the PPG sensor, and tests T1–T5 were conducted. Based on the result, the AIS.31 test was conducted by collecting a

second derivative value of approximately 166 Mb.

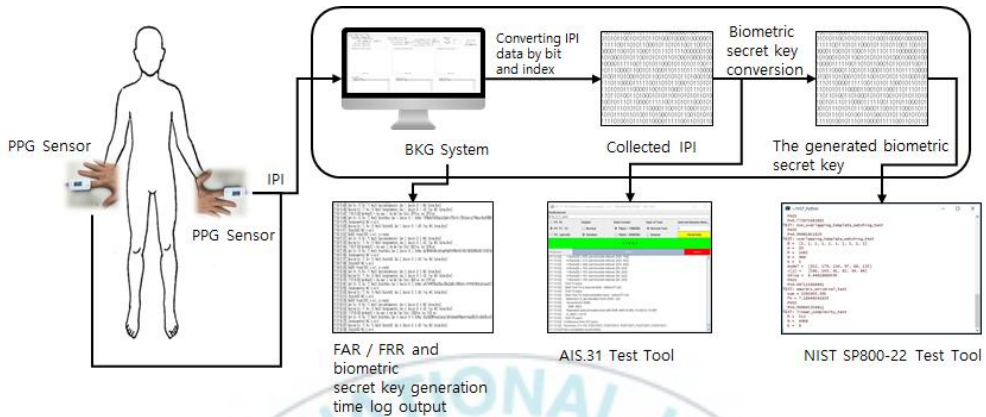


Figure 5.2 Biometric secret key generation simulator test configuration

5.2.2. Result

As shown in Table 5.4, the tests were only passed on the 4-, 5-, and 6-bit blocks. Moreover, the tests were passed only when the index of the passed block was 5 or higher. The failure and pass sections of the entropy test were then analyzed using the 4-bit derivative value

Table 5.4 IPI of first and second derivative value AIS.31 test result

Bit	Index	Derivative	T1	T2	T3	T4	T5
4	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F

		2	F	F	F	F	F
		1	F	F	F	F	F
	3	2	F	F	F	F	F
		1	F	F	F	F	F
	4	2	F	F	F	F	F
		1	F	F	F	F	F
	5	1	P	P	P	P	P
		2	P	P	P	P	P
	6	1	P	P	P	P	P
		2	P	P	P	P	P
	7	1	P	P	P	P	P
		2	P	P	P	P	P
5	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F
		2	F	F	F	F	F
	3	1	F	F	F	F	F
		2	F	F	F	F	F
	4	1	F	F	F	F	F
		2	F	F	F	F	F
	5	1	P	P	P	P	P
		2	P	P	P	P	P
	6	1	P	P	P	P	P
		2	P	P	P	P	P
6	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F
		2	F	F	F	F	F
	3	1	F	F	F	F	F
		2	F	F	F	F	F
	4	1	P	F	F	F	F
		2	F	F	F	P	F
	5	1	P	P	P	P	P
		2	P	P	P	P	P
7	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F
		2	F	F	F	F	F

	3	1	F	F	F	F	F
		2	F	F	F	F	F
	4	1	P	F	F	F	F
		2	P	F	F	P	F
8	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F
		2	F	F	F	F	F
	3	1	F	F	F	F	F
		2	F	F	F	F	F
9	1	1	F	F	F	F	F
		2	F	F	F	F	F
	2	1	F	F	F	F	F
		2	F	F	F	F	F
10	1	1	F	F	F	F	F
		2	F	F	F	F	F

In the T1 mono-bit test, the number of one should be close to 10,000. However, the intervals of indexes 1–4 did exceeded the test range and had a maximum of approximately 18,000 and minimum of approximately 7,500. On the other hand, indexes 5–7 were distributed nearly around 10,000 as shown in Figure 5.3

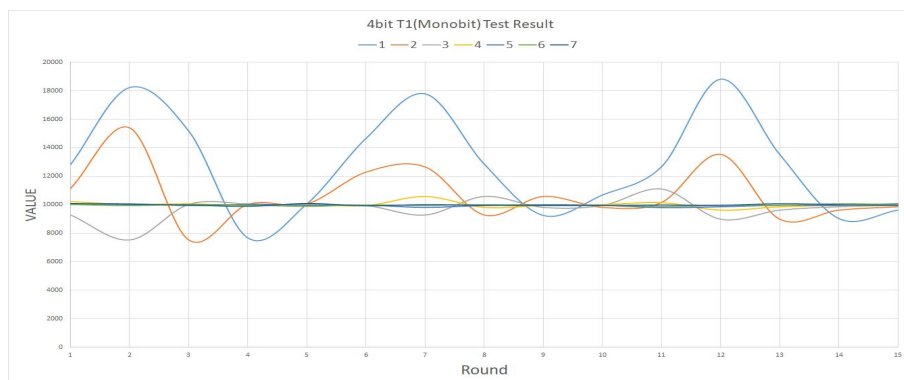


Figure 5.3 Monobit test results

It was expected that the T2 Poker test results would be close to 0. However, the results of indexes 1–4, which did not pass the test, had an average value of 7326. Moreover, as shown in Figure 5.4, the deviation was significant. Indexes 5–7, which passed the test, exhibited a distribution of 4.5–28.5.

