Real-Time Body Area Network Security

Student(s)

Baran Baysal
Alper Bingöl

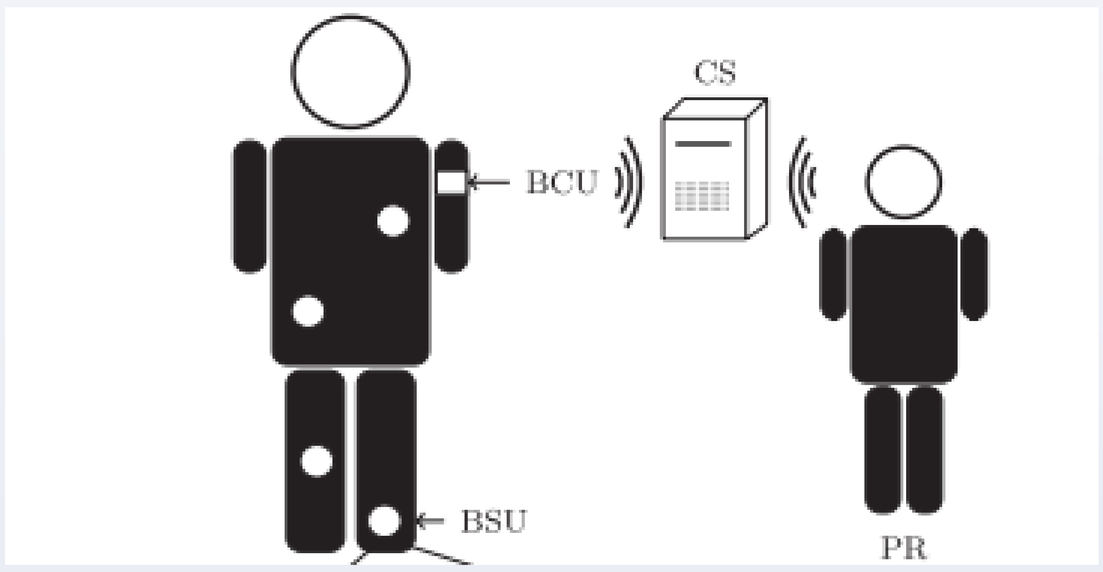
Faculty Member(s)

Albert Levi
Duygu Karaoğlan Altop
Beste Seymen

Sabancı Üniversitesi

PURE
PROGRAM FOR UNDERGRADUATE RESEARCH

ABSTRACT



This project focuses on the key distribution problem in Body Area Networks (BANs), which provide accurate monitoring of the human periphery through the use of biosensors. The functionalities of these biosensors are to effectively and efficiently collect data from vital body parts and share it with an aggregating device that transfers the physiological signals of the corresponding user to a central server. The captured phenomena are highly sensitive to contraventions against privacy. Moreover, they are exchanged among the BAN elements using wireless communication. Therefore, it is essential to build a security mechanism for the protection of the phenomena gathered from the human body, as well as providing secure node-to-host association.

On one hand, we have electrocardiogram (ECG), blood pressure (BP), surface temperature (ST) and hand grip heart rate (HR) sensors, together with a data acquisition unit, using which the collected data can be transferred to a computer. On the other hand, we have a Raspberry Pi (single-board computer), using which a Python-written secure key agreement protocol can be run with the input being physiological signals of ECG and BP. These two modules are required to be integrated with each other. This way, a real-time working security infrastructure can be constructed for BAN security.

OBJECTIVES

- Avoiding potential interference attacks among different BANs and providing secure node-to-host association naturally.
- To generate IPI's while we are gathering data from the host.
- Program a Raspberry pi to do this.

PROJECT DETAILS

```python
import os
import socket
import sys
from os.path import join
import json
import time

def prepare_data(message):
    return(json.dumps(message))

#socket initialization
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('192.168.43.213', 1235))
s.listen(10)

while True:
    connect, addr = s.accept()
    message = 'Hello this is server'
    data_string = prepare_data(message)
    connect.send(message.encode('utf-8'))

    #time.sleep(2)
    incoming = connect.recv(1024)
    print(incoming)

    #connect.close()
    #s.close()
```

```python
import socket, coding, os, json, time, itertools, sys
from os.path import join

#socket initialization
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("193.255.135.254", 1234))

data = s.recv(1024)
incoming = json.loads(data)

print(incoming)

s.send('Hello server')

#s.close();
```

We are using Rasberry Pi to calculate the IPI's from the data we collect from the users. We parsed the text file in such a way that gave us the 'x' and 'y' values we need to calculate the IPI.

We wrote two pieces of code (sockets) to transmit that data from the computer to the Raspberry Pi. The server code parses the data text file and sends it to the socket to be used by the client (Raspberry Pi). The client code that's being run in the Raspberry Pi which converts those parsed data to person specific IPI's.

