Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130618

Key Exchange Techniques Based on Secured Energy Efficiency in Mobile Cloud Computing

B. Karthikeyan[1,*], *T. Sasikala*[2] *and S. Baghavathi Priya*[3]

[1] Anna University, Chennai, India
[2] School of Computing, Sathyabama Institute of Science and Technology, Chennai, India
[3] Department of Information Technology, Rajalakshmi Engineering College, Chennai, India

Received: 6 May 2019, Revised: 11 Jun. 2019, Accepted: 14 Jun. 2019
Published online: 1 Nov. 2019

Abstract: In this paper, we propose a novel offloading algorithm to enhance the security in the Mobile Cloud Computing (MCC) by combining Advanced Encryption Standard (AES), Secured Hashing Algorithm (SHA)-256 and Diffie-Hellman key exchange techniques. In the proposed system, the application in the mobile device is divided to offloadable and non-offloadable portions, usually threads. The offloadable thread is now suspended and wrapped in the mobile device. In order to secure the transmission, the wrapped thread is added with the hash value generated from SHA-256 algorithm and the secret key will be exchanged with the help of Diffie-Hellman algorithm. To improve the energy efficiency, a novel offloading algorithm is proposed. The simulation results show that, our proposed algorithm provides better security when compared with other security algorithms and exhibits better energy efficiency.

Keywords: Mobile Cloud Computing, Security, Computation Offloading and Energy Efficiency

1 Introduction

MCC is the combination of cloud computing, mobile computing and wireless networks to provide computational resources to mobile users by extending the capabilities of the mobile device. Major issues faced by the users in wireless networks are lack of privacy and security. Security concerns involving authentication, authorization, digital signature, non-repudiation, encryption and privacy are becoming major issues [1]. Traditionally, in a secured communication, the secret key which is used for encryption needs to be transported over a secured channel to protect it against the hackers. But, the Diffie-Hellman algorithm is used to calculate a shared secret key between the sender and the receiver through the insecure public channel rather than sending it. This calculated key then can be used by both sender and receiver for encryption and decryption. Even though we have these algorithms for securing our communication, we need to have a more secured cryptographic algorithm for our increasing usability. So, there is an impounding need for a combined encryption scheme. A combined encryption scheme is the one that blends 2 or more security algorithms to provide highly secured communication.

Message Digest-based Authentication (MDA) is proposed. Where, they combined the hashing with traditional user ID and password for mutual authentication. This algorithm provided security against man-in-the-middle attacks and repays attack and so forth. Many issues related to the data security problems in cloud computing are investigated. They used 3 layer system structures in which, each layer is responsible for specific functions like user authentication, data encryption and fast recovery of user data [2].

Many issues related to the data security problems in cloud computing are investigated in [3], and they presented a data security model based on cloud architecture. They used 3 layer system structure in which each layer is responsible for specific functions like user authentication, data encryption and fast recovery of user data.

The combination of digital signature and Diffie-Hellman key exchange with AES is used. This scheme provided authentication, data security and verification at the same time. The combination of AES and Diffie-Hellman for authentication and secured key

* Corresponding author e-mail: karthikeyan.b@mail.com

exchange mechanism is used in [4] to secure the data while offloading from mobile to the server and vice versa in the MCC. Moreover, energy efficient offloading algorithm for minimizing the energy utilization in the mobile devices was proposed in [5].

The offloading frameworks and computation offloading techniques are presented in [6] where they analyzed these techniques with their critical issues. In addition, they analyzed different important parameters like offloading methods and level of partitioning and summarized the issues in offloading frameworks in MCC.

The overview of the secure offloading techniques and authentication for mobile devices were discussed in [7] and they also preserved the privacy of offloaded data in the cloud. They covered all the ways to offload the data in a secure manner.

A novel mobile cloud execution framework was proposed in [8] which is used to execute mobile applications in the cloud-based virtual environment. This framework can be controlled by users and a mobile app. They provided encryption and isolation to protect the data from eavesdropping. And also the communication issues were also addressed to support timely migration of application.

An energy-efficient multisite offloading algorithm was proposed in [9] to partition the application in multi way using graph partitioning algorithm. They adopted the multiway graph partitioning to solve the problem. They obtained better energy consumption and execution time.

