BioSec: A Biometric Authentication Framework for Secure and Private Communication among Edge Devices in IoT and Industry 4.0

Muhammed Golec[1], Sukhpal Singh Gill[1], Rami Bahsoon[2] and Omer Rana[3]
[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]School of Computer Science, University of Birmingham, Birmingham, UK
[3]School of Computer Science and Informatics, Cardiff University, Cardiff, UK

***Abstract*— To address the privacy and security related challenges in Internet of Things (IoT) environment, we proposed a biometric method called BioSec to provide authentication in IoT integrated with edge consumer electronics with fingerprint authentication. Further, we ensured the security of biometric data both in the transmission channel and database with the standard encryption method. In this way, BioSec ensures secure and private communication among edge devices in IoT and Industry 4.0. Finally, we have compared three encryption methods used to protect biometric templates in terms of processing times and identified that AES-128-bit key encryption method outperforms others.**

INTRODUCTION

Internet of Things (IoT) technology has a wide range of uses, from consumer applications such as smart homes to infrastructure applications such as energy management[1]. With the growing use of IoT devices, security has become a significant issue in IoT environments. Security in IoT can be examined under the following three main headings: ensuring the security of the collected data, using an encrypted communication channel and using user authentication[2]. User authentication is essential to protect the privacy of personal data. Traditionally, user authentication in IoT devices was undertaken with pin-based password systems[1]. However, biometric systems have started to be used due to the weaknesses of the pin-based password, such as being forgotten, stolen and shared. Because biometric descriptors are inherent in an individual, it is more difficult to manipulate, share or forget these characteristics[3]. At the same time, Biometric data cannot be changed even if it is stolen because it has unique characteristics of the person, and can lead to privacy problems in case of leakage[4]. Therefore, we can use encryption methods to secure biometric data.

Motivation and Our Contributions

The aim of this study is to provide security in IoT systems with fingerprint authentication to prevent unauthorized access to the system and ensure the privacy of the IoT system. If a person's fingerprint is stolen, the security of IoT and the privacy of personal data in IoT will also be compromised. Since fingerprints are also unique, they bring a second privacy concern with them in case they are stolen. Therefore, we propose ***BioSec***, a framework for private use of Edge Consumer Electronics (ECE) and secured communication among IoT devices. In our study, we recommend protecting the security of IoT with fingerprint authentication due to its cheap sensor cost and high recognition performance. In addition, we use one of the Standard Encryption methods to secure biometric data both in the transmission channel and in the database. Standard encryption methods do not negatively affect the overall performance, such as feature transformation (stenographical encryption) and biometric cryptosystem (lightweight cryptography) methods used in literature.

BACKGROUND

A literature review shows a limited number of studies on biometric authentication in IoT systems. Wencheng et al. proposed a system that ensures the security of IoT with fingerprint authentication[2]. Although fingerprints are secured using a biometric cryptosystem (lightweight cryptography) in the database, they are vulnerable to transmission channel attacks. Nemanja et al. proposed a system in which two different biometric methods are used for authentication[5]. In this study, biometric security was not taken into account, although the biometric recognition performance was high and the sensor cost was low. Dirk et al. proposed a multimodal biometric

authentication that combines Hand Geometry and Gesture via a hardware device "Leap Motion Controller" to provide authentication[6]. In this study, biometric security is protected by stenographical encryption, but it is not very practical to use it because the system is still a prototype and the cost of the sensor.

BIOSEC SYSTEM

The BioSec System Design

We designed the BioSec system to ensure security and privacy with biometric authentication, which consists of two layers: client and server. Raspberry Pi-4 (RPi), as an IoT device is used as a client. The server part of the system is installed on a server. In the BioSec system, first the fingerprints of the user(s) to be registered into the system are taken with the help of a sensor, and these fingerprint images are sent to the RPi device. After fingerprint images have been processed at the RPi, a biometric template consisting of key (minutiae) points to be used in fingerprint authentication are obtained. Minutiae points are friction ridge skin impressions believed to be unique to each fingerprint. The biometric template is sent in encrypted form as a precaution against hacking attacks on the communication channel between the server and the RPi. The encrypted biometric templates are stored in an encrypted form into the database as a precaution against biometric template leakage attacks. In this way, security is provided against any database hacking. Figure 1 shows the path followed while creating the BioSec system database. Using a sensor, a user sends his fingerprints to the RPi, and the biometric template is extracted after the necessary image processing operations. The encrypted biometric templates previously saved in the database with the fingerprint template coming from the user are decrypted and on the server one by one. If a match is found, the user is permitted to access the system. If this fingerprint does not match the registered fingerprints, the user is denied access. Figure 2 presents the working scheme of BioSec.

