International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8, Issue-2S2, July 2019

Man in the Middle Attack Prevention for Edge-Fog, Mutual Authentication Scheme

Gohar Rahman, Chuah Chai Wen

***Abstract:** Fog computing is considered emerging technology nowadays. Due to proximity to the end user, fog computing provides a reliable transmission with low latency. In this paper, we have proposed an improved mutual authentication security scheme based on advanced encryption standard AES and hashed message authentication code HMAC in fog computing. Our scheme provides mutual authentication between edge devices and fog server in edge fog cloud computing environment. Further, the scheme has the resistance to man in the middle attack in the fog computing environment. Detailed security analyses are summarized.*

***Index Terms:** Fog computing, Internet of Things, Mutual Authentication, Man In the Middle Attack.*

I. INTRODUCTION

The internet of things is a broad collection of multiple devices that connect to and communicate with applications. To increase the performance, the generated information must be compute and analyze in real time manner [2]. The development of all IoT devices has resulted in the generation of all data volumes that retain a large amount of computing resources, storage spaces and high bandwidth. Some internet of things applications may need to response quickly, some of them may involve secret data, which have necessary to processed and stored locally, and some may generate massive amount of volumes of data which may be a heavy weighted for the network. In addition, more devices are involved (such as smart glasses, smart phones and vehicles) [3].

In order to process all generated IoT device data with sufficient network and computing infrastructure, cloud computing is considered a perfect choice. Cloud computing is a rapidly evolving Internet-based technology that shares computing resources on demand. In cloud computing, end users do not understand where data is stored and how it is handled. They only access the data, process it, and eventually store it in the cloud. If they have an internet connection, they can access the data anytime, anywhere. The technology is highly scalable, flexible and distributed. In cloud computing, computational resources are provided as services to end users. [4]. Cloud computing has been used as an efficient way to process data because of its high computation power and storage capability.

Additionally, since the cloud computing paradigm is a centralized computing model, most of the calculations happen in the cloud.

This means that all data and requests need to be transferred to a centralized cloud. However many IoTs based application require real time response. For example industrial internet and health monitoring system needs high real time, reliability and high efficiency in millisecond. Therefore, cloud computing under the internet of things cannot fulfil requirement high mobility, location awareness and low latency requirements services [5].

Fog computing introduced by cisco [6] to eliminate the aforementioned problems in cloud computing. Fog computing work as a middle layer in cloud IoT architecture where the computations perform by fog computing very close to the end user. The main aim of fog computing is to provide better efficacy and reduced the bandwidth. When the IoT devices senses the data, sent that data to the fog computing for further processing. The fog computing processed the data locally and response back to the end user, while for long storage the data is to be send directly to the cloud computing. Thus fog computing reducing the network traffic and latency [7]. As mentioned before Fog computing is considered a middle layer in IoT cloud architecture as shown in Fig 1. The first layer of fog computing consist of IoTs devices such as end user devices, hand held devices. These IoTs devices also call terminal nodes.

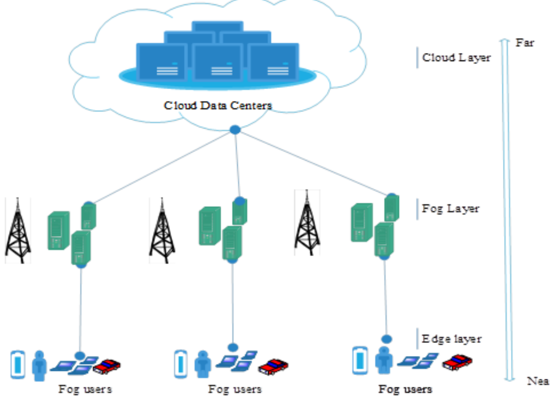

Fig. 1 Architecture of fog computing

Fog computing is the second layer which provider interface to IoT devices to connect with cloud computing. In this layer multiple network devices also called fog nodes (Router, switches, access points) coordinates and share storage and computing information. The top layer in fog computing environment contains a heavy data centers called cloud layer which has the capability to store large amount of data because this layer contain sufficient network and storage resources.