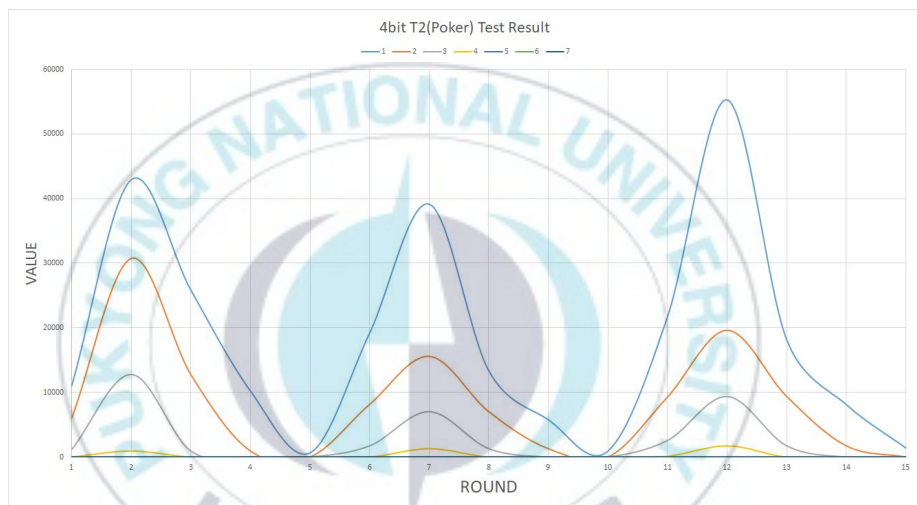


Figure 5.4 Poker test results

The T3 Run test is a test conducted to verify the number of consecutive zeros and ones that should be included within a certain reference value according to the length of the run. The intervals of indexes 1–4 that failed the test were not uniformly distributed beyond the reference value as shown in Figure 5.5. Indexes 5–7, which passed the test, were uniformly distributed within the pass reference range as shown in Figure 5.6.

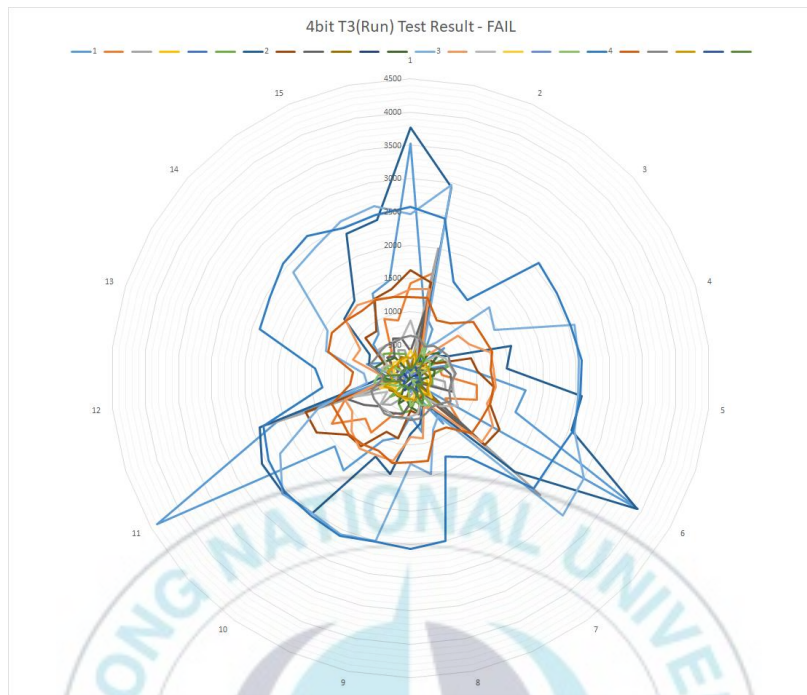


Figure 5.5 Run test fail result

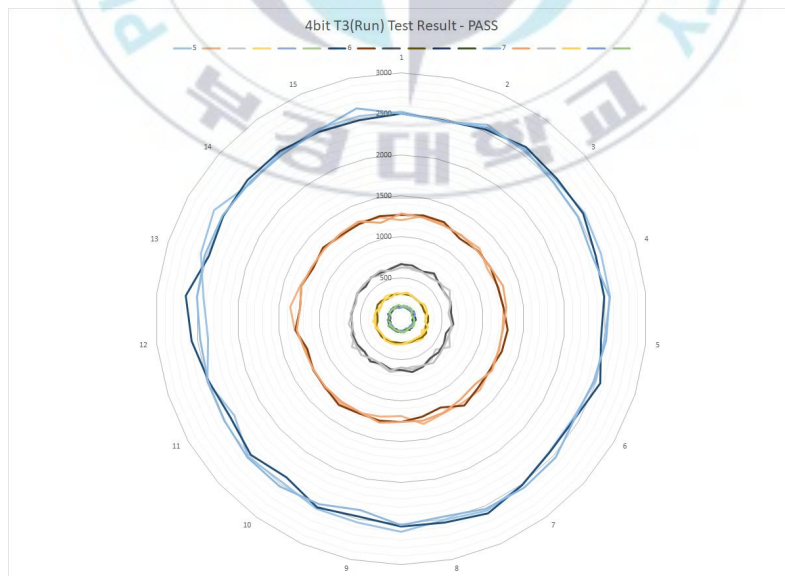


Figure 5.6 Run test pass result

The T4 Long Run test is a test conducted to detect the occurrence of a run with a length greater than 34. Moreover, Table 5.5 shows that a larger the index value results in a higher test-passing rate.

Table 5.5 Long Run test result

Index Round	1	2	3	4	5	6	7
1	P	P	P	P	P	P	P
2	F	F	F	F	P	P	P
3	F	F	F	P	P	P	P
4	F	F	P	P	P	P	P
5	P	P	P	P	P	P	P
6	F	P	P	P	P	P	P
7	F	F	F	P	P	P	P
8	F	F	P	P	P	P	P
9	F	P	P	P	P	P	P
10	P	P	P	P	P	P	P
11	P	P	P	P	P	P	P
12	F	F	F	P	P	P	P
13	F	F	P	P	P	P	P
14	F	P	P	P	P	P	P
15	F	P	P	P	P	P	P

The T5 autocorrelation test is an autocorrelation verification test with a result value greater than 2326 and less

than 2674. Indexes 1–4 index were smaller than the test passing minimum of 2326, and therefore failed the test. For indexes 5–7, which passed the test, it was confirmed that all of the 15 rounds had a stable test passing value as shown in Figure 5.7

