ICIC Express Letters
Volume **15**, Number **12**, December **2021**

ICIC International ©2021 ISSN 1881-803X
pp. **1253–1259**

SECURING APPLE WATCH ECG DATA TRANSMISSION USING ENHANCED RANDOMIZED RAIL FENCING CHIPER

Ari Wijaya Limanto and Sani Muhamad Isa

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ ari.limanto; sani.m.isa }@binus.ac.id

Received February 2021; accepted May 2021

Abstract. *IoT devices, Apple and FitBit smartwatch for example, in the health sector show enhancements and are very promising in recent times because both can record ECG results well and could be used as a basis for differentiating AF from Sinus Rhythm. The recorded data can be assumed as a medical record, and it needs to be securely stored and transferred including external transmission. Attacks that potentially acquire user's personal data, especially health data or medical records are the challenge and will be discussed in the context of this study. The proposed solution uses a transposition cryptosystem, Rail Fencing Cipher, which is a lightweight cryptographic algorithm and one well-studied cipher technique to overcome the challenge. OWASP ZAP shows the data securely transferred without any medium or above alert category in terms of evaluation. Implementation of any machine learning tasks can be a future research direction.*
Keywords: Internet of Things, Cloud platform, Transposition encryption, Man-in-the-Middle attack, Medical record

1. **Introduction.** IoT (Internet of Things) currently is recognized as a concept that has substantial impacts on all aspects of our daily life [1]. IoT devices have light computing power with the ability to process and transmit data and are equipped with at least one type of sensor. The active device itself over the globe is reaching more than 50 billion in 2019 [2]. Utilizing IoT devices in the health sector is very promising [3], but in fact it lacks on the application. Based on statistics from the World Health Organization (WHO), the annual death rate due to heart failure reaches 17.9 million people.

In helping detect cases of heart failure, Apple Watch can be said to have a better sensitivity in collecting ECG data than other devices. One brand that uses a wearable sensor that is getting quite a positive response from the market is FitBit. [4] was checking for Atrial Fibrillation (AF) and Atrial Flutter symptoms using FitBit Blaze device and Apple Watch based on ECG signal recorded. [4,5] show that the AF detection algorithm of the Apple Watch, supported by a doctor's review of the data used in the study, can be used as a basis for differentiating AF from Sinus Rhythm (SR). In addition, Apple Watch has received approval from the U.S Food and Drug Administration (FDA) [6,7].

Although the Apple Watch performs well in recording ECG data, transmitting ECG data from iOs (iPhone/iPad) to data centers or cloud services, or in other words via the public Internet, creates vulnerabilities. The most common potential attacks are Eavesdropping, Sybil attack, Sinkhole attack, Sleep Deprivation attack, and Man-in-the-Middle (MITM) attack [8]. These attacks potentially acquire user's personal data, especially health data or medical records in the context of this study. Apart from security challenges, the public Internet also faces several other challenges including the quick response

DOI: 10.24507/icicel.15.12.1253

requirement, handling multiple users at the same time, along with flexibility in operation [9].

We propose a method on transmitting ECG data using transposition encryption to reduce resource utilization which is secure and lightweight and unique device ID to increase the security. Over power consumption, high bandwidth required, or limited computation resources are common dealt cases when performing advanced security on a data. We use Apple Smartwatch in this research as ECG data source and iPhone as iOs device where the application will run due to ECG recording accuracy and the Smartwatch has been approved by FDA. The proposed method will implement Secure Socket Layer (SSL) as a baseline.

This paper is organized as follows. Section 2 contains related works. Proposed method is explained in Section 3. Section 4 focuses on Man-in-the-Middle attack analysis. Section 5 shows results and discussion. The last part, Section 6, ends this paper with a conclusion as well as future work.

2. **Related Works.**

Few researches were highlighted in this paper in order to provide insights of the solution that have been worked on. These researches show crucial aspects of data security, starting from the encryption model, transmission flow, to optimal end-to-end infrastructure.