Revised Manuscript Received on June 22, 2019.

Gohar Rahman, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 68400 Parit Raja, Batu Pahat, Johor.

Chuah Chai Wen, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 68400 Parit Raja, Batu Pahat, Johor.

The rest of the paper is organized as follows. Section 2 contains benefits of fog computing, Section 3 summarize Application of fog computing. In Section 4 related works of fog computing are discussed.

Section 5 discussed proposed improved mutual authentication scheme. In Section 6 we discussed security analyses of our proposed improved scheme. Finally Section 7 counts conclusion.

II. BENEFITS OF FOG COMPUTING

Fog computing brings the services of cloud near to the edge. Even though the fog and the cloud uses related computing resources such like network, storage devices and collaborate many of the same tools and attributes (virtualization, multi-tenancy) . Therefore fog computing deliver many benefits for IoT devices. Some of the well-known significance of fog computing can be summarized as below.

Low Latency

Fog computing work near to the edge devices therefore it has the ability to support real time services such as gamming and video streaming.

Flexibility and heterogeneity

Fog computing allows collaboration of different physical environments and infrastructures among multiple services.

Scalability

The proximity of fog computing to terminal devices allows scaling the number of connected devices and services.

Geographical and large-scale distribution

Fog computing can provide distributed computing and storage resources to large and widely distributed applications.

III. APPLICATION OF FOG COMPUTING

There are several important areas in which fog computing contain vital role in various IoT application. Various application has been reviewed by the author in [8]. The important application of fog computing are discussing below.

Smart Grid

Smart grid is considering as the next generation electric power supplier computing network. The main component of smart grids are transmission lines, transformer and substations. The utilization of stream of power in smart grid is bidirectional to generate a very strengthen powerful energy distribution network. In smart grid, multiple service provider and customer can observed and control the pricing system, production and consumption of the system in real time manner [9]. The role of smart grid in fog computing is very important where millions of smart meters are fixed in the consumer home. Fog collector also called fog node collecting the information near to the edge where this information are processed by the fog node and filter them locally. If the information required for long storage, fog node sent the data directly to the cloud server [10].

Health Care System

In health care system the services and applications are very delay responsive and make secret information of the patient. The generated data of the patient include sensitive and individual data [11]. Fog computing bring the services in emergency situation where minimum latency required in medical services, ambulance communications or to require the portable accessibility of the patient medical files. [12]. Fog computing gives more benefits to those patients who suffer from stork diseases.

Augmented Reality (AR)

Augmented reality brings digital and virtual things into the real world. Augmented reality requires low latency and high information processing rates to provide the correct information as specified by the client location [13]. The augmented reality applications are very intolerant for latency. Some delay in the response can damage the client experience. Fog computing is designed to provide low latency services [14].

Traffic Control System

Fog computing play vital role in traffic control system where the video camera fixed on road side senses the flashing light of an ambulance and can automatically change the traffic from one track to another. In this system intelligent street light collaborate locally with sensors and identify the person who’s going on footpath and examine the movement of cyclists. The sensor evaluate the distance and speed of the approaching vehicles. In addition the smart lighting system switched on and switched off when the traffic is going to be passes. Here the intelligent traffic light work as a fog devices measure the distance and coordinate to create green traffic signal and send the green traffic signal to approaching vehicles [15]. This system is more useful to maintain the traffic in a steady wise manner and prevent the accidence.

IV. RELATED WORKS

Stojmenovic et al. [16] used two techniques. The aim of the techniques is to achieve authentication and authorization in fog computing environment where the connection between fog server and cloud is fragile. The author presented in the first techniques, the CSP server transfer its own secret to the fog, so that later authentication is possible. They tested energy consumption in fog computing environment due to man in the middle attack. Secondly they present another hybrid encryption technique, but no experimental **results** for both the scheme are present.