A secure and cost efficient offloading scheme was proposed in [10]. In this work, they used the combination of regular rekeying and random padding. The timing attacks against MCC system is considered in their work and to the migration effectiveness of random padding.

A secure partitioning of mobile application was proposed in [11], in which most sensitive parts of the application were kept in the mobile device itself and the rest will be offloaded to the cloud server.

An energy-efficient offloading-decision algorithm based on Lyapunov optimization was proposed in [12] to balance the energy and delay tradeoff. This algorithm decides when to offload an application and minimizes the average energy consumption on the mobile devices. Various offloading techniques in MCC were analysed in [13] with their advantages and applications. Along with these, major issues related to offloading and they discussed the new research areas in the mobile cloud offloading.

A Cloud Manager-based Re-encryption Scheme(CMReS) was proposed in [14]. This scheme provided better security and reduced the processing load in the mobile devices. The turn around time and resource consumption were condensed by CMReS.

The detailed survey of security and privacy issues in the MCC were discussed in [15]. The recent works in the field of MCC security were presented with the solutions. Finally the open issues in this field were also discussed.

To ensure datra security of the users, task secure offloading model was proposed in [16]. They used encryption of sensitve data while offloading. A receding horizon-based online algorithm was proposed and they used pareto-optimal method for generating the task offloading strategy.

To support offloading decisions at runtime, two parallel algorithms were defined in [17] based on the depth-first and dijikstra algorithms. Offloading scheduling is generated at runtime improved the performance of the applications and reduced the energy usage.

A novel secured and optimized framework was proposed in [18]. This method improved the energy efficiency while offloading. Based on this framework, the offloading decision is made using a formulated 0–1 integer linear programming model. The decision is made dynamically at runtime based on execution time, energy consumption and so forth. Also they added a new security layer to protect the data using AES. A novel authentication scheme for MCC is presented in [19]. They used hashing scheme to enhance the security and integrity of the message.

2 The Proposed System

2.1 Secured Offloading

In the proposed system, the application in the mobile device is divided to offloadable and non-offloadable portions, usually threads. The offloadable thread is now suspended and wrapped in the mobile device. In order to secure the transmission, the wrapped thread is added with the hash value generated from SHA-256 algorithm. The proposed system architecture is as shown in Fig. 1. In the proposed architecture, we are using three ways of protection scheme. Firstly, to generate keys for key exchange step, Diffie Hellman algorithm is used. Then digital signature is used for authentication, there after, user's data file is encrypted or decrypted using AES algorithm. With this algorithm data will be uploaded into cloud server by double encryption. Initially, data will be encrypted using AES algorithm and again re-encryption will be done by SHA-256 and similarly data will be downloaded from the cloud server by decrypting the file as exactly the reverse of encryption process. All this is implemented to provide trusted network at the server end. For the same reason, two separate servers are maintained, one for encryption process known as (trusted) computing platform and another known as storage server for storing user data file. When a user wants to upload a file to the cloud server, first key is exchanged using Advanced Diffie Hellman key exchange at the time of login, then the client is authenticated using digital signature. Finally user's data file is encrypted using hybrid encryption algorithm and only then it is uploaded to cloud storage server. The client can download the same file from Cloud server. When a

user logins, first encryption keys are exchanged, the file to be downloaded is selected, authentication takes place using digital signature and AES and SHA-256 algorithm is used to decrypt the saved file and the client is allowed to access the file. This system architecture was explained below in pseudo codes.

2.1.1 Pseudo code: SHA2-512 Algorithm

Process the message in successive 512-bit chunks
Break message into 512-bit chunks for each chunk
Create a 64-entry message schedule array
$w[0,\ldots,63]$ of 32-bit words
Copy chunk into first 16 words $w[0,\ldots,15]$
of the message schedule array
Extend the first 16 words into the remaining 48 words
$w[16,\ldots,63]$ of the message schedule array
Initialize working variables to current hash value
Compression function
Add the compressed chunk to the current hash value
Produce the final hash value (big-endian)
To further enhance the security, the thread and the hash value are combined and the whole will be encrypted with AES algorithm with a secret key.