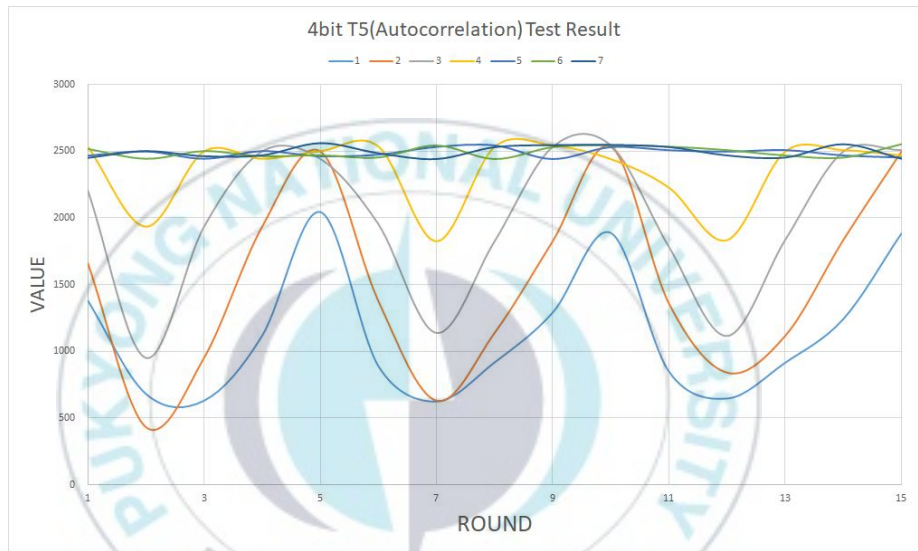


Figure 5.7 Autocorrelation test result

The entropy test revealed that the second derivative IPI value was the highest. For more accurate experiments, more second derivative IPIs were collected, and tests T0–T5 were conducted. In this experiment, a commercial PPG sensor with higher stability than the multi-wavelength LED PPG sensor used in the previous experiment was used. The specifications of the Ubpulse 340 sensor used in the experiment are shown in the Table 5.6[22].

Table 5.6 Ubpulse 340 sensor specifications

Device	Specifications
Ubpulse 340	<ul style="list-style-type: none"> – Optical, using light absorption modulation via capillary filling pulsations. – 940nm Infrared LED – Light noise is minimized using ELP (Environment Light Protection) technology. – High Precision Peak Detection from 2nd Derivatives of PPG. – Clock Resolution: 0.000976 sec. – Clock Accuracy : 0.002% – The clock is divided by 32 from main clock. – Main clock : 32.768kHz Quartz Crystal Oscillator with accuracy : +- 20ppm (0.002%)

The results are presented in Table 5.7 below. As a result, it was confirmed that the 4-bit 6-index, 5-bit 5,6-index, 6-bit 5-index, and 7-bit 3,4-index passed the test, as shown in Table 5.7. The test was mainly passed by the upper index section, which contained many high-entropy bits. On the other hand, 8, 9, and 10-bits could not pass the test because the low and high entropies were used simultaneously.

Table 5.7 Seed piece entropy test results

Bit	Index	T0	T1	T2	T3	T4	T5
4 (205 round)	1	F	F	F	F	P	F
	2	F	F	F	F	F	F
	3	F	F	F	F	F	F
	4	F	F	F	F	P	F
	5	F	P	F	P	P	P
	6	P	P	F	P	P	P
	7	P	P	P	P	P	P
5	1	F	F	F	F	P	F
	2	F	F	F	F	F	F
	3	F	F	F	F	P	F
	4	F	F	F	F	P	F
	5	P	P	P	P	P	P
	6	P	P	P	P	P	P
6	1	F	F	F	F	P	F
	2	F	F	F	F	F	F
	3	F	F	F	F	P	F
	4	F	F	F	F	P	F
	5	P	P	P	P	P	P
7	1	F	F	F	F	P	F
	2	F	F	F	F	P	F
	3	P	F	F	F	P	F
	4	P	F	F	F	P	F
8	1	F	F	F	F	P	F
	2	F	F	F	F	P	F
	3	F	F	F	F	P	F
9	1	F	F	F	F	P	F
	2	F	F	F	F	P	F
10	1	F	F	F	F	P	F

5.3 FAR/FRR Test

Five pairs of experimenters conducted the false acceptance rate (FAR) test. The BCH function parameters used for the seed synchronization were $n = 255$, $k = 87$, $t = 26$. In addition, 5-bit seed pieces were used. The FRR test was performed using the same experimental setup as that of the FAR test. Unlike previous researches, all the steps from PPG signal measurement to seed piece generation were performed in real time.

5.3.1. FAR Test

As shown in Table 5.8, the FAR test revealed that 0,0% of the 1,037 secret key generators failed 0 times.

Table 5.8 FAR Test result

Experimenter	Number of attempts	Number of matches	FAR
5 pairs	1,037	0	0.00%

5.3.2. FRR Test

The result of the FRR test was 12.57%, which is 5,696 times of the total 45,317 secret key generation attempts, as shown in Table 5.9. Therefore, the seed agreement rate was

87.43%, which is larger than 85%.

Table 5.9 FRR Test result

Experimenter	Number of attempts	Number of matches	Number of mismatches	Seed agreement rate	FRR
5 pairs	45,317	39,621	5,696	87.43%	12.57%

5.4 Biometric Secret Key Randomness Test

The NIST SP800-22 technique was used for the randomness test of the biometric secret key generated from the collected seed information. The NIST SP800-22 is a general evaluation method for pseudo-random tests. In this study, 8 out of 15 items were tested. A summary of each test is presented below [23].

- (1) Frequency Monobit: tests whether 0 and 1 are uniformly distributed
- (2) Run : tests whether the number of 1s in a certain length M is $M/2$
- (3) Test for the Longest Run of Ones in a Block: tests whether the number of runs of the maximum length of consecutive 1s in each block appears uniformly in the block

- (4) Binary Matrix Rank: tests the linear dependencies between fixed-length substrings of sequences
- (5) Discrete Fourier Transform: tests the deviation of the occurrence frequency of the predicted pattern
- (6) Maurer's "Universal Statistical": tests for compression without loss of information
- (7) Linear Complexity: the information that the sequence considers to be random
- (8) Approximate Entropy: tests the frequency of duplicate patterns

5.4.1. Entropy Test Results By Number of Seed Pieces

As the input of four hash functions (SHA1, SHA256, SHA384, SHA51), 1–6 seed pieces were used to generate 128-bit and 256-bit biometric secret keys, and the size of the generated biometric secret keys was 1,028,016 bits. The NIST 800–22 test requires at least 1,000,000 bits, and 55,000,000 bits in 55 iterations. The usability of the biometric secret key was examined by conducting a test using a minimum number of bits. As a result, it was confirmed that even if the number of seed pieces is large, as shown in Table 5.10, the test cannot be passed.

