Secure Key Agreement for Wearable Medical Devices

by

Alexander Kasparek

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

December 2019

APPROVED:

Professor Krishna Venkatasubramanian, Thesis Advisor

Professor Craig Shue, Thesis Reader

Professor Craig E. Wills, Head of Department

1 Abstract

In this thesis we explore if a proposed random binary sequence generation algorithm can be combined with a separately proposed symmetric key agreement protocol to provide usable security for communications in Wireless Body Area Networks (WBAN). Other previous works in this area fall short by only considering key generation between two of the same signals or allowing for key generation between two different types of signals but with the cost of a significant signal collection time requirement. We hoped to advance this area of research by making secure key generation more efficient with less signal collection time and allowing keys to be generated between two sensors that measure two different physiological signals. However, while the binary sequence generation algorithm and key agreement protocol perform well separately, they do not perform well together. The combined approach yields keys that have good properties for use in a WBAN, but the generation rate is low.

2 Acknowledgments

Thank you Professors Krishna Venkatasubramanian and Craig Shue for your guidance during the process of researching and writing my thesis. This was my first research project in the area of cybersecurity and one of the most challenging projects I have worked on, but it has also been the most rewarding project I have worked on. Through this process I have gained a deeper understanding of the processes and expectations of academic research. Thank you both for your parts in getting me to this point in my academic career.

Thank you to my family for your continued support and love. I would not be where I am today without each of you. I would also like to thank my roommates for always being fun to hang out with and helping me to decompress after the more stressful days.

Contents

1 Abstract 2
2 Acknowledgments 3
3 Introduction 5
4 Related Work 6
4.1 Physiological-Signal-Based Key Agreement (PSKA) . . . 6
4.2 SKA-PSAR 7
4.3 MFBSG 8
5 Background 9
5.1 HeartBeats-Based Security (HBBS) 9
5.2 Secure Key Agreement protocol using Physiological Signals (SKA-PS) 10
6 Methodology 12
7 Results 18
7.1 Goal 1: Key agreement for ECG-ECG 18
7.2 Goal 2: Key agreement for ECG-BP 27
7.3 Results Summary 31
8 Conclusions 33

3 Introduction

Wireless Body Area Networks (WBAN) are a collection of small wearable or implantable devices that can be used to monitor a patient's vital signs or even control them (e.g., pacemakers with remote operating or monitoring capabilities) [11]. Interference with the proper operation of the devices can result in delayed diagnosis and treatment, and potentially put a patient's life in serious danger.

Not only is data security important for patient safety, protecting health-related, personally identifiable information data for patients is required by law under the Health Insurance Portability and Accountability Act (HIPAA) [2]. For sensitive information that is stored or transmitted electronically, certain measures must be taken to ensure the security and privacy of the information.

In order to achieve authentication and integrity mechanisms in WBANs, it is prudent to take advantage of the physiological signals that can already be measured by each of the nodes that are attempting to communicate. Features of those signals can be used for the authentication and key agreement protocols. Interpulse interval (IPI), or heart rate, can be derived from many physiological signals and thus if two nodes measure two different signals, the IPI derived from those two signals during the same time will be similar. Measuring IPI is useful in the case where the two nodes setting up a secure communication session cannot measure the exact same signal.

As suggested in [19], there are four design goals for key generation schemes that utilize physiological signals:

1. *Length and randomness* - The keys that the two devices agree upon should be long enough and contain enough entropy to prevent an attacker from brute-forcing key values.
2. *Distinctiveness* - The features extracted from the signals of one subject are significantly different from the features extracted from another subject - i.e. knowing the feature set for one subject will not assist an attacker in deriving the keys generated by the same process for another subject.
3. *Temporal Variance* - Knowing the feature set derived during one round of key agreement (by knowing the physiological signals at that time) will not provide an advantage for deriving the keys agreed upon for future executions - i.e. an attacker cannot calculate features from a subject's signal measurements at one time and then use that feature set to derive the key calculated at a later time.
4. *Low latency* - The time needed for measuring and processing signals is minimized - while still meeting the other design goals. This is similar to a practical issue addressed in [11], "Conflict between security and safety."

There have been a number of solutions to this issue that have been proposed, such as [21], [5], [20]. These methods allow binary sequences to be extracted from physiological signals (such as electrocardiogram - ECG) so that they can be used for authentication of users and integrity of information transmitted over a WBAN.

4 Related Work

In previous studies ([14,16,17]), researchers have found that the four least significant bits (LSBs) from IPI measurements are suitable for use in authentication protocols in WBANs. In [5], Altop et al. propose two different methods for extracting binary sequences from physiological signals for use as cryptographic keys. The relevant method here is the time-domain technique. It uses IPI information calculated derived from signals such as ECG, ABP, and PPG. Then, a IPIs are summed in groups and circular uniform quantization is applied to each summed group to quantize each value into an s-bit integer. Each integer is then encoded using Gray encoding. For SKA-PS, Altop et al. choose to quantize each value into 4-bit integers. Building a feature set using this method and others that use the 4 LSBs from IPIs does produce sufficiently random binary sequences for use as cryptographic keys. However, in order to produce a 128-bit sequence, 32 IPI values (or 33 R-peaks of an ECG signal) are needed. Considering an average heart rate of 60 bpm (or 1 beat per second), this would require 33 seconds of sensing time.

In contrast, works such as [13] extracts 16 bits per IPI, thus necessitating only 8 IPI values (or 9 R-peaks of an ECG signal) for a 128-bit key. Considering the same average heart rate of 60 bpm, only 9 seconds of sensing time would be required. The following subsections describe in more detail some binary sequence extraction methods and key agreement protocols that have been proposed for use in WBANs.

4.1 Physiological-Signal-Based Key Agreement (PSKA)

The proposed method for secure key agreement is an extension of the physiological-signal-based key agreement (PSKA) scheme proposed in [20].

PSKA works as follows: first, the two nodes (the sender and the receiver) measure a physiological signal for a fixed amount of time at a fixed sampling rate. Features are extracted from the signals to generate the feature sets Fs (sender's feature set) and Fr (receivers feature set). During the feature extraction step, the communicating nodes of the WBAN calculate a windowed Fast Fourier Transform, which breaks the original signal into component sinusoidal signals which gives frequency and power properties for each component signal. These (frequency, power) pairs are then quantized into 13-bit representations by concatenating the binary representation of the frequency and power. Frequency is quantized into 8 bits and power is quantized into 5 bits. The sender then generates a random key and a polynomial of degree n.

Parts of the key act as the coefficients for the polynomial. The feature points are used as inputs for the polynomial to generate a set of input-output pairs P. Once the legitimate points on the polynomial have been calculated, the sender then generates a large number of random points that do not lie on the polynomial (chaff points) as the set C. Vault R is then generated by randomly permuting the points in C and P so that the two sets are indistinguishable.

The sender generates a message containing IDs, IDr, R, *No*, MAC(Key, R

$|No$ |IDs), where IDs and IDr are the ids of the sender and the receiver, and No is a nonce (unique random number). The sender then transmits this message to the receiver.

When the receiver receives the vault, it computes the feature points from the physiological signal that it measured. It attempts to unlock the vault using the Lagrangian interpolation which allows a vth-order polynomial to be reconstructed as long as the receiver knows at least v+1 points on the polynomial. From its feature set, it takes v + 1 points at a time and attempts to unlock the vault, finishing when it can successfully verify the MAC. The receiver then sends an acknowledgement to the sender in the form of a MAC.

[20] only considers using the same physiological signal at the sender and receiver side. Because FFT measures spatial properties of a signal it works well when the receiver and the sender measure the same signal. However, two different signals such as ECG and ABP will have different spatial properties and so works in this space consider temporal properties of the signals that keys are being derived from.

4.2 SKA-PSAR

SKA-PS with Augmented Randomness (SKA-PSAR) is a key agreement protocol outlined in [18] that builds on SKA-PS. After calculating the IPI values for the signals and grouping them, SKA-PSAR runs its set reconciliation protocol without quantizing the IPI values first. After the IPI sets are reconciled, the receiving sensor quantizes, binarizes, and gray encodes the resulting IPI sets into

$$IPIb_1...IPIb_n$$

, where IPIb denotes the binary representation of the IPI value. If, in

$$IPI_1...IPI_n$$

, there are equivalent IPI values, SKA-PSAR appends extra bits to the end of

$$IPIb_2...IPIb_n$$

in order to increase the randomness of the resulting key.

After the receiver generates its key with the reconciled IPI sequence, the receiver computes an HMAC on a message m with the key and sends it to the sender. The sender, because it doesn't know exactly which IPI values were reconciled, builds possible keys by following the same quantization, binarization, and gray encoding steps as the receiver to get a set of possible keys. It generates the HMACs with the possible keys on the same message m until a matching HMAC is computed. Finally, the sender computes the HMAC of a different message n and sends it to the receiver for verification.