In [17] a framework was proposed based on policy based security. The proposed framework aimed to support secretly share information, cooperation and data reuse in a fog computing environment. The framework uses attribute-based authentication. First the framework recognize the user and then on the basis of user attribute user can access the resources or services. Secondly the framework consist of a number of modules. These modules are used to this framework recognize the users and confirm that the user's attributes authorize him to access a particular resource or services. The framework contains a number of modules. The basic aim of these modules is to define rules and stored them for the user services and request also for sending information to different type of fog nodes.

The author further explained about the modules that it is important during real time computation and one of the module is called policy enforce module, and is allocated in fog node, cloud data server and edge devices. Therefore, as the authors say, this is a preliminary framework that does not take into account all the nuances of the federal fog ecosystem. To solve the anonymity problem of Maged et al [18]. The author in [19] presented a model. The scheme is resistance to many attack, however. As this model is based on public key cryptosystem. Though, public key cryptosystems have expensive computation that are considered impractical in fog end user equipment due to the inherent characteristics of the end user design (i.e., limited memory, processing, and battery power). Due to this reason the design goal of this scheme is not fit for mutual authentication between fog user and fog server.

Octopus: An Edge-Fog Mutual Authentication Scheme

In this section we will briefly overview the Maged scheme [18]. The scheme consist of three stages, the explanation is given below in detail.

The First Stage

In this stage system is initialized. Registration authority RA contains his own public key Pk_{RA} and Sk_{RA}. The public key Pk_{RA} is known to all fog servers FS. Where every server in fog environment have their own public and private keys (Pk_{FS}, Sk_{FS}). The RA know the identity of Fog servers FS in advance.

The Second Stage

This stage of the scheme is called registration phase. Where every fog user FU registered himself/herself with the registration authority RA. The registration authority is the responsible secure control server assigning and securely stored all security parameters used during communication between fog user and fog server. The registration authority maintain the identity of fog network F and fog servers FS. In this phase the important steps are given below.

Step 1: Fog user send the identity ID_{FU} to The registration authority RA.

Step 2: Registration Authority RA select a master secret key K_{FU} for fog user.

Step 3: Fog user FU stored the master secret key K_{FU}.

Step 4: The registration authority computes the fog server FS secret key K^{FU-FS} for fog user FU.

Step 5: Finally registration authority RA sends the Identity of Fog user ID_{FU} and the computed secret key K^{FU-FS} encrypted under the public key of RA and sign with the signature key of RA Sk_{RA} to each fog server FS located in fog network

The Third Stage

This stage of the scheme is known authentication phase. Where fog user FU and fog servers FS authenticated each other and share the session key for secure communication. The important steps of this stage are given below.

Step 1: Fog user generate random number rFU.

Step 2: Fog user FU send identity IDFU and random generated number rFU to the fog server FS

Step 3: Fog server check the identity of fog user IDFU. If the identity IDFU is registered with the fog server FS .The fog server FS will accept the request otherwise reject the request.

Step 4: Fog server FS fetches secret key K^{FU-FS} for fog user

Step 5: Fog server FS pick random nonce r_{FS}

Step 6: Using symmetric encryption fog server FS computes the encryption E pk_{FS} (k^{FU-FS}), (r_{FU}, r_{FS}).

Step 7: Fog server FS reply to Fog user FU with the Tuple include ID_{FU}, ID_F, ID_{FS}, E pk_{FS} (k^{FU-FS}), (r_{FU}, r_{FS})

Step 8: Fog user FU when received the tuple from Fog server FS, Then calculate the symmetric key k^{FU-FS} locally by input the ID_F, ID_{FS}, and K_{FU} to the hash function. i.e, $k^{FU-FS}= H (ID_F, ID_{FS}, K_{FU})$.

Step 9: Fog user FU decrypt the received tuple received from Fog server FS and decrypt it. Check the validity of r_{FU} if the match fail fog user FU abort the communication otherwise continue.