[10] focuses on cryptographic methods in monitoring ECG using Fully Homomorphic Encryption (FHE). This method is known to perform quite heavy computations because of the analysis process during the encryption process. The framework in this research provides an optimized FHE encryption method, while adhering to the HIPAA regulations in protecting patient data, but can significantly reduce the computational load. To be able to transmit data from IoT devices to cloud services in real time, the proposed solution requires a minimum bandwidth of 2 Mbps.

In line with the topics discussed in this study, [11] offers a solution with Black Networks, which are networks that secure all data, including metadata, associated with each frame or packet in the IoT protocol. One thing to be highlighted is the recommendation of a separate key management system to generate, distribute, store, revoke, change, and use keys. The shared key referred to in this study is a symmetric shared key. In practice, most of the shared keys are carried out without using a key management system. It means that there is no key rotation in the future.

In regards to mobile application, the security aspect is concerning. A framework is introduced called iSec [12-20], introducing three pillars which are storage, access, and data transfer development strategy. Universal Device Identifiers (UDIDs) are utilized in this research and it seems to be the most reliable way which can be used to identify the requests for the data sent to the server. XMLParser in this research which will enable to read and write the security configuration stored in an external XML file could potentially open vulnerabilities to attackers though.

A proposed authentication protocol using Elliptic Curve Cryptography (ECC) based on three factors password, smart card and biometric is introduced in 2020 [13]. The evaluation results have also been tested for the formal security verification using the widely-accepted AVISPA tool. The result proves that the protocol is secure against known attacks including the replay and Man-in-the-Middle attacks. Unfortunately, the evaluation does not measure the total time taken for the authentication process.

The proposed algorithm, Rail Fence Cipher (RFC), is efficient (in terms of complexity), lightweight and energy efficient [14] and can be used for encryption in mobile devices. By increasing the complexity of determining the pattern used for the transposition, the method expects to remove security vulnerabilities of RFC. Attacks analysis in this research is limited to Man-in-the-Middle attack due to HIPAA compliance regarding patient medical records.

3. **Proposed Method.** This section will cover how the proposed solution would be delivered. In general, the proposed solution uses a transposition cryptosystem mostly used for mobile chatting. The transposition cipher is a lightweight cryptographic algorithm and one well-studied cipher technique. In transposition cipher, the Ciphertext (CT) is obtained from characters of the Plaintext (PT) permutations. The character set in Figure 1, both in PT and CT is the same, but the positions of the characters are shifted in CT. The main goal of the transposition cipher is to create diffusion. It is widely used in cryptographic algorithms, including Rail Fence Cipher [15,16], double transposition [17], Myszkowski transposition [18]. The word CSPRN is an abbreviation for Cryptographically Secure Pseudo-Random Number, which is used to retrieve a session key for encrypting the text messages. The proposed solution data flow diagram is shown in Figure 2. User