4.3 MFBSG

In their work in [21], Zheng et. al. propose a binary sequence generation algorithm that utilizes 5 features of ECG signals to extract 16 bits in total from one ECG cycle:

1. RR Interval: 4 bits
2. RQ Interval: 3 bits
3. RS Interval: 3 bits
4. RP Interval: 2 bits
5. RT Interval: 4 bits

In order to collect this information from each ECG cycle, there are three rounds of processing that are applied, whereas in [13], only one round of processing needs to be applied to detect R peaks. Another shortcoming of [21] is that only sensors that can measure ECG signals would be able to use this technique to generate binary sequences. The advantage of [13] is that any sensor that measures a signal from which IPI can be derived (such as ECG, BP, and PPG) can use its technique to derive binary sequences.

5 Background

5.1 HeartBeats-Based Security (HBBS)

[13] is the inspiration for this research. The authors propose an efficient binary sequence generation algorithm to produce binary sequences that are suitable for use as cryptographic keys.

For binary sequence extraction from ECG signals, the authors propose the following approach:

1. *R-peak detection*
   First, the authors use their proposed algorithm for detecting the R-peaks of a given ECG signal.
2. *IPI Generation*
   n IPI values are calculated by taking the difference in time of occurrence between successive n+1 ECG values as in Equation 1:

$$IPI(n) = \sum_{i=2}^{i=n+1} ECG(i) - ECG(i-1) \tag{1}$$

3. *FMIS Generation* A finite monotonic increasing sequence (FMIS) is generated. An FMIS is defined by a sequence N such that $N(x+1) \geq N(x)$. The authors generate the FMIS by summing consecutive IPIs, as shown in Equation 2:

$$FMIS(n) = \sum_{i=0}^{i=n} IPI(i) \tag{2}$$

4. *Cyclic Block Encoding*
   After the FMIS is generated, each element of the sequence is encoded into a 23-bit codeword using cyclic block encoding. By using cyclic block encoding, more bits can be extracted from each IPI.
5. *Binary Sequence Extraction*
   Finally, from each 23-bit codeword, the authors extract 16 bits by taking bits 5-16. In order to generate the random binary sequence, each 16-bit sequence extracted from the IPI values are concatenated together.

Security of HBBS

In order to test their approach, the authors collected ECG signals from 89 subjects, including some who were healthy, some with arrhythmia (collected from the MIT-BIH Arrhythmia database [8]), and some with other various cardiac diseases. From each subject, a series of 128-bit RBSs were generated using 8 IPIs per FMIS and with 16 bits extracted from each IPI.

The authors calculated the average Hamming distances between RBSs generated from different subjects. Hamming distance is the average number of

bits between two binary sequences that differ. If the RBS generation technique generates binary sequences that are truly random, the Hamming distance between any two different binary sequences will be 0.5. The results of the average hamming distances are as follows:

Table 1: Average Hamming Distance

Data Source	Average Hamming Distance
Healthy Subjects	0.495
MIT-BIH Database Subjects	0.493
Cardiac Patients	0.486

The RBSs were also run through 9 of the NIST randomness tests. To pass each of the 9 tests, each binary sequence must produce a p-value where $p - value > 0.01$. The results were as follows:

Table 2: Results of NIST Randomness Tests (P-values)

Test No.	Test Name	Healthy Subjects	MIT-BIH	Cardiac Patients
1	F-Test	0.782	0.758	0.769
2	N-Test	0.993	0.990	0.986
3	B-Test	0.764	0.755	0.761
4	R-Test	0.862	0.721	0.698
5	LR-Test	0.217	0.304	0.168
6	FFT-Test	0.038	0.026	0.031
7	Lc-Test	0.963	0.956	0.915
8	AE-Test	0.997	0.995	0.992
9	C-Test	0.791	0.786	0.704

Following the results from the Tables 1 and 2, the binary sequences generated using the proposed method pass different measures of randomness.

5.2 Secure Key Agreement protocol using Physiological Signals (SKA-PS)

For our key-agreement protocol, we use SKA-PS which was built for key agreement between devices that measure different physiological signals. Secure Key Agreement protocol using Physiological Signals (SKA-PS) is a scheme for secure key agreement in Body Area Networks proposed in [6].

SKA-PS follows a set reconciliation paradigm. Set reconciliation is the process of taking two sets and reconciling their differences so that one set becomes equivalent to the other. SKA-PS set reconciliation is the process of finding missing set elements by representing the sets in a specified prime field Fq as characteristic polynomials, evaluating those polynomials on a set of given points, and then

using rational interpolation so that a set B can be reconciled with a set A.

In SKA-PS, one sensor (the source biosensor, or sender) captures one physiological signal, and then processes it according to the previous steps. Every group of successive 4-bits are converted to their decimal representations to create the set A. Set A is then further broken up into sets of a specified length s, and from each set a characteristic polynomial is created over the prime field Fq using the set elements as the roots of the polynomial. A number of the sets, r, are sorted in ascending order and then concatenated together to generate a key Ks. Each set's characteristic polynomials are evaluated over a set of evaluation points E and these evaluations along with HMACs of a publicly-known message m are sent to the conforming biosensor (receiver).

The conforming biosensor measures a separate physiological signal and in the same way as the source biosensor, processes it and grouped into 4-bit numbers to create the set B. It then also evaluates its characteristic polynomials using the same evaluation point set E as the source biosensor. It then uses the polynomial evaluations that it receives and calculates DE which is the division of the conforming sensor's evaluations by the source's evaluations. It then attempts to perform rational interpolation using DE and E as inputs. If the source and conformer's IPI values are close enough, a rational function can be evaluated and it uses the roots of the rational function to remove and add set elements as needed so that its set matches exactly with the source's set. It then sorts each set of length s and uses them to create a key Kc. If the HMAC produced using m (same as the source) matches one of the HMACs received by the conformer, the conformer sends an acknowledgement indicating which key was agreed upon.

Security of SKA-PS

In the authors' evaluations, SKA-PS achieved high TAR and a low FAR. The average randomness of the generated keys, which was measured using Shannon Entropy, was between 0.854 and 0.863.

To determine the distinctiveness of the generated keys, the authors calculated the Hamming Distance between keys generated by two different subjects. It is expected that two different keys from two different subjects (distinctiveness), or two different keys generated for the same subject at a different time (temporal variance), would have a Hamming Distance of 0.5. SKA-PS achieves a distinctiveness of between 0.49 and 0.5. However, the keys generated by SKA-PS achieve a temporal variance of between 0.164 and 0.191. This means that between two keys generated at two different times, only between 16.4% and 19.1% of the bits were different, far lower than the expected 50% if the keys were perfectly random.

As is stated in the paper, although keys are lacking distinctiveness, an attacker does not know which bits between two keys will be the same even if he or she knows a previously generated key. The authors observe that such an attack does reduce brute force complexity, but it is still a minimum of 2^{116} .

6 Methodology

Because the authors of [13] did not propose a key agreement protocol to complement the binary sequence generation, we decided to determine if, when paired with SKA-PS for key agreement:

1. Key agreement is possible using only ECG measured from different leads.
2. Key agreement is possible using ECG and another signal (for this research, blood pressure (BP)). ECG and BP signals are both measures of heart activity and can be used to calculate the inter-pulse interval (IPI), a measure of heart rate variability.

1. *Data gathering* We gathered data from two publicly available signal recording databases located at Physionet [4].
   To determine if key agreement is successful in goal 1, we downloaded 16 readings of two leads of ECG signals from 13 different subjects from the European ST-T Database (EDB) [3]. EDB contains ECG recordings, each reading is two hours long and sampled at 250 samples per second.
   To determine if key agreement is successful in goal 2, we downloaded the ECG and BP signals for 16 subjects from the publicly-available Fantasia database [12]. The Fantasia database contains signal readings from 40 healthy subjects, with half being 21-34 years old and half being 68-85 years old. Only 20 of the 40 subjects had readings for blood pressure in addition to ECG, and we found that our peak-detection algorithm worked well for 16 subjects of those 20 with both signal readings.
   For both goals, the following steps were followed in a similar fashion:
2. *IPI Generation*
   First, we generate IPI values.
   ECG signals were run through a generic Pan-Tompkins R peak detection function [15]. IPIs are calculated from the difference in time between successive R peaks.

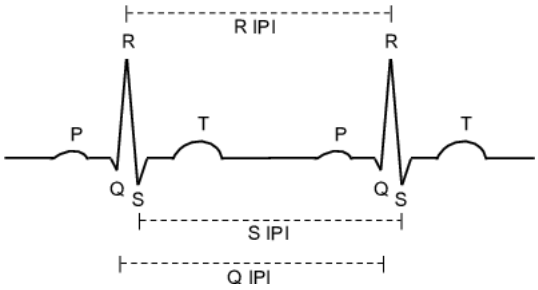


Figure 1: R peaks of an ECG signal, [1]

We run BP signals (goal 2) through a peak detection function from [9]

to detect systolic peaks. IPIs are calculated from the difference in time between successive systolic peaks.