2.1.2 Pseudo code: AES-256 Algorithm

```
Cipher (byte in[4*Nb], byte out[4*Nb],
    word w[Nb*(Nr + 1)])
Begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[0, Nb-1])
for round = 1 step 1 to Nr–1
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state, w[round*Nb, (round + 1)*Nb-1])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[Nr*Nb, (Nr + 1)* Nb-1])
out = state
end
```

Now the encrypted thread is sent to the server for execution and the secret key will be exchanged with the help of Diffie-Hellman algorithm to secure the transmission further. In the server side, the encrypted message is received and is decrypted using the secret key. Then hash value is generated for the message and it is compared with the received hash value. If both hash values match, the virtualized software in the cloud server will separate the thread and hash value, execute the thread and return back the result. The entire process is shown in Fig. 1.

2.1.3 Psuedo Code for the Proposed Secured Offloading

At Mobile Node:

1.Let user task $T = T_1, T_2, \ldots, T_n$
2.The offloader selects T_x for offloading.
3.Let $M = T_x \| uid \| pwd$.
4.Calculate the Hash value & attaches it to M.
Let $M_1 = M \| H(M)$.
5.Encrypt M_1 with the secret key, K.
$C_t = E_k(M_1, K)$
6.Send the C_t to the cloud server and distribute the key(K) through secured channel.

At cloud server:

1.receives C_t and K
2.Decrypt C_t with the key(k)
$M_1 = D_k(C_t, K)$
3.Split the message(M_1) and hash value. $M_1 = M \| H(M)$
4.Generate the hash value for $M(H_1(M))$.
5.Compare $H_1(M)$ and $H(M)$. If it matches, execute the thread and return the result.

2.2 Energy Efficient Offloading

The mobile node sends a task to the cloud server for processing/execution. The cloud server receives the request from all the mobile nodes connected to it and starts processing them. Meanwhile, the mobile node runs other local tasks while waiting for the results from the cloud server, or simply goes idle if there are no other tasks to be performed. When the cloud server completes the execution of the task, it sends a frame with a list of IDs whose task has been completed. If the mobile node gets its ID in the message, it wakes up to receive the message.

2.2.1 Pseudo Code

1.Mobile device wraps a thread for execution to the server and it goes to sleep mode.
2.The server creates a virtual software for the mobile device.
3.The thread is executed in the virtualized mobile at the cloud server.
4.Upon completion, server broadcasts the list of addresses (whose thread has been completed) to all the mobile nodes.
5.Upon reception of the message, if it consists of mobile node's address, then mobile node will wake up and be ready to receive the result.
6.The server then sends the result to all the mobile nodes which are waiting.