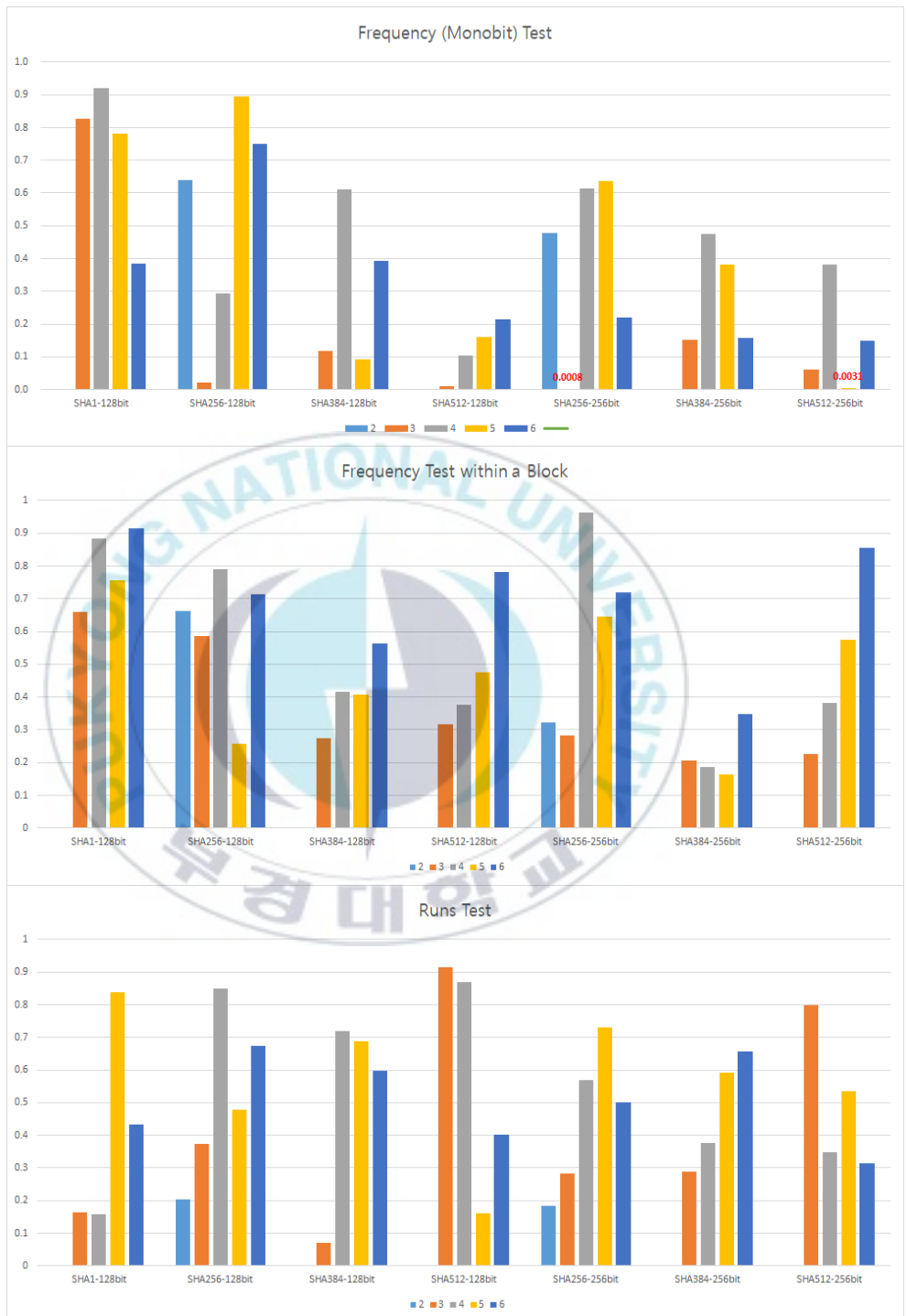
Table 5.10 Entropy test results by number of seed pieces