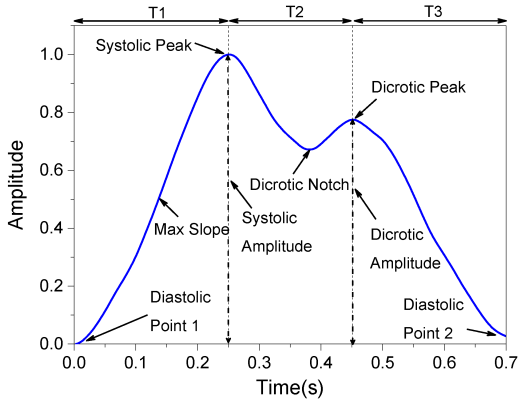


Figure 2: Systolic peaks of a BP signal, [10]

Below are examples of peak detection on an ECG signal and a BP signal that we ran on data that we collected using MATLAB.

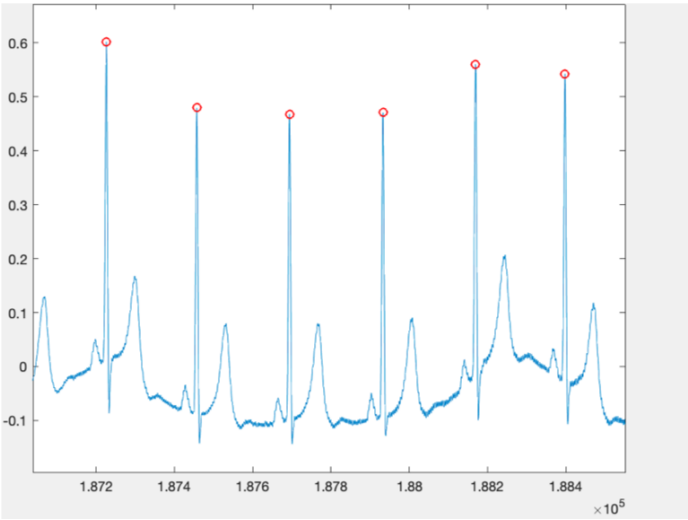


Figure 3: R peaks of an ECG signal, marked with red circles

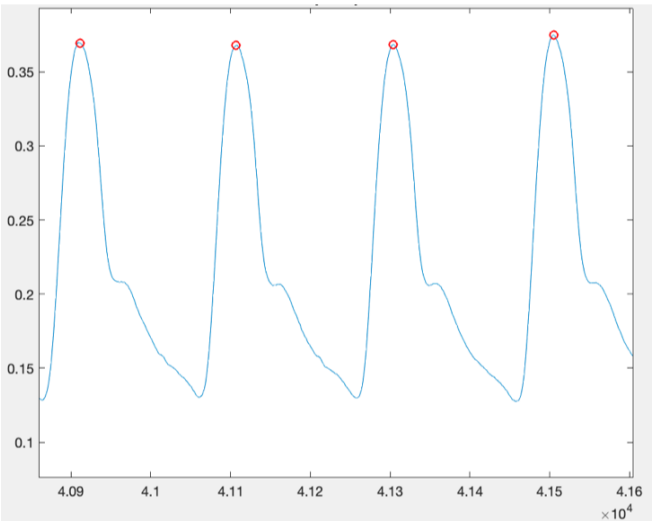


Figure 4: Systolic peaks of a BP signal, marked with red circles

3. *Finite Monotonic Increasing Sequences Generation*
   After x IPIs are calculated from the signal, a finite monotonic increasing sequence (FMIS) is generated using (2). In our testing, we tried x = 3:16 in order to determine how constructing the FMIS impacts key agreement.

4. *Cyclic Block Encoding*
   Previous work for extracting random binary sequences from IPI data have suggested using the four least significant bits due to their randomness properties. However, in order to generate a minimum of 128-bits for use in a symmetric key agreement scheme, this requires a minimum of 32 IPIs, or the detection of 33 R peaks for ECG signals and (in the case of goal 2) 33 systolic peaks for BP signals.
   Gaining inspiration from [13], we hoped to reduce the number of IPIs required to generate these binary sequences using systematic cyclic encoding to expand the number of bits that can be extracted from each IPI value. Using the parameters n (minimum output code length), and k (length of the cyclic polynomial), we can expand the binary representation of an IPI value of x bits into an encoded form with a minimum of n bits.
   For our purposes, we take each IPI value and encode it to a minimum of 23 bits using this procedure.

5. *Binary Feature Extraction*
   Now, from each expanded IPI bit representation, we extract a number of bits from each to use in SKA-PS as the key agreement protocol.
   From each 23-bit binary representation, we extract bits i through j. We tested a range of values for i and j for both goals.

6. *Key Agreement*

For key agreement testing using our binary sequence generation method, we use SKA-PS which was described earlier. However, we do not use the full protocol. Given r (number of sets that must match between the two sequences), s (the number of elements in each of the r sets), and d (the allowable number of differences between a set from the sender and receiver), we simply check if, between the two sets A and B, there are at least r subsets of length s that match within d differences. This would mean that running the full SKA-PS protocol would result in a matched key. If there are not at least r subsets of length s that match within d differences, the full SKA-PS protocol would fail to generate a symmetric key.

SKA-PS represents each set element of feature sets produced from signals as 4-bit integers. This means that for each subset between the two signals, up to $4 * d$ bits can be reconciled, or up to $4 * r * d$ total bits. We tried increasing and decreasing the bit-length representation of each element to determine how key agreement rates would be affected. In order to make sure that at least 128-bit symmetric keys would be generated, Equation 3 must hold, where b is the bit-length representation of each feature point:

$$(b * r * s) - (b * r * d) = b * r * (s - d) >= 128 \tag{3}$$

Equation 4 must hold as well to prevent information leakage, as each set in the SKA-PS protocol is represented using a polynomial and sets are reconciled using rational function interpolation.

$$d <= floor(\frac{s-1}{2}) \tag{4}$$

When choosing our values for r, s, and d for the various values of b, we made sure that equations 3 and 4 held. Also, chose s so that d could be at least 1 so that some elements were reconcilable. We also chose those parameters that would allow for the most possible sets of size s so that there were more sets to use to potentially agree on a key.

SKA-PS requires that each set of size s is sorted in ascending order before binarization and concatenation can be performed to generate possible keys. This is because order of the elements in a set cannot be determined at the receiver/conforming biosensor. For example, if one set during a round of SKA-PS on the sender side is 5, 2, 9 and the set on the receiver side is 9, 2, 5, those sets contain all of the same elements, but the receiver, given the information it receives from the sender, cannot determine that it should swap elements 5 and 9 so that it is exactly the same as the set on the sender side. Therefore, the set 2, 5, 1 is the same as 5, 1, 2. To reduce the impact of this property of SKA-PS, the authors limit s, and thus the potential values for d are also limited since d is bounded by s.

With all of this in mind, the table below shows our chosen values for r, s, and d based on b. For b = 2, we set s to 2 because with two bits, each element in the set can be chosen from only four numbers (0, 1, 2, 3).

Table 3: r, s, d based on b)

b	r	s	d
2	32	2	0
3	15	4	1
4	11	4	1
5	9	5	2
6	6	5	1
7	4	7	1
8	4	5	1

7. *Metrics* For each subject we generate all of the possible binary sequences that can be used for key agreement for the given length of the signal. We simulate the key agreement as explained in the previous section, calculating the false acceptance rate (FAR), false rejection rate (FRR), true acceptance rate (TAR), true rejection rate (TRR), the hamming distance between two synchronized signals for the same subject, the hamming distance between two desynchronized signals for different subjects, and the hamming distance between two signals for different subjects (one ECG and one ABP).

   (a) *TAR* The true acceptance rate is the average number of binary sequences generated from synchronized ECG and ABP signals from one subject that, when run through SKA-PS, produce a matching key.

   In order to calculate the TAR, given the generated RBSs, we run each synchronized signal through our shortened SKA-PS algorithm. The result is the percent of successful key generations.

   (b) *TRR* The true rejection rate is the average number of binary sequences generated for two subjects' signals or from one subject's desynchronized signals that fail to produce a matching key. A true rejection means that either an impersonation or replay attack has failed.

   In order to calculate the TRR, given the generated RBSs, we run each desynchronized pair of signals for one subject and each pair of signals from different subjects through our shortened SKA-PS algorithm. The result is the percent of unsuccessful key generations.

   (c) *FAR* The false acceptance rate is the average number of binary sequences generated for two subjects' signals or from one subject's desynchronized signals that (erroneously) produce a matching key. A false acceptance means either an impersonation or replay attack has succeeded, which is undesirable.

   In order to calculate the FAR, we determine the percent of successful key generations when the two signals are either desynchronized or

from two different subjects.

(d) *FRR* The true acceptance rate is the average number of binary sequences generated from synchronized ECG and ABP signals from one subject that, when run through SKA-PS, erroneously fail to produce a matching key. A false rejection means that two legitimate signals (synchronized and from the same user) fail to produce a matching key.

In order to calculate the FRR, we determine the percent of unsuccessful key generations when the two signals are both synchronized and from the same subject.

(e) *Hamming Distances* As a measurement of temporal variance, we use Hamming distance in three different ways. First, we calculate the Hamming distances between synchronized binary sequences from the same subject. In order for key agreement to be successful, this be as close to zero as possible. Second, we calculate the Hamming distances between desynchronized binary sequences from the same subject. In order for the feasibility of replay attacks to be reduced, this should be as close to 0.5 as possible. Third, we calculate the Hamming distances between binary sequences signal 1 and signal 2 generated from different subjects. In order for the feasibility of impersonation attacks to be reduced, this should be as close to 0.5 as possible.