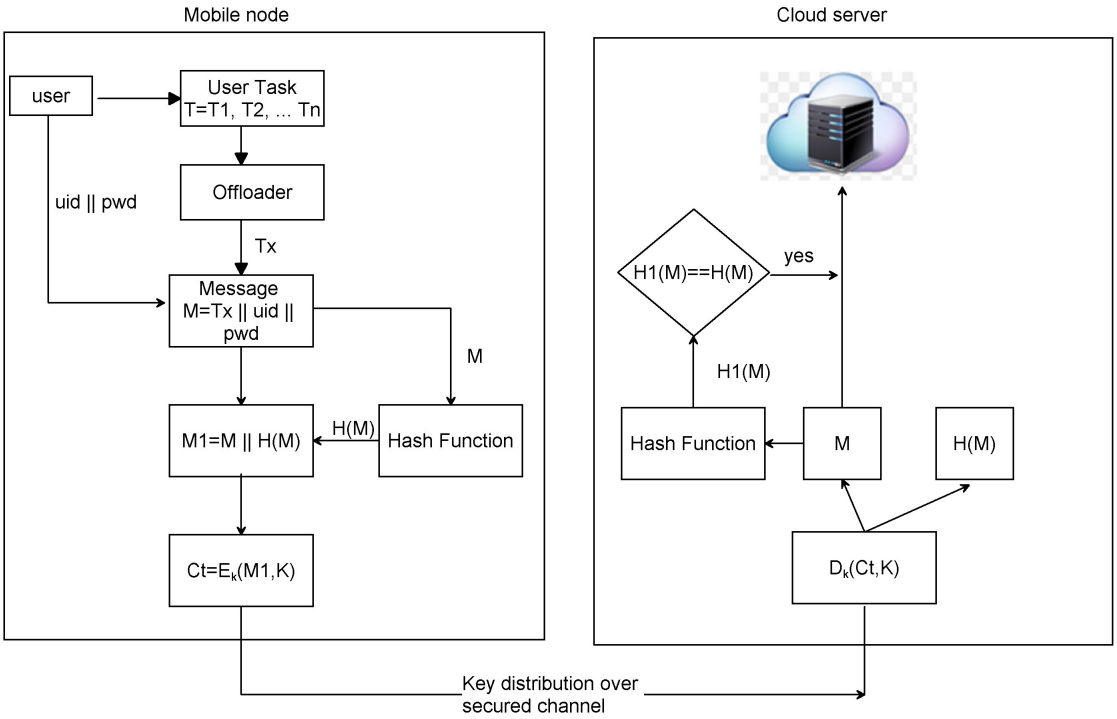

Fig. 1: Proposed system architecture

3 Performance Analysis

3.1 Security Value

The Security value is used to measure the secure value of a cipher. It is denoted as H(m), the amount of information in the message. A comparison of the average security value shown is in Fig. 2.

It is calculated as:

$$H(m) = -\sum_{0 \le i \le n-1} p(m_i) \log 2 p(m_i)$$

where $P(m_i)$ represents probability of m_i. If every symbol in the message has an equal probability, i.e., $m = \{m_0, m_1, m_2, \ldots, m_{255}\}$, then the security value $H(m) = 8$. It is an ideal value of security value for message source m. For a better encryption scheme, the security value of encrypted data, close to the ideal case, is expected.

Table 1 shows the average security value for our proposed algorithm in comparison with the existing algorithms like Digital Encryption Standard (DES), Triple DES(SHA-256), Rivest Cipher(RC2),AES and BLowFish(BF).

Table 1: Security value

Security algorithms	DES	SHA-256	RC2	AES	BF	Proposed Algorithm
Average Security Value	5.25	6.34	6.77	7.60	4.01	7.669

Fig. 2: Comparison of average security value

Table 2: Encryption and decryption time (in ms)

Data length (ASCII character)	AES Encryption time(in ms)	Proposed Encryption time	AES Decryption Time	Proposed Decryption Time
16	47	51	62	65
32	63	67	101	108
48	79	84	141	152
96	155	167	267	285
192	316	325	548	554
288	467	475	828	836
384	625	659	1097	1121
480	786	799	1379	1401
576	955	980	1653	1673
672	1111	1211	1941	1961

Table 3: Energy Consumption

Algorithm	Energy consumption (in %)
Local Execution	100
EEODA	58
Proposed algorithm	51

3.2 Vulnerability Score (S_v)

To calculate the number of attacks our scheme can prevent, a measure called vulnerability score (S_v) is used. The value of S_v is calculated as:

$$S_v = N_{\text{success}}/N$$

where N is the total number of attacks and N_{success} is the number of successful attacks.

Our proposed algorithm is validated in Scyther, a tool for checking the vulnerability. This is shown in Fig. 3 and Fig. 4 respectively. To compromise the system, this tool generates different types of attacks. Our proposed algorithm is executed 10 times in this tool and no attacks were found.

So vulnerability score is:

$$S_v = 0/10 = 0.0$$

which shows that our algorithm provides more security.

Table 2 provides the comparison of encryption time and decryption time for both AES and the proposed algorithm. The result shows that the encryption and decryption time of our algorithm is more than those with AES algorithm because our algorithm combines AES and Diffie Hellman for encryption and SHA-1 for message digestion.

Table 3 and Fig. 5 provide the energy consumption level of our proposed algorithm in comparison with the existing Energy-Efficient Dynamic Offloading Decision Algorithm (EEODA). The result shows that our algorithm provides lower energy consumption in mobile nodes.