RPi + Buttons: A mechanism has been established that sends fingerprint images taken from the fingerprint sensor through buttons. *Button 1* sends the fingerprint of a user

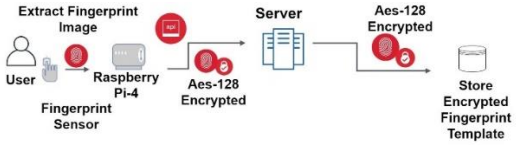

Fig. 1. Fingerprint Enrollment Stage

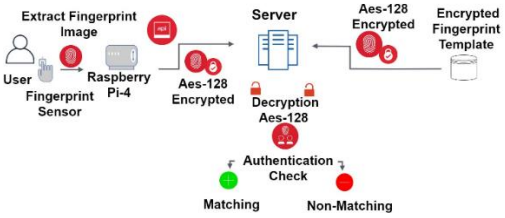

Fig. 2. Operation scheme of the BioSec system

registered in our system, while *Button 2* sends the fingerprint of a user who is not registered in our system. The flashing of the LED light differs according to the match status of the sent fingerprints: (i) the LED light turns on intermittently three times in a row if a match is found; (ii) the LED light turns on once if there is a mismatch. Moreover, in all cases, the system returns matching, mismatch or no match on the command screen. Figure 3 shows the client part of our system.

Configuration Settings: The server part of the system uses 12GB RAM and Intel (R) Core (TM) i5-7300HQ 2.5GHz clock frequency CPU. The operating system is Windows 10 Pro-64 Bit.

API and Database: In our system, an Application Programming Interface (API) was written using the Python language and flask framework. After sending a fingerprint template from RPi to the server in our system, an API call is used to reach the match results on the server. FVC-2004 DB3 fingerprint database was used, which is available for use on the Internet[8]. The first fingerprint images, based on 8 images for each user, are used for testing. The remaining seven fingerprints were stored in the database in an encrypted form after image enhancement. We use an open-source fingerprint matching algorithm[9].

The BioSec System Implementation

Fingerprint Enrollment Stage: First of all, fingerprint images of the users who wants to be registered in the system are passed through the necessary pre-processing, and biometric templates are created with feature extraction. In our system, biometric templates are shown with the Des variable and subsequently recorded in the database in an encrypted form.

AES-128 Bit method has been chosen as the encryption method. FVC2004 DB-3 database contains 80 fingerprint images of 10 users. Each user has eight different fingerprint images. From these eight fingerprint images, the first images are reserved for later use for system performance and match testing. As seen in Figure 4, *Des* refers to biometric templates containing minutiae extracted from fingerprint images. Des data containing minutiae points extracted from fingerprint images should not be confused with DES used in the encryption method. The result is returned according to the threshold value.

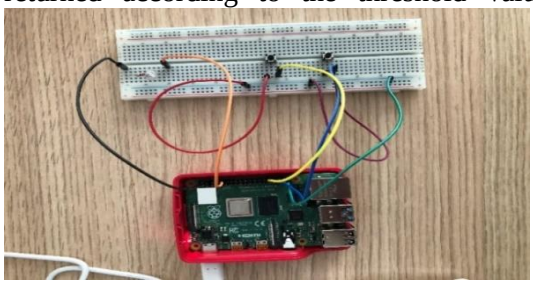

Fig. 3. In the client section, buttons that send Raspberry Pi and fingerprint images are seen.

With the des_from_csv() method, all encrypted Des data in the fingerprint database that we created earlier are returned as a dictionary. Figure 5 shows the utils.py file. There are four functions here. These include removedot(), which reduces noise by improving the given image, get_descriptors() which returns minutiae points of the given image and calculate() function which measures the similarity of the two minutiae according to the threshold value.

Client Part of the BioSec System: In the client part of our system, the fingerprint images obtained from the fingerprint sensor are sent to the RPi, where Des data containing fingerprint minutiae points are obtained with the removedot() and get_descriptors() methods – this data is then sent to the server in an AES-128 bit encrypted form,

Pseudo Code for Creating Fingerprint Database

```
Start
fvc2004_db3 = the path of fingerprint images
insert_data_into_db(fvc2004_db3)
def insert_data_into_db( PATH) :
   names, des_list, times = list() # Create 3 empty lists
   for i in PATH:
      if i != first picture:
         des = get_descriptors(i) # Extract minutiae points of i and assign
         names.append(i) # Add the image's name to the names list
         Time_start = time.time() # Turn on timer for the duration of the encryption process
         Encrypted_des = Encrypt(des) # Encrypt des
         Time_end = time.time()
         time_result = 1000 * (Time_end - Time_start) # To calculate as second
         times.append(time_result) # Append encryption time to the times list
         des_list.append(Encrypted_des) # Add encrypted des to des_list
   else: # Creates a test folder to test system accuracy
      test_files = f'{db_name}_test'   # Create a folder named test_files and copy the i image there
   # end for loop
 df = pd.DataFrame({"Name": names, "Des": des_list, "Encrypt_Time": times}) # Save names, des_list and times information as an excel file
End
```