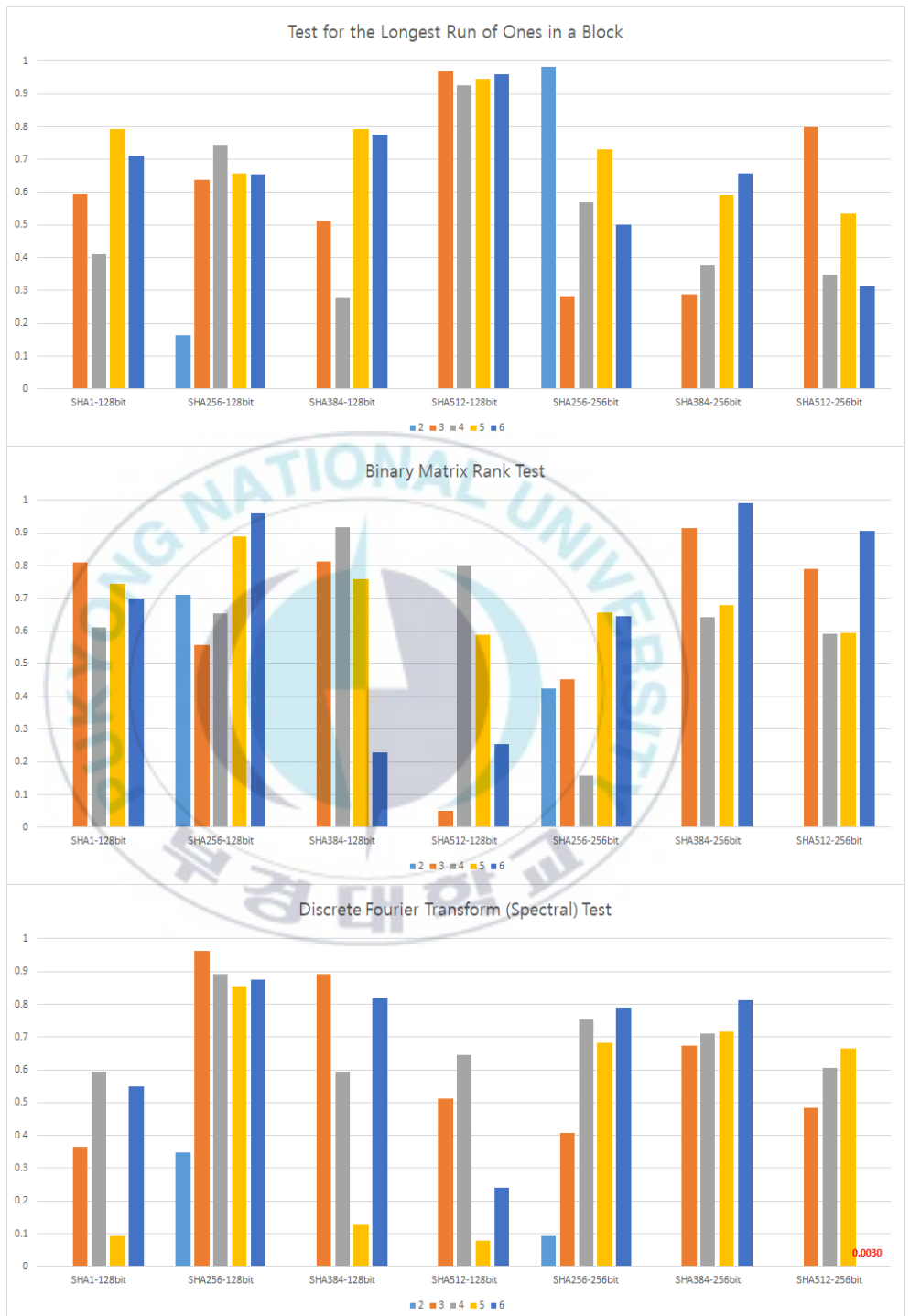
Hash function	Key length	Number of seed piece	Result
SHA1	128 bit	3	PASS
		4	PASS
		5	PASS
		6	PASS
SHA256	128 bit	2	PASS
		3	PASS
		4	PASS
		5	PASS
		6	PASS
	256 bit	2	PASS
		3	FAIL
		4	PASS
SHA384	128 bit	5	PASS
		6	PASS
	256 bit	3	PASS
		4	PASS
		5	PASS
		6	PASS
SHA512	128 bit	3	PASS
		4	PASS
		5	FAIL
		6	PASS
	256 bit	3	PASS
		4	PASS
		5	FAIL
		6	FAIL

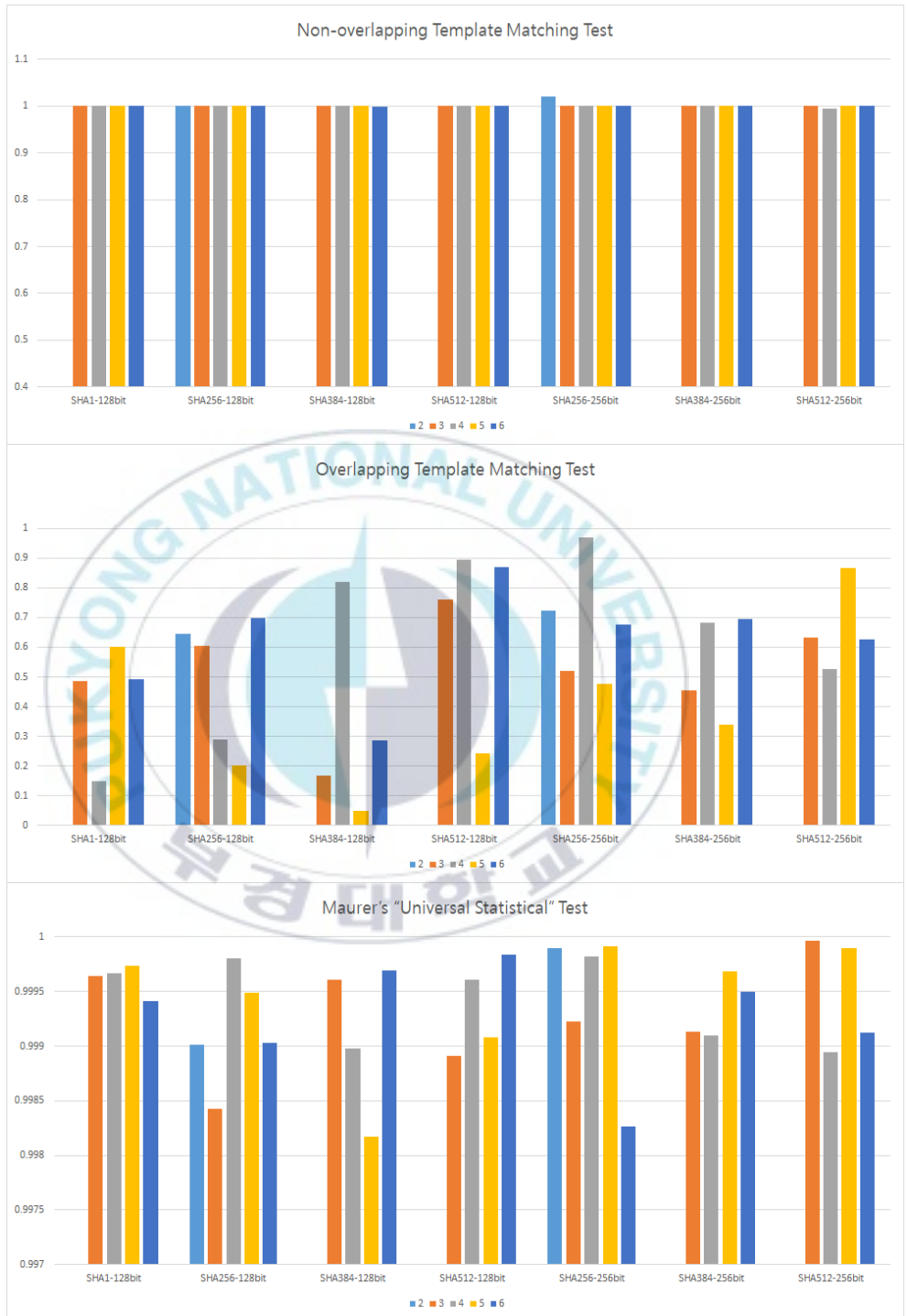
5.4.2. Correlation Between the Number of Seed Pieces and the Randomness