Step 10: Fog user FU pick the session key and compute the encryption E (k^{FU-FS}, r_{FS}, K_S)

Step 11: Fog user FU reply to fog Server FS including tuple (ID_{FU}, ID_F, ID_{FS}, E (k^{FU-FS}, r_{FS}, K_S)

Step 12: This is the final step where fog server FS decrypt the received tuple and check for validity of random nonce r_{FS} and K_S, if the value of random nonce matched the fog server FS will accept otherwise abort the communication.

Weakness of the Maged Scheme

In this section we point out the Maged scheme is cannot resist the man in the middle attack. The adversary in the middle easily replaces the identity of Fog user FU. In the Maged scheme we assume theirs exist an eavesdropper who attempts to become a legal user superseding to the original user. We show the man in the middle attack on octopus scheme in Fig 2. & Fig.3. The attacker attacked in second and third stage of the scheme as mentioned above. The procedure for that is given below.

The Second Stage

Step 1: When the fog user want to register with the RA. Fog user send identity ID_{FU} to the registration authority RA through public channel. The eavesdropper in the middle intercept the Identity of Fog user ID_{FU} replace as the identity as ID_{FU}'.

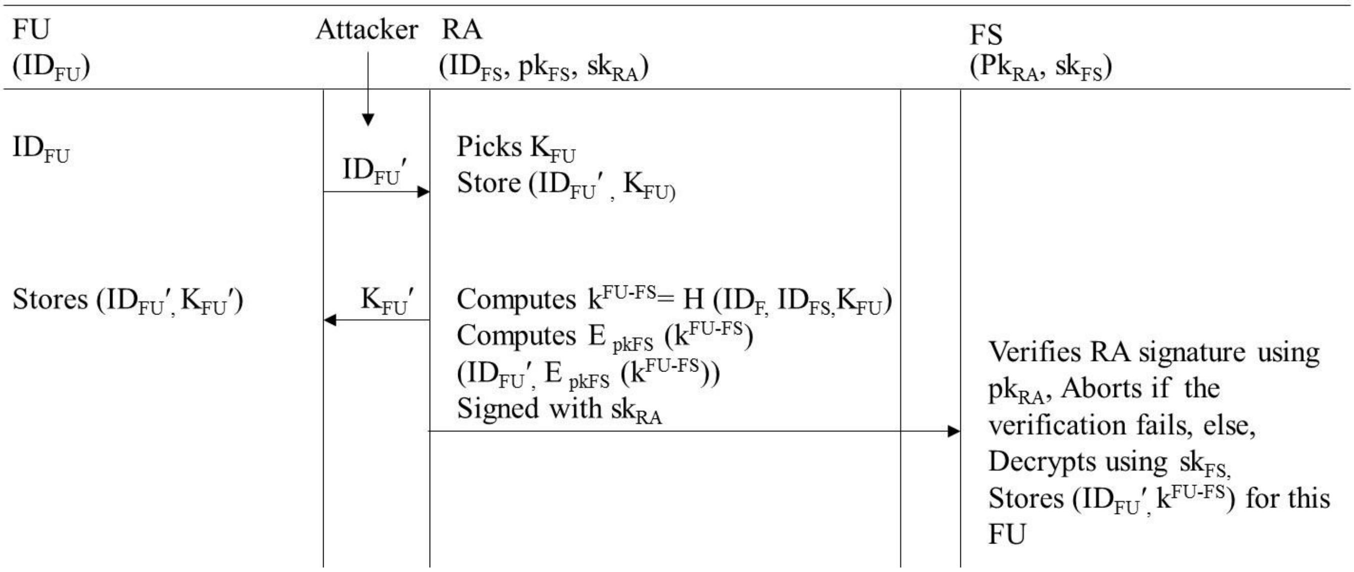
FU
(ID_{FU})
Attacker
RA
$(ID_{FS}, pk_{FS}, sk_{RA})$
FS
(Pk_{RA}, sk_{FS})
ID_{FU}
ID_{FU}'
Picks K_{FU}
Store (ID_{FU}', K_{FU})
Stores (ID_{FU}', K_{FU}')
K_{FU}'
Computes $k^{FU\text{-}FS} = H(ID_F, ID_{FS}, K_{FU})$
Computes $E_{pkFS}(k^{FU\text{-}FS})$
$(ID_{FU}', E_{pkFS}(k^{FU\text{-}FS}))$
Signed with sk_{RA}
Verifies RA signature using pk_{RA}, Aborts if the verification fails, else, Decrypts using sk_{FS}, Stores $(ID_{FU}', k^{FU\text{-}FS})$ for this FU