Fig. 4. Creating the system fingerprint database

Pseudo Code for Utils.py

```
Start
def removedot (invertThin) :
   #Define a function to make image enhancement for the image given as argument and return the enhanced image as a return.
def get_descriptors (img) :
   # Define a function to send the image which is given as an argument to the function defined above for image enhancement. Extract minutiae points from the enhanced image and return.
des_from_csv () :
   # Define a function to return fingerprint minutiae points in the created database as dictionary forma
def calculate (one, two) :
   # Define a function to compare the two minutiae points it received and return True or False according to the given threshold value.
End
```

Fig. 5. Methods in Utils.py file

With the Flask API, the matching results on the server part are returned to the RPi through the communication channel. Button1 sends the fingerprint image not found in the database of a user previously registered to the system database, to the RPi. Button2 sends the fingerprint image of a user who is not in the system to the RPi. Since the image sent from Button1 matches a user registered in the system and the correct person, the LED in our system blinks three times in a row. Figure 6 shows the correct match status. Since the fingerprint image we sent from Button2 is not registered in the system, the LED blinks only once. Figure 7 shows the mismatch status. Also, if a registered fingerprint image is sent from Button2 to the system and this fingerprint image matches another registered person, in case of false acceptance, the LED blinks only one time. Figure 8 shows the false acceptance situation.

RESULT ANALYSIS

Performance of Fingerprint Authentication System

The accuracy of the system is calculated by drawing a FAR - FRR graph. Along with the changing threshold value, FAR and FRR

amounts also change based on a trade-off with each other.

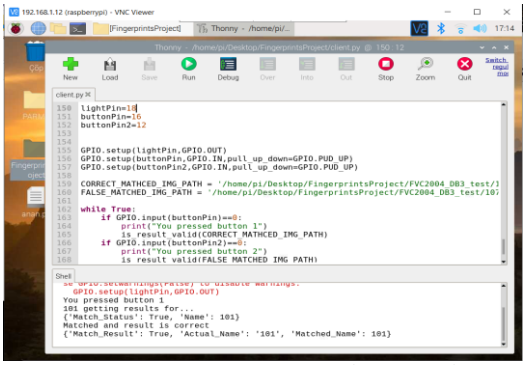

Fig. 6. Raspberry Pi shows the state of accurate fingerprint matching

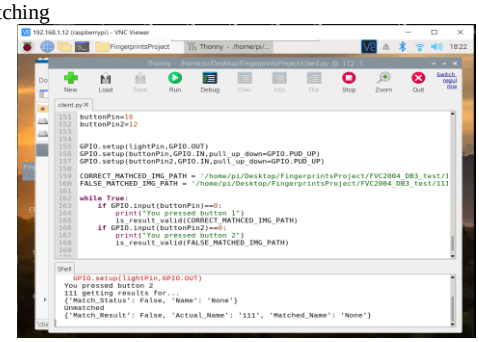

Fig. 7. The case of sending a fingerprint that does not exist in the database from Button 2

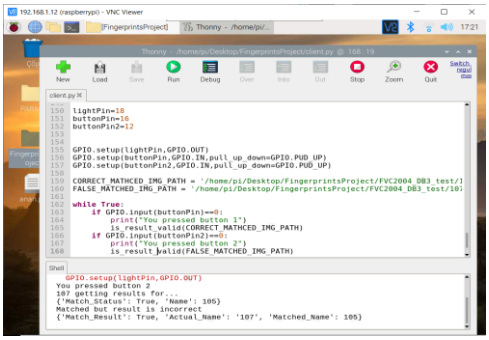

Fig. 8. False acceptance status with one of the fingerprints in the database

Figure 9 shows the FAR - FRR graph of our Fingerprint biometric authentication system that we use in our system. Our EER rate was found to be 30%. When our threshold value was 34, our system gave an optimum performance. The overall performance of our system demonstrates 70% accuracy.

Performance Evaluation of Encryption Algorithms

In the BioSec system, biometric data must be encrypted both in the transmission channel and in the database to ensure the security of biometric data.

By testing three different methods, encryption-decryption performance times were calculated. Table 1 shows the execution times of the three encryption methods with different key lengths.