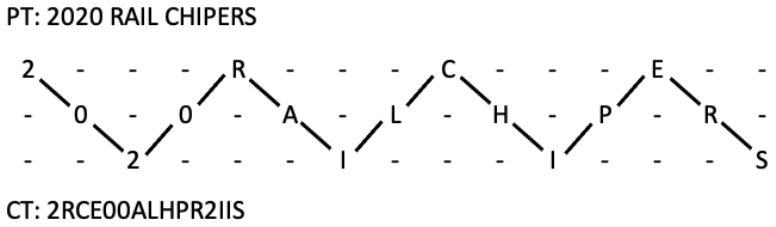

FIGURE 1. Basic architecture of the proposed protocol

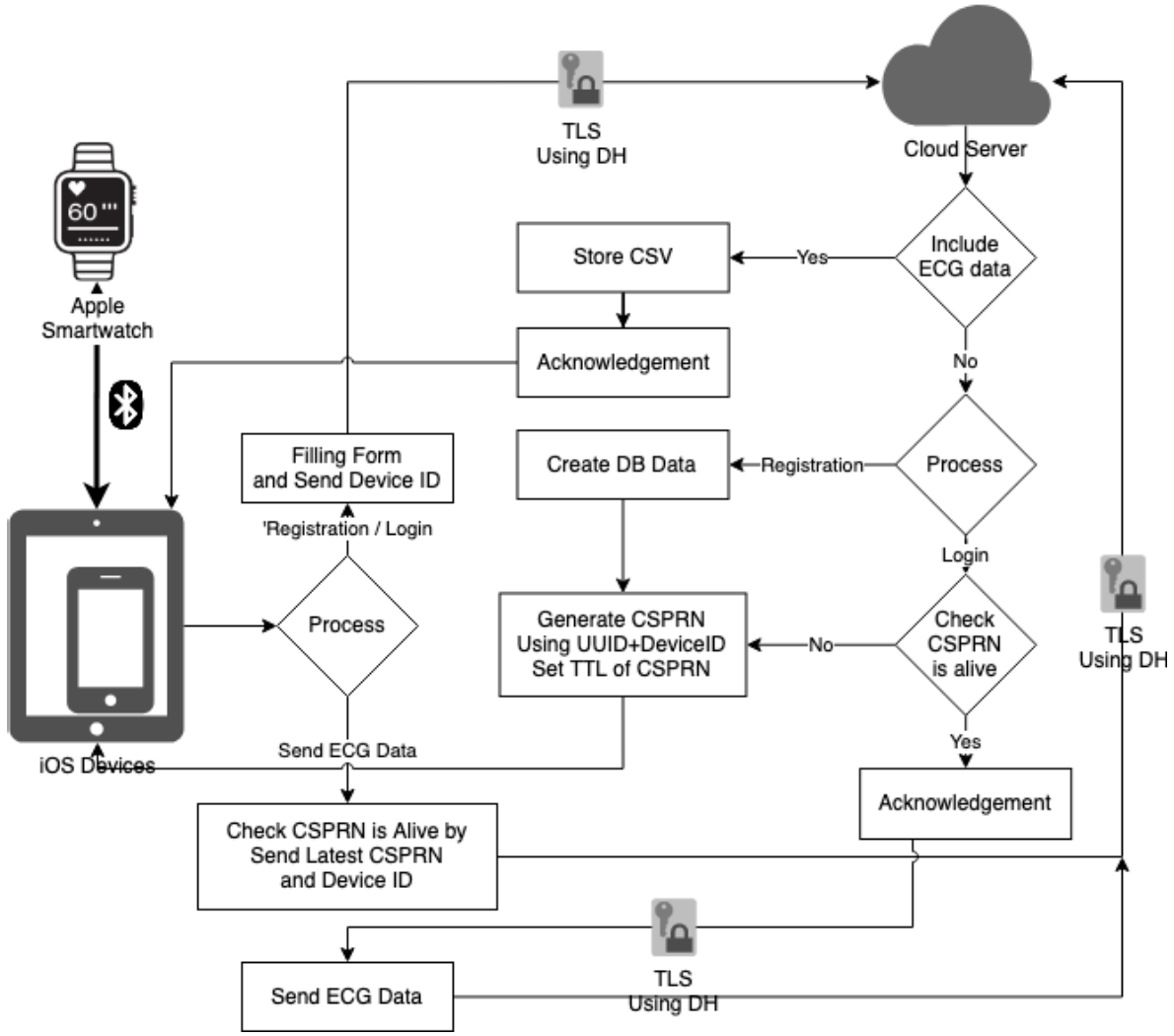

FIGURE 2. Data flow diagram of the proposed solution

needs to log in or register from the iOs device to verify or acquire valid CSPRN from the server. Once the CSPRN is stored in both server and device, iOs application will run in the background, listening periodically for new ECG data. CSPRN will be generated overtime due to having Time to Live (TTL) on the server. If the CSPRN is already expired, iOs device will renew the CSPRN in the beginning of sending ECG data. Data will be sent over secure communication channel (HTTPS) in addition with transformed ECG records that will be explained in more detail in Section 3.3. iOs device stores ECG logs which are already sent to prevent redundancy.

3.1. **System architecture.** The proposed solution in this paper uses technology stacks, such as web server in Go Language, MySQL database management system (DBMS) which will be used to store patient data in the server, in-memory database which will be used to store CSPRN with TTL in the server, and iOs application in Swift to listen and send ECG data to the server. In [18] the CSPRN can be generated in the mobile device, but in terms of security and authenticity, this paper suggests that the CSPRN is generated in the external server/web server.