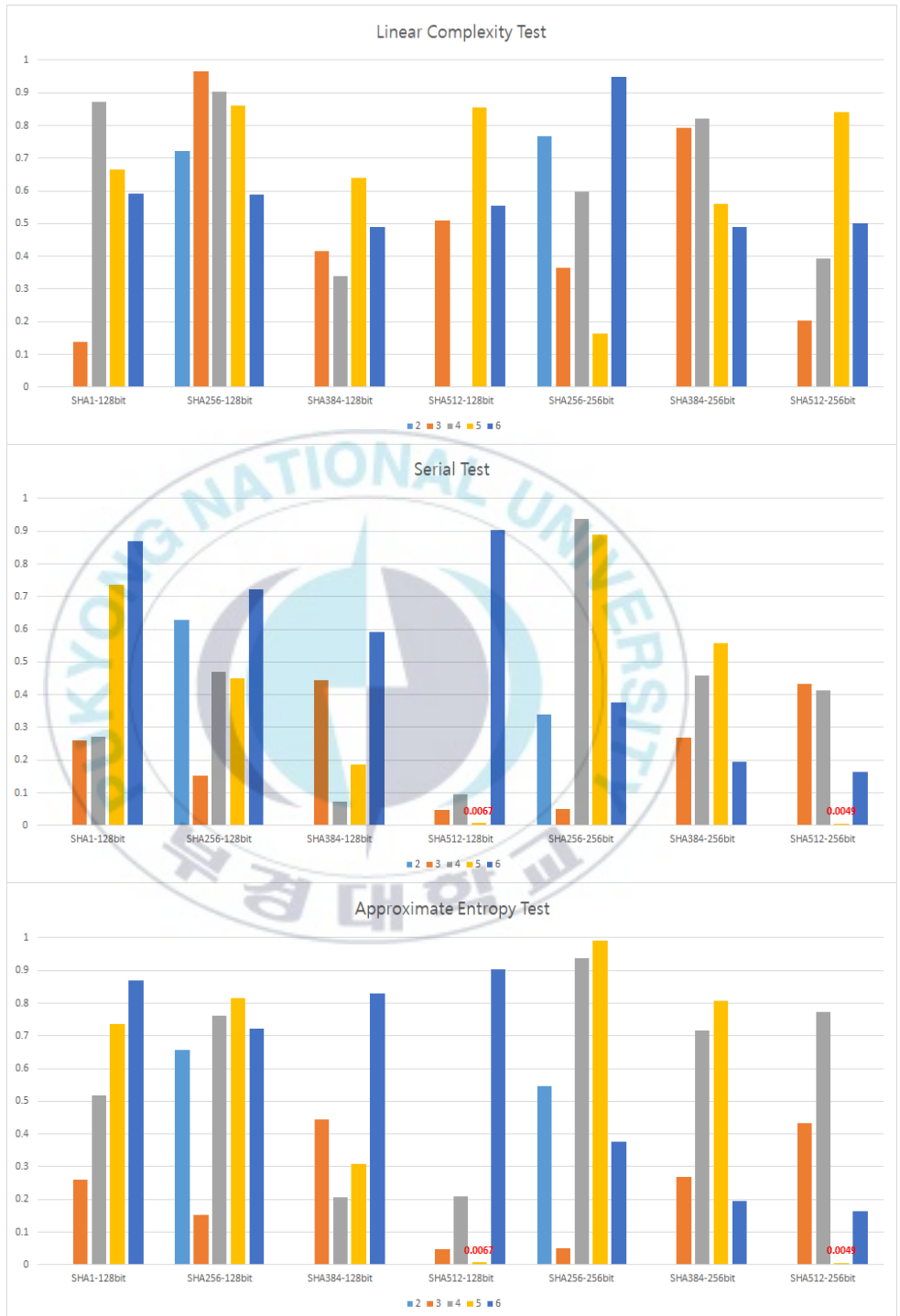
To confirm the correlation between the number of seed pieces and the randomness, the experimental results are presented in Figure 5.8. The NIST SP800-22 defines randomness as when the P-value is greater than 0.01. As can be seen from the graph below, the correlation between the number of seed pieces and the P-value was very small.