We started with measuring all metrics we can gather when comparing one subject's data against itself for TAR and FAR. Composing the FAR metric in this case is failure to generate a key when two different signals collected at different times are used or when a reading from one signal recorded at a given time is used with the same signal recorded at a different time.

From this initial measurement, for the combinations that performed the best, we gathered metrics on key agreements between signals recordings from different subjects.

7 Results

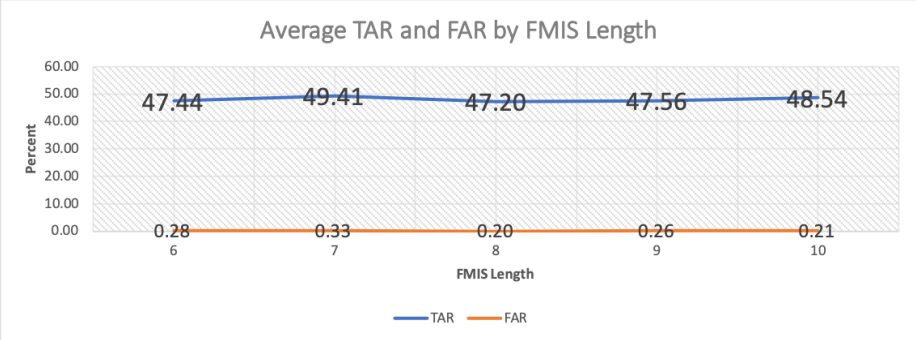
The following sections breaks down the results of our work with [13] into results for each of the two goals, key agreement for two ECG signals and key agreement for ECG and BP signals.

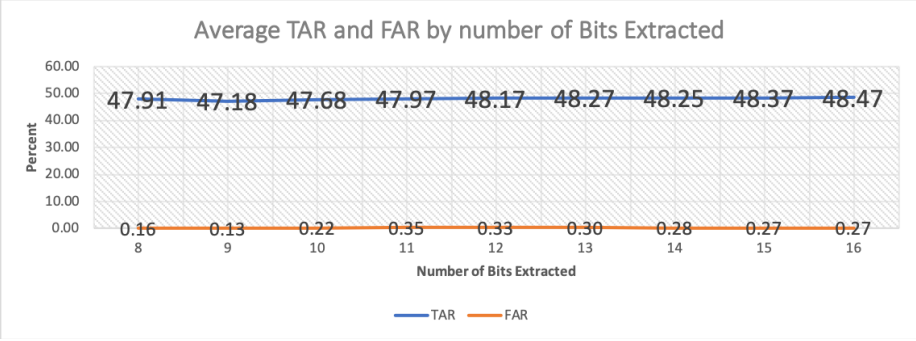
We began by calculating the average TAR and FAR for various values of r, s, d, and b for ECG-ECG and ECG-BP key agreement. For each value of b, we show the average TAR and FAR by FMIS length and by the number of bits extracted from each IPI in the sections marked "TAR and FAR - One Subject".

7.1 Goal 1: Key agreement for ECG-ECG

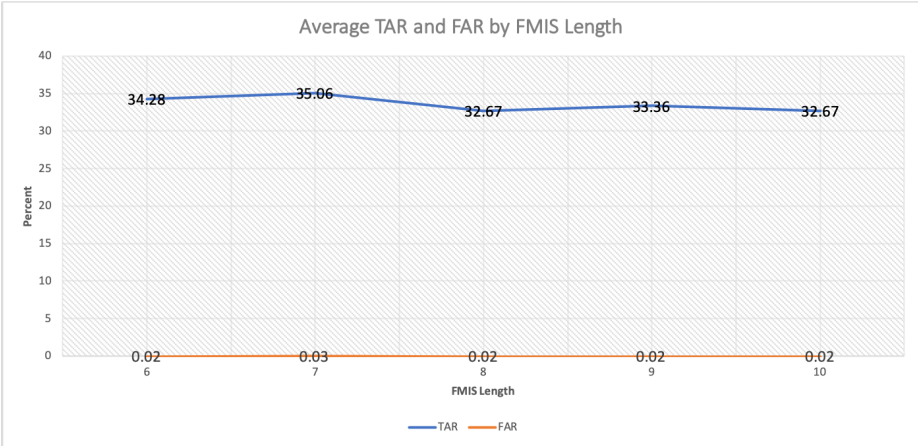
TAR and FAR - One Subject

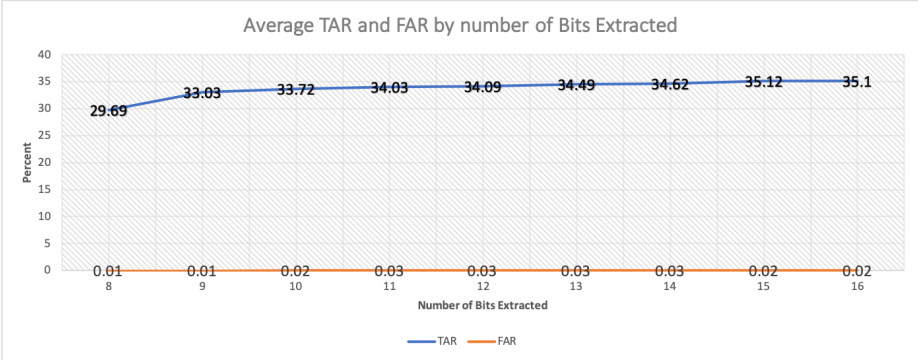
1. b = 2



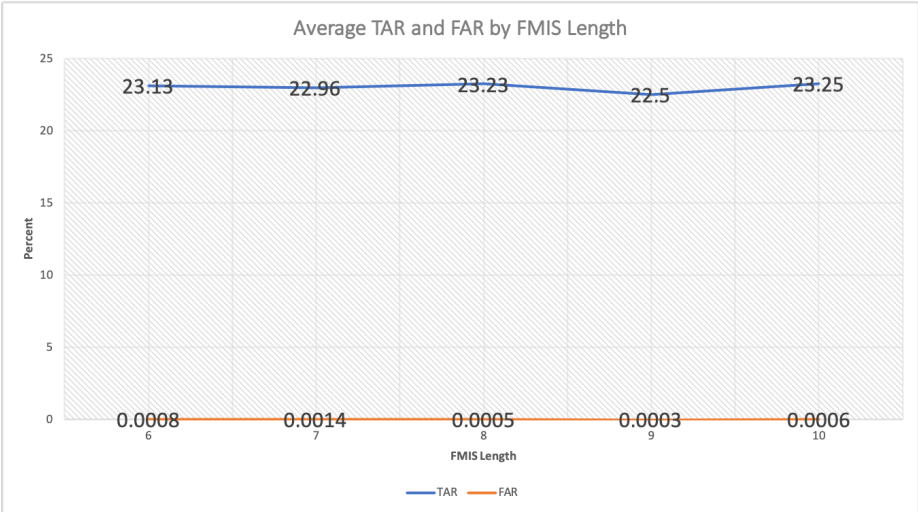


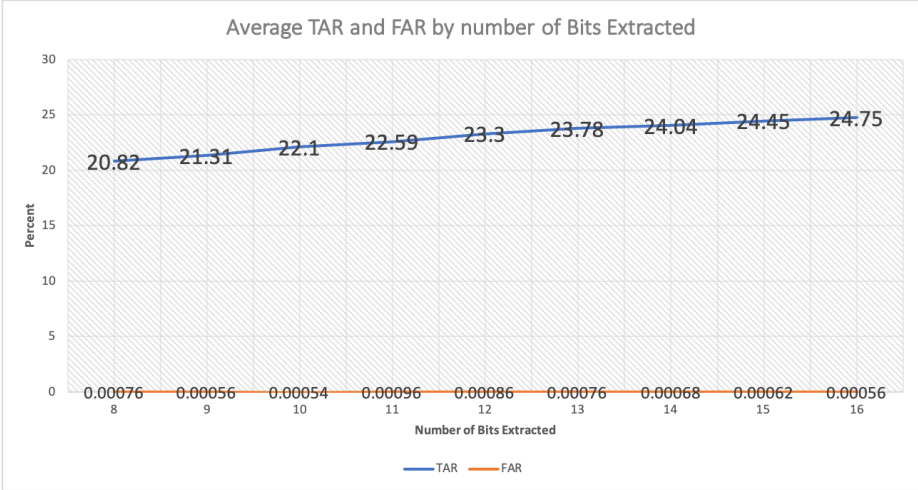
2. b = 3


Average TAR and FAR by FMIS Length
Percent
40
35
30
25
20
15
10
5
0
34.28
35.06
32.67
33.36
32.67
0.02
0.03
0.02
0.02
0.02
6
7
8
9
10
FMIS Length
TAR
FAR