Fig. 2 Man In The Middle attack on Maged Scheme during registration phase

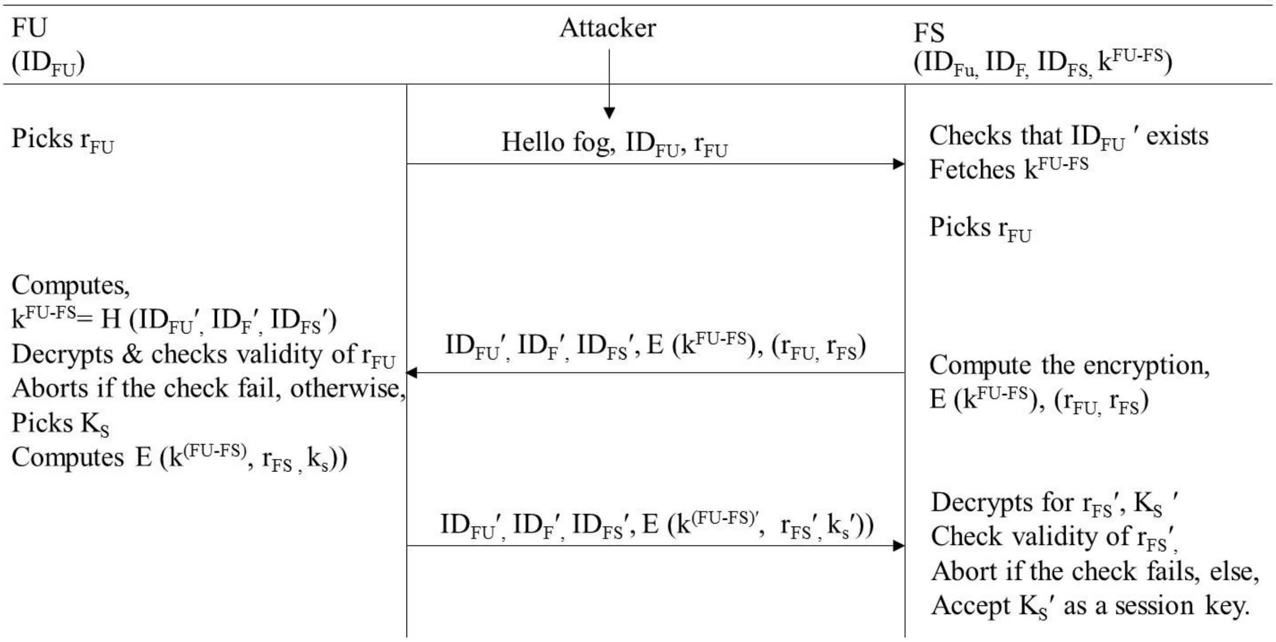
FU
(ID_{FU})
Attacker
FS
$(ID_{Fu}, ID_F, ID_{FS}, k^{FU\text{-}FS})$
Picks r_{FU}
Hello fog, ID_{FU}, r_{FU}
Checks that ID_{FU}' exists
Fetches $k^{FU\text{-}FS}$
Picks r_{FU}
Computes,
$k^{FU\text{-}FS} = H(ID_{FU}', ID_F', ID_{FS}')$
Decrypts & checks validity of r_{FU}
Aborts if the check fail, otherwise,
Picks K_S
Computes $E(k^{(FU\text{-}FS)}, r_{FS}, k_s))$
$ID_{FU}', ID_F', ID_{FS}', E(k^{FU\text{-}FS}), (r_{FU}, r_{FS})$
Compute the encryption,
$E(k^{FU\text{-}FS}), (r_{FU}, r_{FS})$
$ID_{FU}', ID_F', ID_{FS}', E(k^{(FU\text{-}FS)'}, r_{FS}', k_s'))$
Decrypts for r_{FS}', K_S'
Check validity of r_{FS}',
Abort if the check fails, else,
Accept K_S' as a session key.