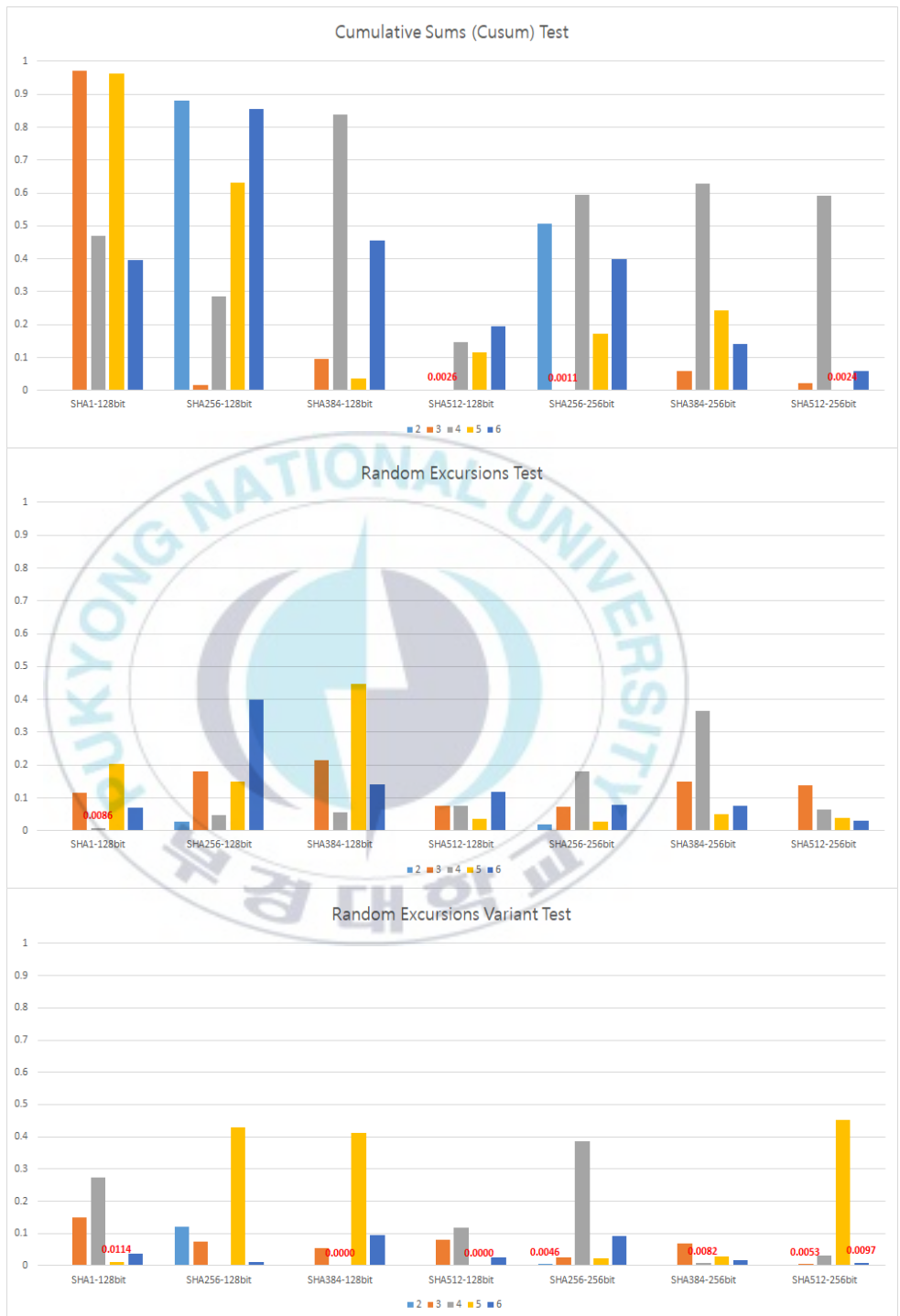


Figure 5.8 Biometric secret key randomness test results

5.5 Biometric Secret Key Generation Time Test

The biometric secret key generator is used to determine the time required for the biometric secret key generation.

5.5.1. Experimental Environment

The size of the seed piece pool used for the IPI generation time test was set as 200. The number of seed pieces required to generate the key was set as 2, and the IPI was measured. First, when the seed pieces of the seed piece pool were filled with two pieces, the key generation was performed, and the time required was confirmed using the biometric secret key generator log. Thereafter, when 200 of the seed pieces were filled in the seed piece pool, the batch key generation was performed, and the time spent on the log was checked.

5.5.2. Result

The average biometric secret key generation time was approximately 33 s in the individual test, as shown in Table 5.11, and 0.028 s in the batch test.

Table 5.11 IPI generation time test result

	Average (ms)	Total (ms)
Individual creation	33,039	3,303,861
Batch creation	28	1,790

5.6 IPI Recovery Test

In this experiment, 1,000 key exchange attempts were performed based on the IPI data collected over 4 h, to test the measurement recovery algorithm. In the experiment, a 5-bit IPI index value and BCH (255, 87, 26) were used in the same manner as in the FAR / FRR test.

5.6.1. Entropy Test

To verify the entropy of the recovered IPI, the Mono-Bit, Poker, Run, Long Run, and Autocorrelation tests among the nine tests of the AIS.31 were conducted. A total of 120,000 bits of data were processed in five rounds.

Table 5.12 Result of entropy test of corrected seed piece

Test	Limits	Value	Result
Monobit	9645 < value < 10346	1:9800 2:9978 3:9743 4:9963 5:9978	Pass
Poker	1.03 < value < 57.4	1: 15.7568 2: 20.4059 3: 27.5520 4: 19.6415 5: 13.8368	Pass
Run	All Passed	1: Pass 2: Pass 3: Pass 4: Pass 5: Pass	Pass
Longrun	Long Run = 34	1: Pass 2: Pass 3: Pass 4: Pass 5: Pass	Pass
Autocorrelation	2326 < value < 2674	1:2504 2:2431 3:2511 4:2482 5:2498	Pass

As a result of the test, all the five tests were passed, as shown in Table 5.12. Therefore, the corrected IPI does not significantly affect the reduction of entropy.

5.6.2. Number of Recovery

Table 5.13 shows the results of the recovery test. The number of IPIs restored by the maximum threshold value was 15, and the number of IPIs restored by the minimum threshold value was four.

Table 5.13 Number of IPI corrections

Recovery of maximum threshold	Recovery of minimum threshold
15	4

The total number of recovered IPIs was 19 and the key generation rate was 99.4%. However, when the IPI was not recovered, the key generation rate was 91%. Therefore, it was confirmed that the proposed IPI recovery algorithm helps reduce key generation time.

Chapter 6. Conclusion

In this thesis, several studies on secret key generation methods using biosignals, and the main research on secret key generation methods using the IPI collected from the PPG sensor, were discussed. Among them, fuzzy vault, which is a secret key generation method using representative biosignals, was confirmed as vulnerable to correlation attack; and fuzzy commitment revealed that the efficiency of secret key generation could be reduced due to misdetection.