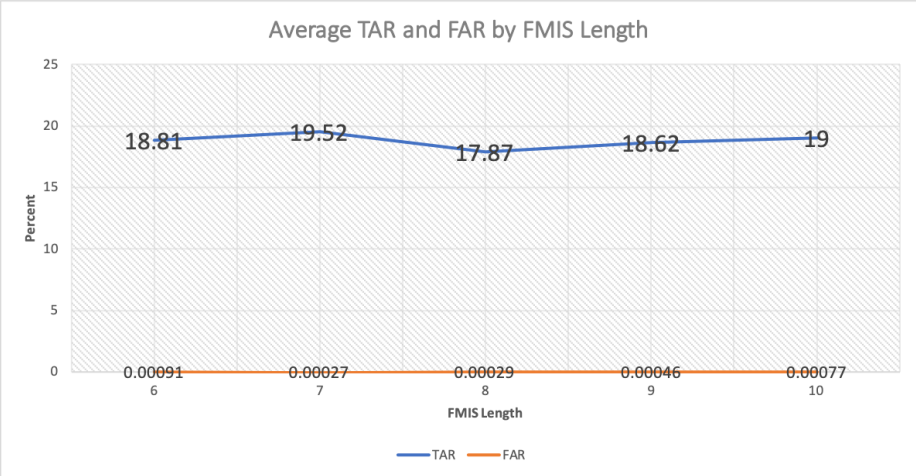

Average TAR and FAR by number of Bits Extracted
Percent
40
35
30
25
20
15
10
5
0
29.69
33.03
33.72
34.03
34.09
34.49
34.62
35.12
35.1
0.01
0.01
0.02
0.03
0.03
0.03
0.03
0.02
0.02
8
9
10
11
12
13
14
15
16
Number of Bits Extracted
TAR
FAR

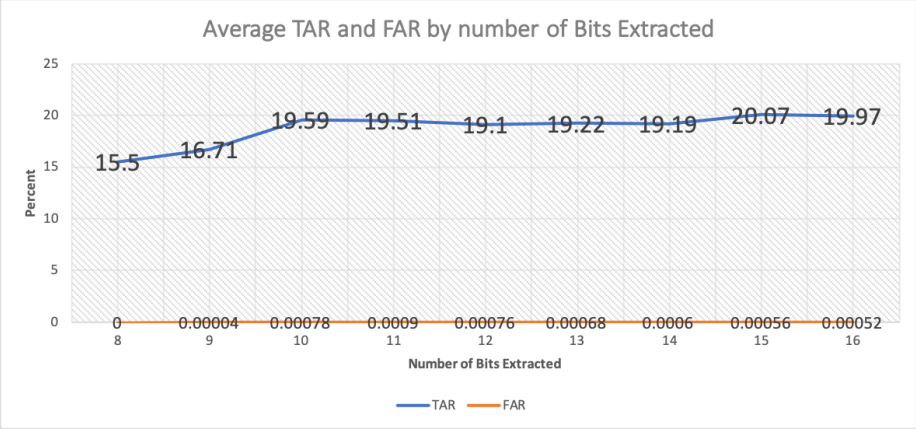
3. b = 4


Average TAR and FAR by FMIS Length
25
20
15
10
5
0
Percent
23.13
22.96
23.23
22.5
23.25
0.0008
0.0014
0.0005
0.0003
0.0006
6
7
8
9
10
FMIS Length
TAR
FAR

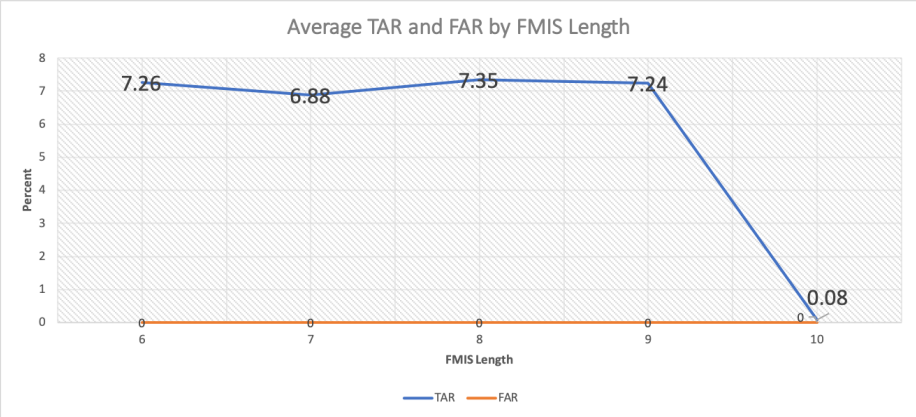

Average TAR and FAR by number of Bits Extracted
30
25
20
15
10
5
0
Percent
20.82
21.31
22.1
22.59
23.3
23.78
24.04
24.45
24.75
0.00076
0.00056
0.00054
0.00096
0.00086
0.00076
0.00068
0.00062
0.00056
8
9
10
11
12
13
14
15
16
Number of Bits Extracted
TAR
FAR

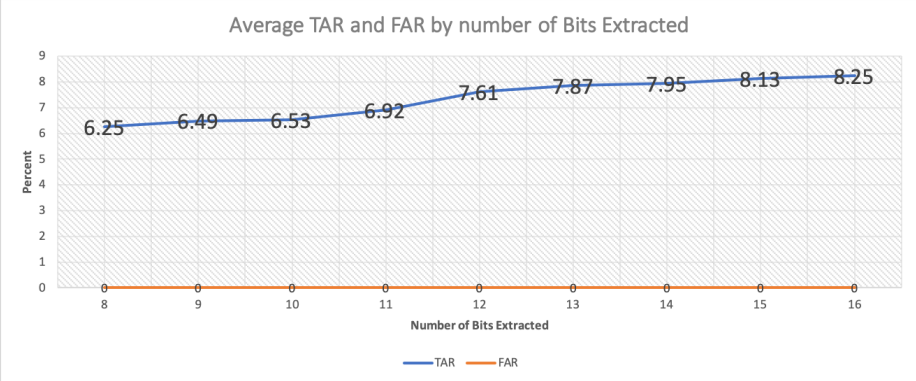
4. b = 5


Average TAR and FAR by FMIS Length
25
20
15
10
5
0
Percent
18.81
19.52
17.87
18.62
19
0.00091
0.00027
0.00029
0.00046
0.00077
6
7
8
9
10
FMIS Length
TAR
FAR

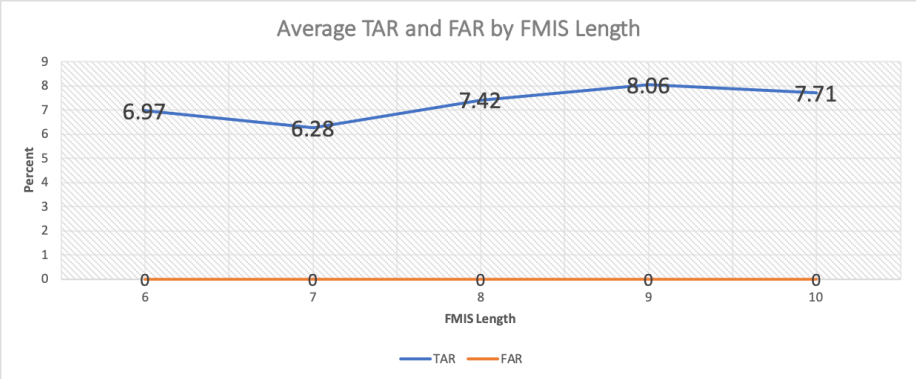

Average TAR and FAR by number of Bits Extracted
25
20
15
10
5
0
Percent
15.5
16.71
19.59
19.51
19.1
19.22
19.19
20.07
19.97
0
0.00004
0.00078
0.0009
0.00076
0.00068
0.0006
0.00056
0.00052
8
9
10
11
12
13
14
15
16
Number of Bits Extracted
TAR
FAR

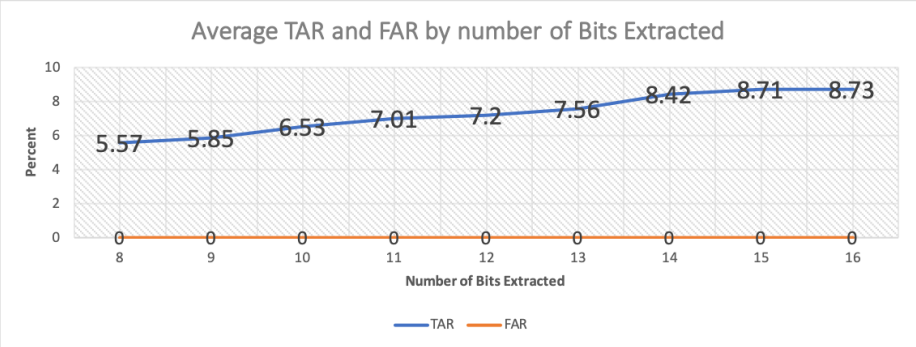
5. b = 6


Average TAR and FAR by FMIS Length
Percent
FMIS Length
7.26
6.88
7.35
7.24
0.08
TAR
FAR