Fig. 3 Man In the Middle Attack on Maged scheme during authentication phase

Step 2: The registration authority stores the intercepted identity ID_{FU}' and pick a master secret key K_{FU} and send to the attacker. The attacker send the compromised master secret key $K_{FU}{}'$ to the Fog user. In this stage, the attacker can easily gain the master secret key K_{FU}'.

Step 3: The registration authority also computes encryption for fog server and send the compromised identity ID_{FU}' of fog user FU to server. i.e. $(ID_{FU}', E\ pk_{FS}\ (k^{FU\text{-}FS}))$ Signed with sk_{RA}.

Step 4: The fog server FS store the compromised Identity ID_{FU}' of fog user

The Third Stage

The attacker in this stage comes in the middle when fog user FU and Fog server FS authenticate each other and communicate the session key. The attacker action are given below in the following steps.

Step 1: when fog user sends random nonce r_{FU} and identity ID_{FU} to the fog server FS. The attacker in the middle compromised the communication and get the identity ID_{FU} and replace as ID_{FU}'. The attacker shows himself/herself to fog server that the sending identity is from fog user but actually, it is the compromised identity by the attacker.

Step 2: The Fog server check the identity of the attacker ID_{FU}'. The matching will become successful because fog server FS have already stored compromised identity ID_{FU}' during registrations stage.

Step 3: The Fog server FS encrypt the random nonce of fog user FU and fog server FS by using the secret symmetric key $k^{\ FU\text{-}FS}$. Reply back with ID_{FU}, ID_F, ID_{FS}, $E\ (k^{FU\text{-}FS}), (r_{FU}, r_{FS})$.

The attacker replaces the received message from the fog server FS as ID_{FU}', ID_F', ID_{FS}', E (k^{FU-FS}), (r_{FU}, r_{FS}) and send to the fog user FU. Here the fog user FU cannot understand that the message is come from the attacker but it think that it is the actual message from the fog server FS.

Step 4: The fog user FU computes the secret key locally i.e. K^{FU-FS}= H (ID_F', ID_{FS}', K_{FU}'). Encrypted the session key by using secret key K^{FU-FS}. Here the identities ID_F', ID_{FS}', and K_{FU} ′ are not selected by the authorized originator but these are replaces by the attacker. Therefore the attacker can easily takes control on the session key, and encrypt decrypt the confidential information between fog user FU and Fog server FS. Thus the Maged et all scheme failed due to Man In The Middle Attack.

V. PROPOSED IMPROVED MODEL

This section introduced our proposed improved mutual authentication security scheme in fog computing environment. The proposed improved mutual authentication security scheme provides a light weight strong mutual authentication between the fog user and fog servers. Our security scheme is divided into three section/phases. The registration phase where all fog server and fog user must registered with central control server also called registration authority. While in authentication phase, every fog user and fog server must authenticate each other before sharing the session key. The third phase of our scheme is session key establishing. Where session key is established between fog user and fog server locally in securely manner. The explanation of the phases involve in our proposed mutual authentication security scheme are below.

Registration Phase

In the registration phase every fog user registered with the registration authority. When fog user wish to communicate with the fog server. It must register through registration authority. For registration, fog user must have valid identity. Figure 4 show the registration phase. The registration steps are discussing below.

Step 1: The fog user encrypts the identity ID_{FU}, of fog user FU by using shared symmetric key K_{FU-RA}.

Step 2: Fog user FU generate tag T_1 from cipher text C and generated random nonce n_1 by using a Hash message authentication code HMAC. Send the tag T_1 and C to the registration authority.