Based on the analyzed vulnerabilities, the seed piece error correction, misdetection recovery algorithm, IPI rearrangement, IPI selection, and seed piece pool method were proposed to overcome the limitations of fuzzy commitment

Based on the proposed methods, a BKG system was designed and implemented to verify the optimal parameters for secret key generation. First, it was confirmed that the entropy of the infrared LED was the highest by conducting an experiment on the entropy difference of the IPI data per LED wavelength.

Second, the AIS.31 entropy test was performed on the seed pieces collected using the biometric secret key generation simulator, and the 5-bit seed piece was confirmed as having the highest entropy. Based on this, the FAR/FRR test results were 0% for FAR and 12.57% for FRR.

Third, an entropy test was performed according to the number of seed pieces, to confirm the secret key randomness generated based on the seed piece. Moreover, the correlation between the number of seed pieces and entropy was very small.

Finally, it was verified by the IPI generation time test that approximately 28ms is required for the generation of the batch key, and it was confirmed that the IPI recovered by this test does not significantly affect the entropy.

To improve safety and performance, it is necessary to continue to increase the number of experimenters and IPI data. Furthermore, further research on continuous protocols and algorithms to reduce computation load is required.

Reference

- [1] S. Bao, C. C. Y. Poon, Y. Zhang and L. Shen, "Using the Timing Information of Heartbeats as an Entity Identifier to Secure Body Sensor Network," in IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 6, Nov. 2008, pp. 772–779.
- [2] Sang-Ho Choi, Ki-Young Shin, Jeauk Kim, Seung-Oh Jin, Tea-Bum Lee, "Classification Model of Chronic Gastritis According to The Feature Extraction Method of Radial Artery Pulse Signal," in Journal of the Institute of Electronics and Information Engineers, 51.1, 2014.1, 185–194.
- [3] G. H. Zhang, C. C. Y. Poon and Y. T. Zhang, "A biometrics based security solution for encryption and authentication in tele-healthcare systems," in 2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, Bratislava, 2009, pp. 1–4.
- [4] M. Rushanan, A. D. Rubin, D. F. Kune and C. M. Swanson, "SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks," 2014 IEEE Symposium on Security and Privacy, San Jose, CA, 2014, pp. 524–539.
- [5] J. Mohana and V. T. Bai, "128 bit key generations from the dynamic behavior of ECG for securing wireless body area network," in ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 18, 2015, pp. 8048–8051,

- [6] P. Bagade, A. Banerjee, J. Milazzo and S. K. S. Gupta, "Protect your BSN: No Handshakes, just Namaste!," 2013 IEEE International Conference on Body Sensor Networks, Cambridge, MA, USA, 2013, pp. 1–6.
- [7] H. Zhao, R. Xu, M. Shu and J. Hu, "Physiological–Signal–Based Key Negotiation Protocols for Body Sensor Networks: A Survey," 2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taichung, 2015, pp. 63–70.
- [8] C. C. Y. Poon, Yuan–Ting Zhang and Shu–Di Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and m–health," in IEEE Communications Magazine, vol. 44, no. 4, April 2006, pp. 73–81.
- [9] Masoud Rostami, Ari Juels, and Farinaz Koushanfar, "Heart–to–heart (H2H): authentication for implanted medical devices," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13), 2013, pp.1099–1112,
- [10] S. Cherukuri, K. K. Venkatasubramanian and S. K. S. Gupta, "Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in 2003 International Conference on Parallel Processing Workshop, 2003 Proceedings., Kaohsiung, Taiwan, 2003, pp. 432–439.
- [11] Pirbhulal, S.; Zhang, H.; Mukhopadhyay, S. C.; Li, C.; Wang,

- Y.; Li, G.; Zhang, Y. T. "An Efficient Biometric-Based Algorithm Using Heart Rate Variability for Securing Body Sensor Networks," *Sensors*, vol. 15, 2015, pp. 15067–15089.
- [12] K. K. Venkatasubramanian, A. Banerjee and S. K. S. Gupta, "PSKA: Usable and Secure Key Agreement Scheme for Body Area Networks," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, Jan. 2010, pp. 60–68.
- [13] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao and D. Chen, "OP FKA: Secure and efficient Ordered-Physiological-Feature-based key agreement for wireless Body Area Networks," in *2013 Proceedings IEEE INFOCOM*, Turin, 2013, pp. 2274–2282.
- [14] E. K. Zaghouani, A. Jemai, A. Benzina and R. Attia, "ELPA: A new key agreement scheme based on linear prediction of ECG features for WBAN," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Nice, 2015, pp. 81–85.
- [15] A. Kholmatov, B. Yanikoglu, "Realization of correlation attack against the fuzzy vault scheme," in *SPIE Security Forensics Steganography and Watermarking of Multimedia Contents X*, vol. 6819, Jan. 2008, pp. 1–7,
- [16] Preda Mihailescu, "The Fuzzy Vault for fingerprints is Vulnerable to Brute Force Attack", 2007, eprint arXiv:0708.2974,
- [17] Daesung Moon, Seung-Hoon Chae, Yongwha Chung, Sung-Y

- oung Kim, and Jeong-Nyeo Kim, "Robust Fuzzy Fingerprint Vault System against Correlation Attack," in Journal of the Korea Institute of Information Security and Cryptology, v.21, no. 2, 2011, pp. 13–15
- [18] PhysioBank, physionet.org/mimic2 (accessed on 19 November 2018)
- [19] Fen Miao, Shu-Di Bao and Ye Li, "Physiological Signal Based Biometrics for Securing Body Sensor Network," in IntechOpen, November 28th 2012.
- [20] Cho, K. and Chung, B., "Lightweight biometric key agreement scheme for secure body sensor networks," in International journal of communications, 2016, 218–222.
- [21] Hojoong Park, Ju-Sung Kang, Yongjin Yeom. "Probabilistic Analysis of AIS.31 Statistical Tests for TRNGs and Their Applications to Security Evaluations," in Journal of the Korea Institute of Information Security & Cryptology, 26.1, 2016.2, pp.49–67.
- [22] Latxtha <http://www.laxtha.com/ProductView.asp?Model=ubpulse%20340> (accessed on 19 November 2018)
- [23] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," in National Institute of Standards and Technology (NIST), special publication August 2008, 800–22.