3.2. **Obtaining CSPRN.** CSPRN will be stored in two places, first is in the in-memory database in the server, second is in the local mobile storage inside the iOs application. To obtain CSPRN from the mobile apps, firstly user needs to either register as new user or log in as registered user. User UUID and device ID are required to generate CSPRN. Device ID in the iOs can be obtained by accessing variable *UIDevice.current.identifier ForVendor?.uuidString*. These two values would be stored in in-memory database as the key to validate CSPRN later on. CSPRN in this paper is using 512-bits to extend the security [19]. Process "Generate CSPRN Using UUID+Device ID and Set TTL of CSPRN" especially how to generate the CSPRN in Figure 2 is more elaborated in Figure 3. Everytime user is logged in, the device ID is stored to be compared when the data is received. Variable totalEmptyArray determines how many prefixes before putting first plain text in encryption process.

```
func GenerateCSPRN (sharedKey string) (string, int) {
    CSPRN := ""                                          // Empty CSPRN
    size := 512
    seed := mathrand.Intn(100) + 100                     // Seed or Key
    sequence := mathrand.Intn(100) + 100                 // Sequence number
    totalEmptyArray := (mathrand.Intn(rangeInitialEmpty) + 16)
                        * aes.BlockSize                  // Total whitespace before content
    loop := size / 128                                   // Loop would be 4 times
    for i := 0; i < loop; i++ {
        CSPRN = CSPRN + AES(seed, sequence, sharedKey)   // Concatenation of result
        sequence++
    }

    return CSPRN, totalEmptyArray
}
```

FIGURE 3. Generating CSPRN

3.3. **Randomized Rail Fence Cipher (RRFC).** This section will explain more detail on RRFC. This algorithm randomizes the length (l_i) and the starting positions (S_i) of each Downward Diagonal Fence (DDF). Upward Diagonal Fences (UDF) on the other hand can be obtained by joining endpoint and starting points (S_i) of two consecutive DDFs. Permutation of the rails is the key to generating the ciphertext.

A randomized fence can be implemented by providing two parameters, the initial position (S_i) (the rail from where it starts) and the length (l_i). In Figure 4 for instance, the first DDF can be characterized as $(2, 4)$. Similarly, the next DDF can be represented by $(3, 3)$. Using this notion, we can enlist all DDFs of Figure 4 as $<(2, 4)(3, 3)(1, 5)>$. The same representation works equally for determining the UDF.

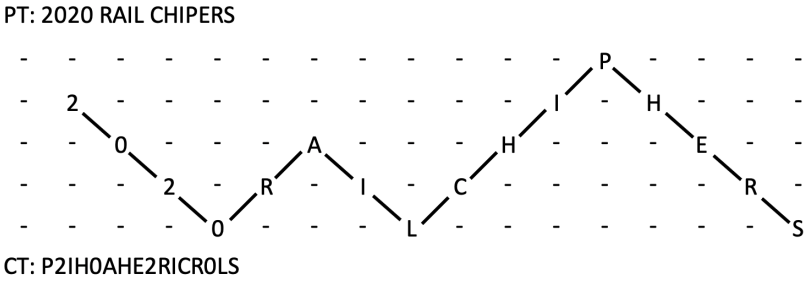

FIGURE 4. RRFC example

4. **Man-in-the-Middle Attack Analysis.** MITM can be categorized into several types [20,21], such as *Spoofing based* attack which the attacker intercepts the legitimate with the aid of spoofing and controls the data in and out without hosts being aware of the attacker existence, *TSL/SSL MITM attack* which the attacker bridges two separate SSL connection and relays traffics or messages between the communication of endpoints or targets, *BGP based MITM attack* which the attacker transmits the stolen traffic to the target or known as IP hijacking as well where there are possibilities of traffic manipulation, and lastly *false base station based MITM attack* which the attacker creates a fake transceiver station and then manipulate the target traffics. Most of the MITM attacks involve traffic through a middleman but a novel MITM scheme is not available though. MITM attacks can be prevented by implementing several cryptographic techniques such as elliptic curve cryptography [13], secure key distribution [22] and authentication methods.