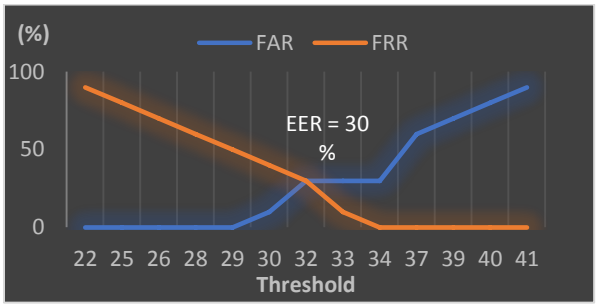

Fig. 9. FAR-FRR graph of the fingerprint authentication algorithm used in the BioSec system

Table 1 Transaction Times of Standard Encryption Methods According to Key Lengths

Encryption Method	Key length (bit)	Encryption Time(s)	Decryption Time(s)
AES	128	1,016209	0,695434
AES	192	1,018884	0,757293
AES	256	1,033279	0,698176
DES	64	2,019291	1,740159
3DES	128	25,57337	26,09123
3DES	192	25,58849	26,08399

It was seen that the algorithm with the best performance was AES-128 bit with an encryption time of 1.02s and a decryption time of 0.695s.

THE EFFECT OF BIOSEC ON CONSUMER EXPERIENCE

In our study, we designed a system with biometric authentication to protect user privacy in IoT devices. Based on this system, users who want to log into an IoT device will be able to enter the system with fingerprint authentication, thus ensuring the security and privacy of IoT. Biometric data used to provide security in IoT devices raises a second privacy concern in case of stolen data, because such information can be used to identify individuals. We proposed the BioSec system to ensure the privacy of biometric data of users. The availability of the system will increase using biometric authentication because users will not have to carry a token with them or require a password/pin they have to remember.

CONCLUSIONS AND FUTURE WORK

In this article, a BioSec framework is proposed for biometric authentication for secure and private communication between edge devices in IoT and Industry 4.0. The security level of IoT devices has been further increased by protecting biometric data used with encryption methods. Biometric data sent and stored in data transmission channels and the database of the system are secured by using encryption. In our system, processing times are compared using three different encryption methods, and the fastest algorithm is used for encryption. The system can be made much more secure by improving the BioSec privacy mechanisms and including other biometric markers. Since symmetric encryption methods are used in the system, the security of biometric data is also compromised in case the key used in encryption is stolen.

The utility of the system can be improved for both security and processing speed by solving the key distribution problem or finding lightweight encryption algorithms that can be used on edge devices, or by using biometric methods with high-performance such as iris.

ACKNOWLEDGEMENTS

Muhammed Golec would express his thanks to the Ministry of Education of the Turkish Republic, for their support and funding.

REFERENCES

[1] S. S. Gill and A. Shaghaghi. Security-Aware Autonomic Allocation of Cloud Resources: A Model, Research Trends, and Future Directions. Journal of Organizational and End User Computing, 32(3), pp.15-22, 2020.

[2] W. Yang et al., "A privacy-preserving lightweight biometric system for internet of things security," IEEE Commun. Mag., vol. 57, no. 3, pp. 84–89, 2019.

[3] S.F. Aghili et al, LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. Future Generation Computer Systems, 96, pp.410-424, 2019

[4] M. Ghahramani et al., RSS: An energy-efficient approach for securing IoT service protocols against the DoS attack. IEEE Internet of Things Journal, 2020.

[5] N. Maček, I. Franc, M. Bogdanoski, and A. Mirković, "Multimodal Biometric Authentication in IoT: Single Camera Case Study," 2016.

[6] L.-P. Shahim, D. Snyman, T. Toit, and H. A. Kruger, "Cost-Effective Biometric Authentication using Leap Motion and IoT Devices," 2016.

[7] S. S. Gill and R. Buyya. SECURE: Self-protection approach in cloud resource management. IEEE Cloud Computing, 5(1), 60-72, 2018

[8] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2004: Third fingerprint verification competition," in Biometric Authentication, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–7, 2004.

[9] K. Jankoski, python-fingerprint-recognition. https://github.com/kjanko/python-fingerprint-recognition

Muhammed Golec is a MSc student with Queen Mary University of London, UK. Contact him at m.golec@hss18.qmul.ac.uk

Sukhpal Singh Gill is a Lecturer with Queen Mary University of London, UK. Contact him at s.s.gill@qmul.ac.uk

Rami Bahsoon is a Senior Lecturer with University of Birmingham, Birmingham, UK. Contact him at r.bahsoon@cs.bham.ac.uk

Omer Rana is a Professor with Cardiff University, Cardiff, UK. Contact him at ranaof@cardiff.ac.uk