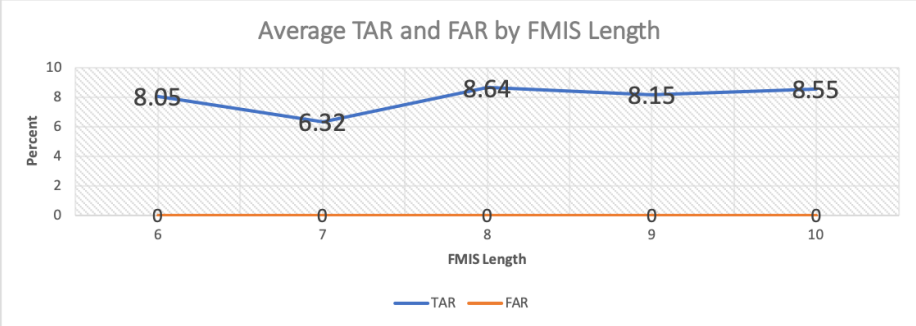

Average TAR and FAR by number of Bits Extracted
Percent
Number of Bits Extracted
6.25
6.49
6.53
6.92
7.61
7.87
7.95
8.13
8.25
TAR
FAR

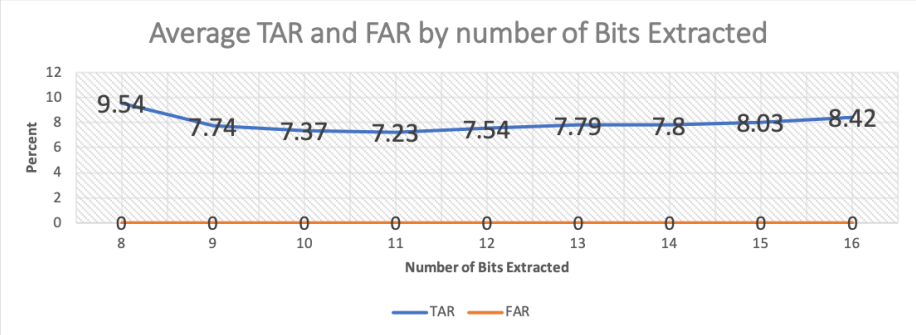
6. b = 7


Average TAR and FAR by FMIS Length
Percent
9
8
7
6
5
4
3
2
1
0
6.97
6.28
7.42
8.06
7.71
0
0
0
0
0
6
7
8
9
10
FMIS Length
TAR
FAR


Average TAR and FAR by number of Bits Extracted
Percent
10
8
6
4
2
0
5.57
5.85
6.53
7.01
7.2
7.56
8.42
8.71
8.73
0
0
0
0
0
0
0
0
0
8
9
10
11
12
13
14
15
16
Number of Bits Extracted
TAR
FAR

7. b = 8





As can be seen, as b increased, the false acceptance rate decreased, but so did the true acceptance rate. This could be an indication that the differences in bits between the two signals used for agreement were more spread out. A smaller b value yields more sets for key agreement, and thus the bits that are reconciled between two binary sequences have a greater spread.

We looked at parameter combinations that achieved at least a 50% TAR and a less than 10% FAR. Those combinations are displayed below. There were no parameter combinations that achieved a 50% TAR and less than 10% FAR for b = 6, 7, and 8.

Table 4: Best individual results

b	FMIS Length	Num bits	start bit	end bit	length RBS	TAR	FAR
2	8	11	13	23	352	83.75%	8.21%
3	9	11	13	23	396	62.60%	1.49%
4	9	11	13	23	396	50.99%	0.0353%
5	10	10	13	22	400	52.98%	0.0217%

Temporal Variance of Binary Sequences

We looked at the temporal variance properties for the most promising parameters for each binary sequence. We determined the temporal variance by calculating the Hamming distances between desynchronized signals and between two binary sequences that were generated from the same signal at different times.

Table 5: Temporal Variance

b	ECG1-ECG2	ECG1-ECG1	ECG2-ECG2
2	0.361	0.360	0.362
3	0.364	0.363	0.365
4	0.364	0.363	0.365
5	0.379	0.379	0.379

We achieved good temporal variance for each tested value of b. It increased slightly as b increased.

Randomness of Binary Sequences

For the entire length of the binary sequence generated we took a measure of Shannon Entropy to get an idea of the randomness of the generated sequences. The code we used to calculate Shannon Entropy is from [7].

Table 6: Entropy of ECG1 and ECG2 signals

b	entropy ECG1	entropy ECG2
2	0.9771	0.9984
3	0.9769	0.9987
4	0.9769	0.9987
5	0.9733	0.9984

Impersonation Resistance (Distinctiveness) of Binary Sequences

In order to get an idea of how well the generated binary sequences resist impersonation attacks, we calculated the Hamming distance measurement between the binary signals collected from one subject and those collected from another subject.

Table 7: Impersonation Resistance

b	ECG1-ECG2	ECG1-ECG1	ECG2-ECG2
2	0.404	0.404	0.404
3	0.405	0.405	0.405
4	0.405	0.405	0.405
5	0.399	0.399	0.398

The generated binary sequences for the different sets of parameters had similar distinctiveness properties. While short of the ideal 0.5, the binary sequences are still distinct from subject to subject.

Randomness of Derived Keys

For each key that was generated we calculated Shannon Entropy.

Table 8: Entropy of Keys

b	entropy of keys
2	0.9642
3	0.9706
4	0.9717
5	0.9697

The combination of the two approaches produces keys with good entropy properties. The ideal Shannon entropy would be 1 to indicate perfect entropy, but the results for each parameter set for the values of b are close to 1. The average entropy for the generated keys across all of the b values are higher than the average entropy of keys generated by SKA-PS.

Temporal Variance and Distinctiveness of Derived Keys

If an attacker knows one key, he or she should not be able to derive later keys. Also, an attacker should not be able to use the knowledge of subject A's keys to compromise subject B's keys. We calculated the Hamming Distance between every pair of different keys from the same subject and from different subjects to measure temporal variance and distinctiveness, respectively.

Table 9: Temporal Variance and Distinctiveness of Keys

b	Temporal Variance	Distinctiveness
2	0.370	0.380
3	0.371	0.378
4	0.397	0.404
5	0.356	0.363

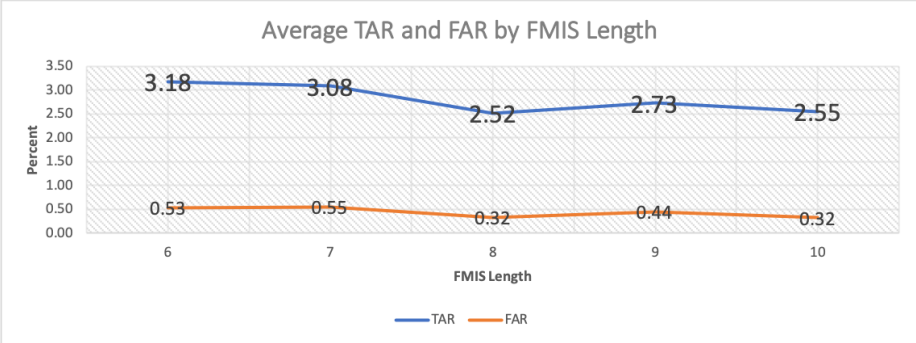
The keys generated keys possess some measure of temporal variance and distinctiveness, although it is short of the ideal 0.5. SKA-PS achieved a distinctiveness of around the ideal 0.5 for all of the parameters that were tested, but only around 0.2 for temporal variance. The combination of HBBS and SKA-PS did better in terms of temporal variance of the keys but did not perform as well in terms of distinctiveness.

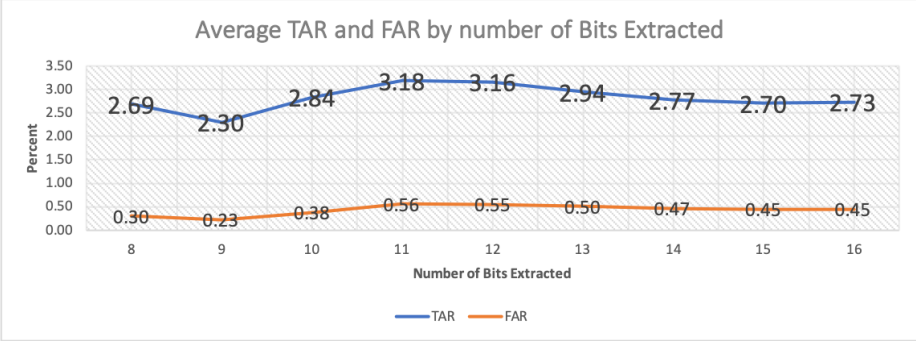
7.2 Goal 2: Key agreement for ECG-BP

TAR and FAR - One Subject

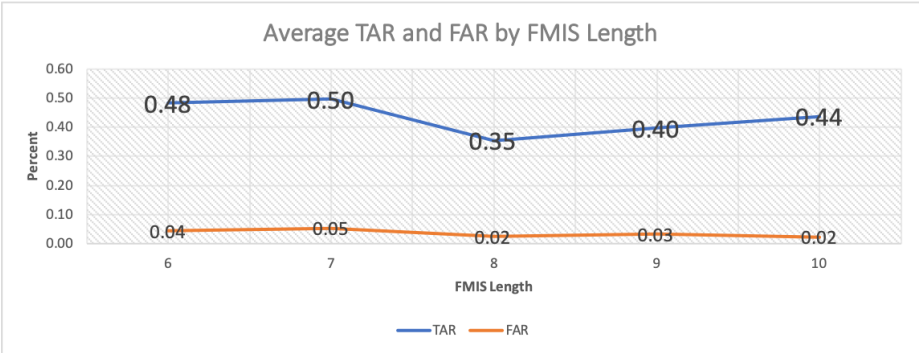
As for Goal 1, we calculated the average TAR and FAR for various values of r, s, d, and b when looking at the ECG and BP signals of one subject. Below are graphs that show results for b = 2, 3, 4, and 5. We then took the most promising results as for ECG-ECG keys and moved on to further tests.