5. **Results and Discussion.** In this research, we use CSPRN which is being requested from mobile devices as identification. User ID, Device ID and previous CSPRN should be sent to validate the legitimate user. Sending out an unregistered Device ID will also reject an unusual attempt. We use a proxy server, OWASP Zed Attack Proxy (ZAP), to be able to listen to any packages going in and out to assess the security level of the data. OWASP can help you automatically find security vulnerabilities and has the ability to simulate active or passive attacks. It is also a well-known tool for experienced pentesters for manual security testing.

We run standard mode in ZAP to perform security vulnerabilities to our server running in Go. Figure 5 shows that there is no high security breach, especially in sensitive

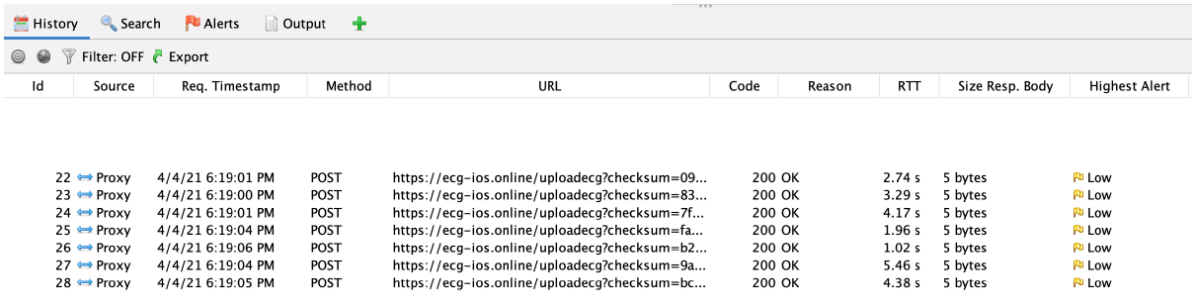

FIGURE 5. ZAP scan alerts

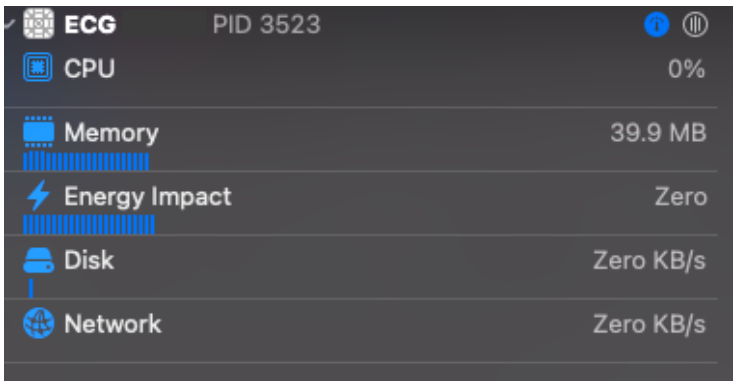

FIGURE 6. Xcode system resource monitoring

information leak (comply to PCI and HIPAA). In addition, the application utilizes low resource allocation reflected in Figure 6.

6. **Conclusion and Future Work.** Based on the result shown in the previous section, the proposed solution shows that ECG data transmission is secured from iOs app to the server. The report does not show signs of security vulnerabilities from listed attacks in Section 2 as well. Furthermore, the app shows efficiency [15] in terms of resource utilization, including while performing the encryption. Current solution does not implement any machine learning tasks yet so it can be a future research direction.

Acknowledgement. Thanks to BINUS University for facilitating and supporting the students to publish their research. Thanks to Mr. Sani Muhamad Isa for guiding me in this research. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. Asemani, F. Abdollahei and F. Jabbari, Understanding IoT platforms: Towards a comprehensive definition and main characteristic description, *2019 5th International Conference on Web Research (ICWR)*, 2019.

[2] I. Chakraborty, A. Chakraborty and P. Das, Sensor selection and data fusion approach for IoT applications, *Advances in Intelligent Systems and Computing Recent Developments in Machine Learning and Data Analytics*, pp.17-33, 2018.

[3] K. Z. Latt and S. H. Phyo, Analysis on healthcare system using IoT, *International Journal of Trend in Scientific Research and Development*, 2019.