Step 3: On receiving the tag T_1 and C, the registration authority RA compute tag T_1' using HMAC. If the HMAC value become equal, decrypt C by using shared symmetric key K_{FU-RA} and Store the identity of fog user.

Step 4: Registration authority pick a master secret key $K_{FU-FS}$for fog user. Encrypt the master secret key with the fog user generated nonce value n_1 and calculate the tag T_2 from the cipher text C and send back to the fog user.

Step 5: Fog user compute the HMAC tag T_2' from the received value, if the value of tag T_2=T_2' become equal fog user decrypt and store the master secret key called k^{FU-FS}

Step 6: The registration authority also pick $K_{FU-FS,}$ identity of fog server ID_{FS}, Identity of fog area ID_F. Send the K_{FU-FS}, ID_{FS}, ID_F and ID_{FU} encrypted them by the public key PK_{FS} and sign with the signature key Sk_{RA}.

Step 7: The fog server verify the signature by using the public key Pk_{RA}. If the verification done successfully. The fog server decrypt the received tuples by using his own symmetric secret key Sk_{FS} and store ID_{FU}, ID_F, ID_{FS}, K_{FU-FS} .

Authentication Phase and key establishment Phase

This phase aims to mutually authenticate every fog user and fog server each other. Whenever every fog user in the network communicates with the fog server, they must execute the authentication process. The mutual authentication of fog user FU and Fog server FS are shown in Fig 5.Mutual authentication and session key establishment steps are discussing below.

Step 1: In this step fog user chose a random nonce value n_2 and sends it the message composed by the generated value n_2 and identity of fog user ID_{FU} by using HMAC to ensure the integrity and authenticity of the fog user FU.

Step 2: Upon receiving the secret parameters sending by the fog user FU. The received message is calculated by the fog server. The fog server verified the message by the associated HMAC value. If the verification successful, fog server FS also generate the random nonce n_3 and retransmit message consist of ID_F, ID_{FS}, n_3, HMAC tag T_4.

Step 3: When the fog user FU received the message from the fog server. It also verify HMAC. If the check become successful then the authentication process will be terminated.

Step 4: In this steps symmetric key SK^{FU-FS} is calculated by the fog user locally by using hash function H (ID_{FU}, ID_F, and K_{FU-FS}). Here the identity of fog user ID_{FU}, Identity of Fog network ID_F and master secret number K_{FU-FS} is used as input to the hash function to calculate the symmetric key for the encryption of the session key. This all process are done after, when the process of authentication is successfully complete. After this fog user pick session key K_S, encrypted them by using symmetric key SK^{FU-FS} and sends the parameter C_5 to the fog server.

Step 5: Upon receiving the encrypted session key fog user calculate the symmetric secret key by using hash function similar to the fog user. Decrypt the received message from the fog user, check for ID_F, ID_{FS}, ID_{FU} and if the matched successful the session key is accept and stored for secure communication.