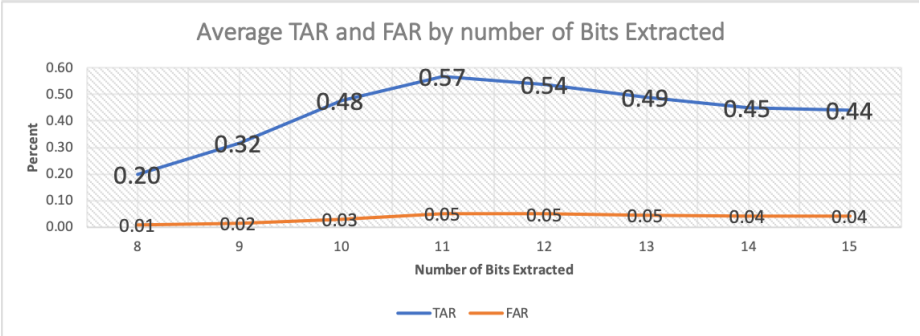
1. b = 2



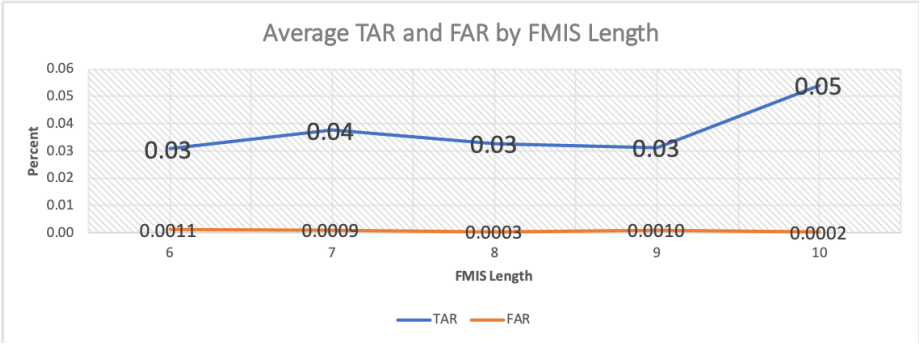


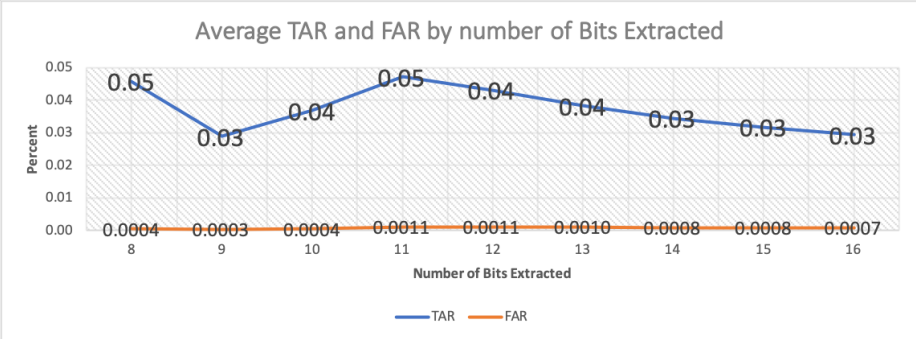
2. b = 3


Average TAR and FAR by FMIS Length
Percent
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.48
0.50
0.35
0.40
0.44
0.04
0.05
0.02
0.03
0.02
6
7
8
9
10
FMIS Length
TAR
FAR

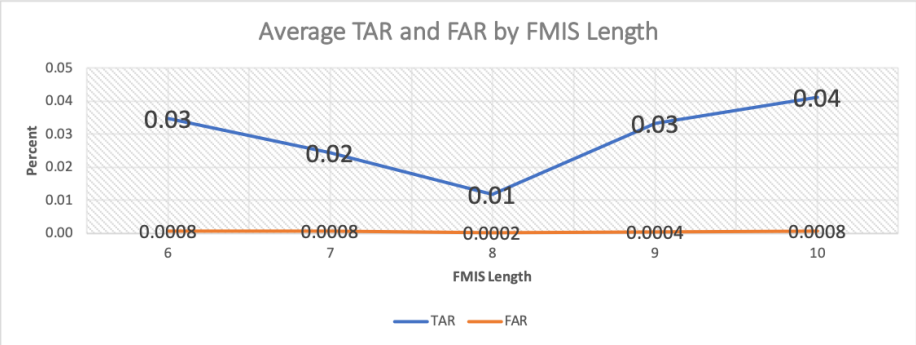

Average TAR and FAR by number of Bits Extracted
Percent
0.60
0.50
0.40
0.30
0.20
0.10
0.00
0.20
0.32
0.48
0.57
0.54
0.49
0.45
0.44
0.01
0.02
0.03
0.05
0.05
0.05
0.04
0.04
8
9
10
11
12
13
14
15
Number of Bits Extracted
TAR
FAR

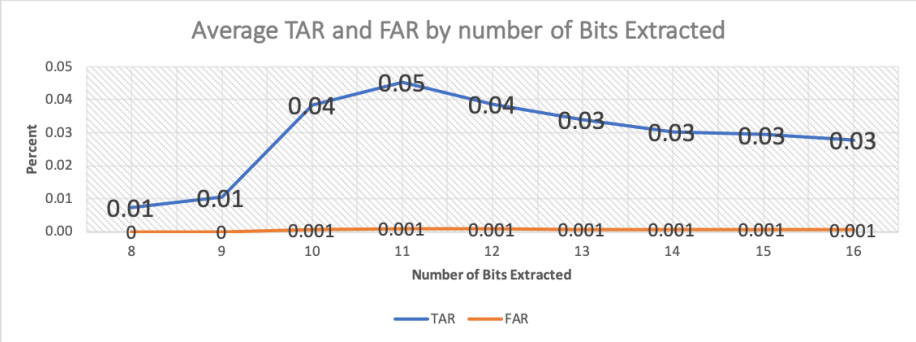
3. b = 4


Average TAR and FAR by FMIS Length
Percent
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0.03
0.04
0.03
0.03
0.05
0.0011
0.0009
0.0003
0.0010
0.0002
6
7
8
9
10
FMIS Length
TAR
FAR



4. b = 5





For b = 6, 7, and 8, the average TAR and FAR both fell to 0. The same overall trend as for ECG-ECG keys is observed, with both the TAR and FAR falling as b increases. However, it is interesting to note that similar TAR and FAR for b = 4 and b = 5 are achieved.

For ECG-BP keys, no parameter set we used achieved a TAR of above 50%. The best result is below:

Table 10: Best individual results

b	FMIS Length	num bits	start bit	end bit	length RBS	TAR	FAR
2	8	11	13	23	352	44.8%	11.8%

This is the only parameter combination for ECG-BP key agreement that we moved on with.

Temporal Variance of Binary Sequences

Table 11: Temporal Variance

b	ECG-BP	ECG-ECG	BP-BP
2	0.363	0.362	0.362

While below the ideal 0.5, the combinations of signals achieve good temporal variance which led to low false acceptance rates when compared to the true acceptance rates.

Randomness of Binary Sequences

Table 12: Entropy of ECG and BP signals

b	entropy ECG	entropy BP
2	0.9702	0.9986

The binary sequences generated with the ECG and BP signals for this parameter combination have high entropy, with the binary sequences generated from the blood pressure signal having slightly higher entropy on average than the binary sequences generated from the ECG signal.

Impersonation Resistance (Distinctiveness) of Binary Sequences

Table 13: Impersonation Resistance

b	ECG-BP	ECG-ECG	BP-BP
2	0.416	0.416	0.415

While below the ideal 0.5, the combinations of signals achieve good impersonation resistance. For each signal combination, the impersonation resistance was only about 5% greater than for the temporal variance measurement.

Randomness of Derived Keys

Table 14: Entropy of Keys

b	entropy of keys
2	0.9272

Temporal Variance and Distinctiveness of Derived Keys

Table 15: Temporal Variance and Distinctiveness of Keys

b	Temporal Variance	Distinctiveness
2	0.351	0.362

The average temporal variance and distinctiveness of the derived keys were about the same. While below the ideal 0.5, it is still much greater than what is seen with the keys generated by SKA-PS.