[4] A. N. Koshy, J. K. Sajeev, N. Nerlekar, A. J. Brown, K. Rajakariar, M. Zureik, M. C. Wong, L. Roberts, M. Street, J. Cooke and A. W. Teh, Smart watches for heart rate assessment in atrial arrhythmias, *International Journal of Cardiology*, vol.266, pp.124-127, 2018.

[5] J. M. Bumgarner, C. T. Lambert, A. A. Hussein, D. J. Cantillon, B. Baranowski, K. Wolski, B. D. Lindsay, O. M. Wazni and K. G. Tarakji, Smartwatch algorithm for automated detection of atrial fibrillation, *Journal of the American College of Cardiology*, vol.71, no.21, pp.2381-2388, 2018.

[6] K. R. Foster and J. Torous, The opportunity and obstacles for smartwatches and wearable sensors, *IEEE Pulse*, vol.10, no.1, pp.22-25, 2019.

[7] N. Isakadze and S. S. Martin, How useful is the Smartwatch ECG?, *Trends in Cardiovascular Medicine*, vol.30, no.7, pp.442-448, 2020.

[8] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant and K. Mankodiya, Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare, *Future Generation Computer Systems*, vol.78, pp.659-676, 2018.

[9] P. K. Dhillon and S. Kalra, A secure multi-factor ECC based authentication scheme for Cloud-IoT based healthcare services, *Journal of Ambient Intelligence and Smart Environments*, vol.11, no.2, pp.149-164, 2019.

[10] S. Ames, M. Venkitasubramaniam, A. Page, O. Kocabas and T. Soyata, Secure health monitoring in the cloud using homomorphic encryption: A branching-program formulation, in *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies*, T. Soyata (ed.), IGI Global, 2020.

[11] S. Chakrabarty and D. W. Engels, A secure IoT architecture for smart cities, *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2016.

[12] A. Majchrzycka and A. Poniszewska-Marańda, Secure development model for mobile applications, *Bulletin of the Polish Academy of Sciences Technical Sciences*, vol.64, no.3, pp.495-503, 2016.

[13] R. K. Chaturvedi and S. Chand, Multipath TCP security over different attacks, *Trans. Emerging Telecommunications Technologies*, vol.31, no.9, DOI: 10.1002/ett.4081, 2020.

[14] A. Banerjee, M. Hasan and H. Kafle, Secure cryptosystem using randomized rail fence cipher for mobile devices, in *Intelligent Computing. CompCom 2019. Advances in Intelligent Systems and Computing*, K. Arai, R. Bhatia and S. Kapoor (eds.), Cham, Springer, 2019.

[15] D. Rachmawati, M. A. Budiman and A. Yusuf, Combination of rail fence cipher algorithm and least significant bit technique to secure the image file, *IOP Conference Series: Materials Science and Engineering*, vol.851, DOI: 10.1088/1757-899X/851/1/012069, 2020.

[16] A. Singh, A. Nandal and S. Malik, Implementation of caesar cipher with rail fence for enhancing data security, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.2, no.12, pp.78-82, 2012.

[17] G. Lasry, N. Kopal and A. Wacker, Solving the double transposition challenge with a divide-and-conquer approach, *Cryptologia*, vol.38, no.3, pp.197-214, 2014.

[18] A. Bhowmick and M. Geetha, Enhancing resistance of hill cipher using Columnar and Myszkowski transposition, *International Journal of Computer Sciences and Engineering*, vol.3, no.2, pp.20-26, 2015.

[19] A. Banerjee, M. Hasan, M. A. Rahman and R. Chapagain, CLOAK: A stream cipher based encryption protocol for mobile cloud computing, *IEEE Access*, vol.5, pp.17678-17691, 2017.

[20] B. Bhushan, G. Sahoo and A. K. Rai, Man-in-the-Middle attack in wireless and computer networking – A review, *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA)*, 2017.

[21] H. Lan, X. Zhu, J. Sun and S. Li, Traffic data classification to detect Man-in-the-Middle attacks in industrial control system, *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, 2020.

[22] A. S. Khader and D. Lai, Preventing Man-in-the-Middle attack in Diffie-Hellman key exchange protocol, *2015 22nd International Conference on Telecommunications (ICT)*, 2015.