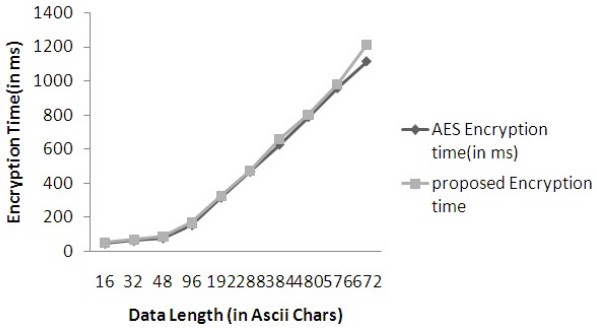

Fig. 3: Data length vs. encryption time

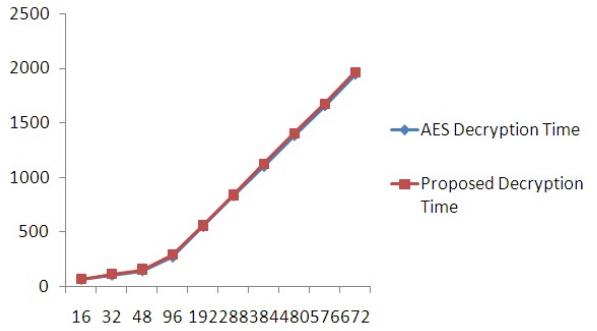

Fig. 4: Data length vs. decryption time

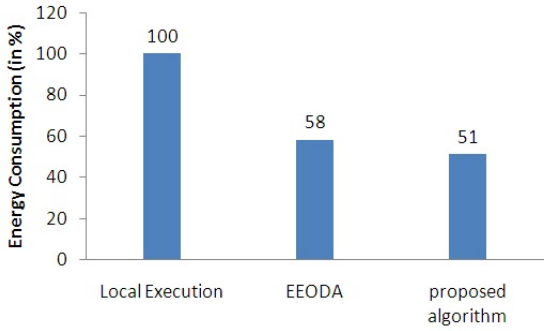

Fig. 5: Energy consumption

4 Conclusion

An offloading algorithm with improved security is proposed. In this paper, the SHA-256 algorithm is used with the combination of AES and Diffie-Hellman. Moreover, an energy-efficient offloading algorithm is also proposed to minimize the energy consumption while offloading. The results show that our proposed scheme exhibits better security and better energy utilization. The result of the proposed algorithm is compared with the existing algorithms like DES, SHA-256, RC2, BF and it has been observed that the security and the energy efficiency are comparatively improved.

References

[1] Q. Fan and L. Liu, A Survey of Challenging Issues and Approaches in Mobile Cloud Computing, 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, 87–90, (2016).

[2] Saurabh Dey, Srinivas Sampalli1 and Qiang Ye MDA: message digest-based authentication for mobile cloud computing, Journal of Cloud Computing: Advances, Systems and Applications, **5**(1), 1–13 (2016).

[3] E. M. Mohamed, H. S. Abdelkader and S. El-Etriby, Enhanced data security model for cloud computing, 8th International Conference on Informatics and Systems (INFOS), Cairo, CC-12-CC-17, (2012).

[4] P. Rewagad and Y. Pawar, Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing, International Conference on Communication Systems and Network Technologies, Gwalior, 437–439 (2013)

[5] B. Karthikeyan, Dr. T. Sasikala and K. Nithya, Secure And Energy Efficient Model With Modified Offloading Algorithm In Mobile Cloud Computing, Asian Journal of Research In Social Sciences And Humanities, **6**(9), 575–594(2016).

[6] Khadija Akherfi, Micheal Gerndt and Hamid Harroud. Mobile cloud computing for computation offload-ing: Issues and challenges. Applied Computing and Informatics, Elsevier, **14**(1), 1–16 (2018).

[7] D. Shibin and G. J. W. Kathrine, A comprehensive overview on secure offloading in mobile cloud computing, 4th International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 121–124 (2017).

[8] Shih-Hao Hung, Chi-Sheng Shih, Jeng-Peng Shieh, Chen-Pang Lee and Yi-Hsiang Huang, Executing mobile applications on the cloud: Framework and issues, Computers & Mathematics with Applications, **63**(2), 573–587 (2012).