7.3 Results Summary

For both ECG-ECG and ECG-BP key generation, both the TAR and FAR generally fell as b increased. ECG-ECG key generation performed better than ECG-BP key generation in terms of correct key generation while ECG-BP key agreement produced less false key generations. The best key generation rate that we achieved was when b=2 for ECG-ECG key agreement which was 83.75%. However, this was accompanied by a FAR in terms of signal replay of 8.21%. SKA-PS achieved a TAR of 100% or very close to 100%, and a FAR of <1%, for the parameters the authors used in their evaluations.

We achieved good temporal variance for the generated keys and binary sequences. The temporal variance of the keys we generated using the chosen parameter sets for both ECG-ECG and ECG-BP was higher than the original SKA-PS. The average temporal variance of the keys produced for b=2,3,4,5 for ECG-ECG keys was between 0.356 and 0.397. The average temporal variance of the keys produced for b=2 for ECG-BP keys was 0.351. While the keys we generated using the hybrid approach were more temporally variant than for SKA-PS it fell short of the ideal 0.5. The temporal variance of the signals was only slightly higher than the temporal variance of the keys.

The distinctiveness of the generated keys was on average slightly higher than for the temporal variance, with an average ranging from 0.363 to 0.404 for ECG-ECG and an average of 0.362 for ECG-BP. SKA-PS achieved a distinctiveness

measurement for generated keys of around the ideal 0.5. The distinctiveness of the binary sequences was slightly higher than for the keys.

In terms of Shannon entropy for the binary sequences and keys, our hybrid approach measured well with the entropy for both the keys and binary sequences being above 0.9 and in many cases close to 1 which indicates perfect entropy. The average Shannon entropy of the keys for vanilla SKA-PS is less than 0.9 for all of the parameter sets that the authors report.

Our results show that this hybrid approach performs relatively well compared to vanilla SKA-PS in terms of key entropy, key temporal variance, and key distinctiveness. However, the hybrid approach falls short in TAR.

8 Conclusions

Secure key agreement is an important area of research in the context of Body Area Networks. Because they communicate private health information and can be used to control a person's vitals, they require a protocol that is lightweight and produces keys that are random, temporally variant, and distinct. We attempt to take a binary sequence generation method, HBBS, and use it with a key agreement protocol, SKA-PS. This combination would help address the weaknesses of both while retaining the strengths of both. SKA-PS allows for key agreement between two sensors that measure two different physiological signals but falls short in that it requires a lengthy signal collection time before the protocol can run. HBBS expands the bit representation of IPI values and transforms them into binary sequences that provide good randomness properties and requires a shorter signal collection time to generate the same binary sequence length. However, HBBS falls short in that the researchers only considered ECG signals and did not build a key agreement protocol on top of it.

In our project, we collect real subject data from Physiobank, process the data, generate binary sequences using HBBS, and then attempt to generate keys using SKA-PS. We vary parameters for both HBBS, such as FMIS length and the number of bits extracted, and SKA-PS, such as b, r, and s, to try to find a combination that allows for the best possible key agreement rate while also preventing false key agreements. We hoped a fusion of the two approaches would lead to a TAR and FAR of close to the original SKA-PS, with a TAR of close to 100% and a FAR of less than 1% for both ECG-ECG and ECG-BP key agreement. We also hoped for temporal variance and distinctiveness of close to 0.5 and an entropy of close to 1 for both the generated keys and the random binary sequences. However, that was not the case. With the parameters we tested, ECG-ECG key agreement outperforms ECG-BP key agreement by a wide margin. The best TAR for ECG-ECG key agreement we achieved was with $b = 2$ is 83% with an FAR of 9%, which is not as good as results achieved with the original SKA-PS. For ECG-BP key agreement, the best result we achieved was with $b = 2$ with a 44.8% TAR and an 11.8% FAR. The keys we generated were more temporally variant and had more entropy than vanilla SKA-PS but they were less distinct.

We learned that a technique such as HBBS can be used to efficiently generate binary sequences that have good randomness, temporal variance, and distinctiveness properties. We also discovered the difficulties associated with ECG and BP peak detection, and that using a good peak detection algorithm is essential for key agreement success. There were some subjects whose data we tried to use for our testing of the hybrid approach, but the signals were very noisy and so the peak detection was not as effective as it should have been. We did not consider the subjects where low peak detection resulted because of the noisiness and irregularities of the signal. If no or very little peaks are detected, then it becomes unfeasible to use a signal feature such as IPI as the basis for cryptographic keys.

We conclude that HBBS requires a more tailored approach for a key agree-

ment protocol. Perhaps some form of quantization can be applied to account for slight variations in IPIs calculated between signals. The authors of HBBS did not clearly state why they chose to encode each IPI into 23 bits. Future work should explore varying codeword and generator polynomial lengths to determine how they affect binary sequence properties and key agreement. As the HBBS authors alluded to in their work, another possible future direction is to explore other forms of encoding, such as convolutional or Hamming encoding.

In summary, while our research did not provide the desired outcome, it raises research areas that can be explored in future work.

Bibliography

[1] *Ecg of a heart in normal sinus rhythm*, https://en.wikipedia.org/wiki/Electrocardiography.

[2] 104th United States Congress, *Health insurance portability and accountability act*, https://www.hhs.gov/hipaa/index.html, 1996.

[3] Taddei A, Distante G, Emdin M, Pisani P, Moody GB, Zeelenberg C, and Marchesi C, *The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography*, European Heart Journal **13** (199), 1164–1172.

[4] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, and Stanley HE, *Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals*, Circulation 101(23):e215-e220 (2000).

[5] Duygu Karaoğlan Altop, Albert Levi, and Volkan Tuzcu, *Deriving cryptographic keys from physiological signals*, Pervasive and Mobile Computing **39** (2017), 65 – 79.

[6] Duygu Karaoğlan Altop, Beste Seymen, and Albert Levi, *Ska-ps: Secure key agreement protocol using physiological signals*, Ad Hoc Networks **83** (2019), 111 – 124.

[7] David Fass, *Entropy*, https://www.mathworks.com/matlabcentral/fileexchange/12857-entropy, 2016.

[8] Moody GB and Mark RG, *The impact of the mit-bih arrhythmia database*, IEEE Eng in Medicine and Biology **20** (2001), 45–50.

[9] Alexandre Laurin, *Bp_annotate*, https://www.mathworks.com/matlabcentral/fileexchange/60172-bp_annotate, 2017.

[10] Walter Edgardo Legnani, *Fractal analysis of cardiovascular signals empowering the bioengineering knowledge - scientific figure on researchgate*, https://www.researchgate.net/figure/Analysis-of-a-pulse-wave-for-an-arterial-pressure-signal-Diastolic-blood-pressure-DBP_fig2_318776373.

[11] M. Li, W. Lou, and K. Ren, *Data security and privacy in wireless body area networks*, IEEE Wireless Communications **17** (2010), no. 1, 51–58.

[12] Iyengar N, Peng C-K, Morin R, Goldberger AL, and Lipsitz LA, *Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics*, Am J Physiol **271** (1996), 1078–1084.

[13] Sandeep Pirbhulal, Heye Zhang, Wanqing Wu, S.C. Mukhopadhyay, and Yuan-Ting Zhang, *Heart-beats based biometric random binary sequences generation to secure wireless body sensor networks*, IEEE Transactions on Biomedical Engineering **PP** (2018), 1–1.

[14] Masoud Rostami, Ari Juels, and Farinaz Koushanfar, *Heart-to-heart (h2h): authentication for implanted medical devices*, Proceedings of the 2013 ACM SIGSAC conference on computer and communications security, CCS '13, ACM, 2013-11-04, pp. 1099,1112 (eng).

[15] Hooman Sedghamiz, *Complete pan tompkins implementation ecg qrs detector*, `https://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector`, 2018.

[16] R.M. Seepers, C. Strydis, P. Peris-Lopez, I. Sourdis, and C.I. De Zeeuw, *Peak misdetection in heart-beat-based security: Characterization and tolerance*, 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014, vol. 2014, Institute of Electrical and Electronics Engineers Inc., 2014-11-02, pp. 5401,5405.

[17] Robert Seepers, Jos Weber, Zekeriya Erkin, Ioannis Sourdis, and Christos Strydis, *Secure key-exchange protocol for implants using heartbeats*, Proceedings of the ACM International Conference on computing frontiers, CF '16, ACM, 2016, pp. 119,126 (eng).

[18] Beste Seymen, *On the establishment of pseudo random keys for body area network security using physiological signals*, Master's thesis, Sabancı University, Orta, Sabancı Ünv. No:27, 34956 Tuzla/İstanbul, Turkey, 2019.

[19] K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, *Ekg-based key agreement in body sensor networks*, Proc. 2nd Workshop Mission Critical Networks (2008), 1–6.

[20] K.K Venkatasubramanian, A Banerjee, and S.K.S Gupta, *Pska: Usable and secure key agreement scheme for body area networks*, IEEE Transactions on Information Technology in Biomedicine **14** (2010-01), no. 1, 60,68 (eng).

[21] G. Zheng, G. Fang, R. Shankaran, M. A. Orgun, J. Zhou, L. Qiao, and K. Saleem, *Multiple ecg fiducial points-based random binary sequence generation for securing wireless body area networks*, IEEE Journal of Biomedical and Health Informatics **21** (2017), no. 3, 655–663.