PROJECT DETAILS II

Algorithm 1 Proposed IPI-based Physiological Parameter Generation Technique

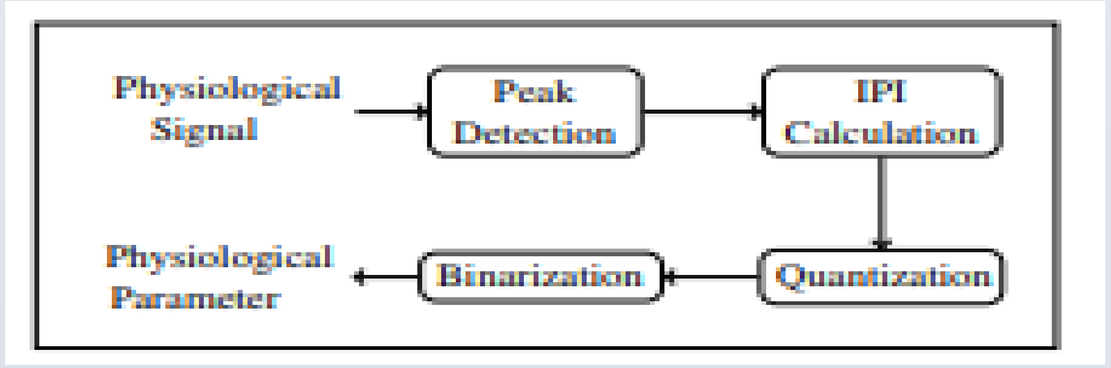
INPUT: $Signal$, l, g, min, max, s, n
OUTPUT: $PhysParam$

1: $P = FindPeakLocations(Signal)$
2: **for all** $i \in \{1, ..., l\}$ **do**
3: $\quad IPI_i^{init} = P_{i+1} - P_i$
4: **end for**
5: $IPI = zeros\ (l/g)$
6: $k = 1$
7: **for** $i = 1 : g : l$ **do**
8: $\quad$ **for all** $j \in \{1, ..., g\}$ **do**
9: $\quad\quad IPI(k) = IPI(k) + IPI^{init}(i + j - 1)$
10: $\quad$ **end for**
11: $\quad k = k + 1$
12: **end for**
13: $len_{part} = floor\ (max - min)/s$
14: $part = zeros\ (len_{part})$
15: $code = zeros\ (len_{part} + 1)$
16: **for all** $i \in \{1, ..., len_{part}\}$ **do**
17: $\quad part(i) = min + i * s$
18: $\quad code(i) = i \bmod 2^n$
19: **end for**
20: $IPI^{quant} = Quantization\ (IPI, part, code)$
21: $PhysParam = GrayEncoding\ (IPI^{quant})$

Symbol		*Description*
$IPI_{(a,b),i}^{(c,d),j}$	i	IPI index
	a, b	User index
	j	Signal type $\in \{ECG, PPG, BP\}$
	c, d	Start time index of IPI
l		Length of the initial IPI sequence
g		Size of the IPI groups
s		Step size for quantization
min, max		Minimum and maximum values of an IPI
n		Bit length of a quantized and binarized IPI
$D_{(d,s,t)}$		Distance between physiological parameters
	d	of different hosts
	s	of the same host at same time
	t	of the same host at different times
FAR		False Accept Rate
FRR		False Reject Rate
HTE		Half Total Error Rate
FAR_{HTE}, FRR_{HTE}		FAR and FRR at HTE
t		Expected value of the elapsed time to generate two matching physiological parameters

We get data from the BSU(Body Sensory Unit) and transmit that data to the CS(Central Server) in real-time using person specific cryptograpy. We get IPI's from the signals that body produces.

These IPIs are created via the signals using the algorithm above. This can be applied on the ECG, PPG AND BP signals in order to derive IPI-based physiological parameters that can be used as cryptographic keys. First of all peak points of signals are determined. Then, IPI sequences are generated by computing the time elapsed between the adjacent peak points. Each IPI sequence is divided into group in order to decrease the effect of measurement errors. Thereafter, these IPI sequences are quantized in order to further decrease the measurement errors. Finally, Gray encoding is applied on the resulting quantized IPI sequences in order to increase the error margin of the physiological parameters generated in different BANs.



CONCLUSIONS

We used real-time body area network securities to obtain IPI's with the help of a computer in between as a CS (Central Server). The BSU's (Body Sensory Unit) gather the data from the user and sends it to the CS, a computer, to parse the file and send the usefull data (our 'x' and 'y' values) to a socket. The client gets those values from the socket to get the peak values and calculate the IPI's.

We did not want to use a computer between the Raspberry Pi and the BSU's (Body Sensory Units) but could not figure out a way to do it. In the end we gather data in text files and send them to the Raspberry Pi. Raspberry Pi run the codes with functions while getting inputs of the text files we have sent. The cryptographic keys will be generated after the whole process.

BANs are the most important building stone of pervasive healthcare, enabling continuous, remote and real-time patient monitoring through the use of biosensors. These small wearable sensing devices are limited in energy and storage, and they collect very important and sensitive personal information. Therefore, light-weight security solutions are required for BANs in order both to preserve the privacy of the user and to provide the security of the exchanged data. The general work of the project aimed to provide that security.

REFERENCES

Karaoglan, D. and Levi, A. (2013). A Survey on the Development of Security Mechanisms for Body Area Networks. The Computer Journal, 57(10), pp.1484-1512.

Karaoğlan Altop, D., Levi, A. and Tuzcu, V. (2017). Deriving cryptographic keys from physiological signals. Pervasive and Mobile Computing, 39, pp.65-79.