[9] R. Niu, W. Song and Y. Liu, An Energy-Efficient Multisite Offloading Algorithm for Mobile Devices. International Journal of Distributed Sensor Networks, 1–6 (2013).

[10] Tianhui Meng, Katinka Wolter, Huaming Wu and Qiushi Wang, A secure and cost-efficient offloading policy for Mobile Cloud Computing against timing attacks, Pervasive and Mobile Computing, 4–18 (2018).

[11] N. M. Dhanya and G. Kousalya Adaptive and Secure Application Partitioning for Offloading in Mobile Cloud Computing, Springer International Publishing Switzerland, CCIS, **536**, 45–53 (2015).

[12] Wu, Huaming, Sun, Yi and Wolter, Katinka, Energy-Efficient Decision Making for Mobile Cloud Offloading IEEE Transactions on Cloud Computing.

[13] Abhishek Bajpai and Shivangi Nigam, A Study on the Techniques of Computational Offloading from Mobile Devices to Cloud Advances in Computational Sciences and Technology, **10**(7), 2037–2060 (2017).

[14] Abdul Nasir Khan, M. L. Mat Kiah, Mazhar Ali and Shahaboddin Shamshirband, A Cloud-Manager-Based Re-Encryption Scheme for Mobile Users in Cloud Environment: a Hybrid Approach Journal of Grid Computing, **13**(4), 651–675 (2015).

[15] Mollah Muhammad Baqer, Azad Md and Vasilakos Athanasios, Security and Privacy Challenges in Mobile Cloud Computing: Survey and Way Ahead Journal of Network and Computer Applications, **84**(4), 38–54 (2017).

[16] Y. Luo, J. Wu, Z. Zhang, W. Shi and Y. Miu, Online Algorithm for Secure Task Offloading in Dynamic Networks, IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications, (ISPA/IUCC), 66–71 (2017).

[17] A. Morichetta, B. Re and F. Tiezzi, Runtime Computation of Optimal Offloading Scheduling, 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Bamberg, 73–78 (2018).

[18] I. Elgendy, W. Zhang, C. Liu and C. Hsu, An Efficient and Secured Framework for Mobile Cloud Computing, in IEEE Transactions on Cloud Computing, (to be published).

[19] S. Dey, S. Sampalli and Q. Ye, Message digest as authentication entity for mobile cloud computing, IEEE 32nd International Performance Computing and Communications Conference*(IPCCC)*, San Diego, CA, 1–6, (2013).

B. Karthikeyan graduated B.E. Computer Sceince and Engineering from Madurai Kamarajar University, M.Tech. Information Technology from Sathyabama University. Currently he is pursuing Ph.D. in Computer Science majoring in Anna University. He is working as an Associate Professor in the Department of Information Technology at Panimalar Engineering College, Chennai. His areas of specialization include Network security, Data Mining and Mobile cloud computing security.

T. Sasikala received the B.E. CSE, M.E. CSE degree, and the Ph.D. degree from the Sathyabama University, Chennai, Tamil Nadu India. She is having 20 years of experience in teaching. Currently she is Professor & Dean for School of Computing, Sathyabama Institute of Science and Technology, Chennai. Her research interests are networks, wireless sensor networks, and heterogeneous wireless networks. Dr. T. Sasikala is a Life Member of CSI. She has published 40 papers in various conferences, including the IEEE International Conference and journals. Currently 16 scholars are doing research under her guidance.

S. Baghavathi Priya working as a Professor in Information Technology at Rajalakshmi Engineering College since 2008. She graduated B.E. Computer Science & Engineering from Manonmaniam Sundranar University, M.Tech. Computer Science & Engineering from Dr. M.G.R. Educational and Research Institute. She received Ph. D. from Jawaharlal Nehru Technological University Hyderabad. She has guided many U.G. and P.G. projects. She has published over 40 peer reviewed research articles. Her research is focused on Grid Computing, Network Security, Machine learning and Big Data Analytics. She received gold medal in M.Tech. degree. She received best paper award in ICTIS 2015 at Ahmedabad and in CAASR International conference 2017 at Dubai. She visited several countries for presenting papers and chairing sessions. She is a Life time member in CSI and IAENG.