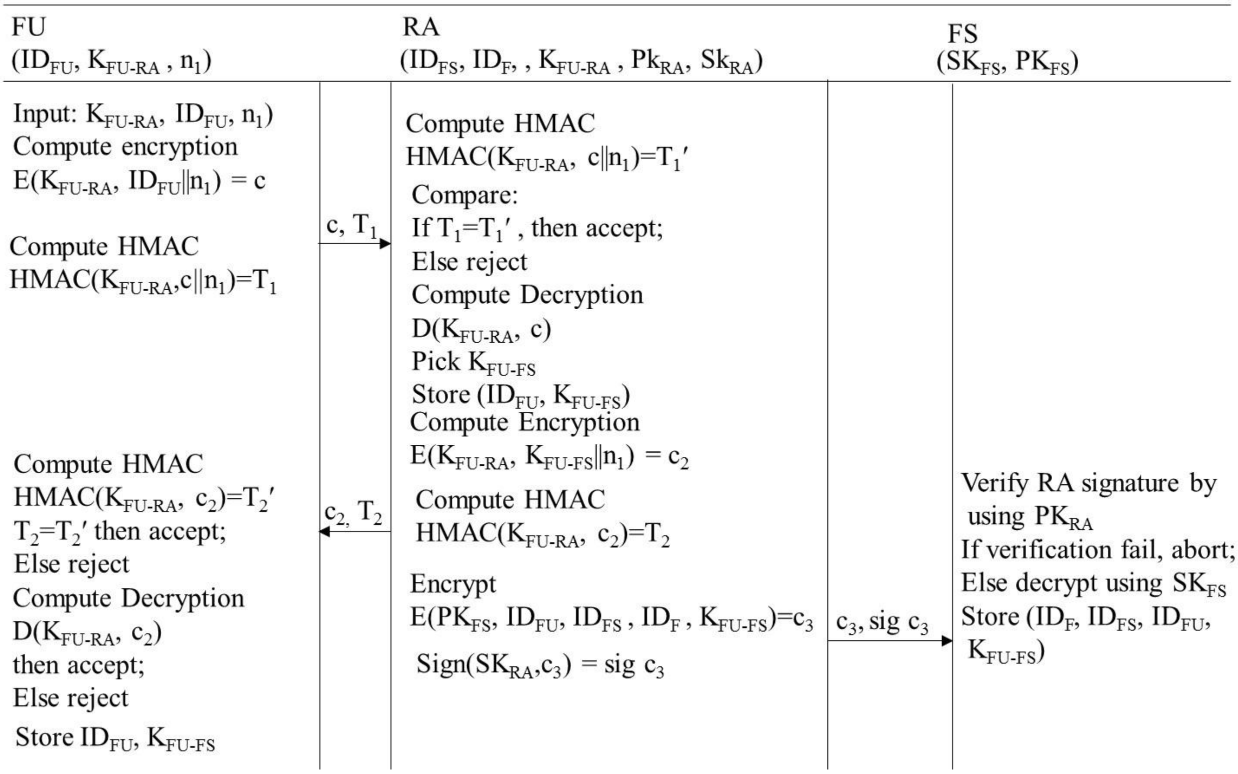
FU
$(ID_{FU}, K_{FU-RA}, n_1)$
RA
$(ID_{FS}, ID_F, , K_{FU-RA}, Pk_{RA}, Sk_{RA})$
FS
(SK_{FS}, PK_{FS})
Input: $K_{FU-RA}, ID_{FU}, n_1)$
Compute encryption
$E(K_{FU-RA}, ID_{FU}||n_1) = c$
Compute HMAC
$HMAC(K_{FU-RA},c||n_1)=T_1$
c, T_1
Compute HMAC
$HMAC(K_{FU-RA}, c||n_1)=T_1'$
Compare:
If $T_1=T_1'$, then accept;
Else reject
Compute Decryption
$D(K_{FU-RA}, c)$
Pick K_{FU-FS}
Store (ID_{FU}, K_{FU-FS})
Compute Encryption
$E(K_{FU-RA}, K_{FU-FS}||n_1) = c_2$
Compute HMAC
$HMAC(K_{FU-RA}, c_2)=T_2'$
$T_2=T_2'$ then accept;
Else reject
Compute Decryption
$D(K_{FU-RA}, c_2)$
then accept;
Else reject
Store ID_{FU}, K_{FU-FS}
c_2, T_2
Compute HMAC
$HMAC(K_{FU-RA}, c_2)=T_2$
Encrypt
$E(PK_{FS}, ID_{FU}, ID_{FS}, ID_F, K_{FU-FS})=c_3$
Sign(SK_{RA},c_3) = sig c_3
c_3, sig c_3
Verify RA signature by
using PK_{RA}
If verification fail, abort;
Else decrypt using SK_{FS}
Store $(ID_F, ID_{FS}, ID_{FU},$
$K_{FU-FS})$

Fig. 4 Registration phase of the proposed improved mutual authentication scheme

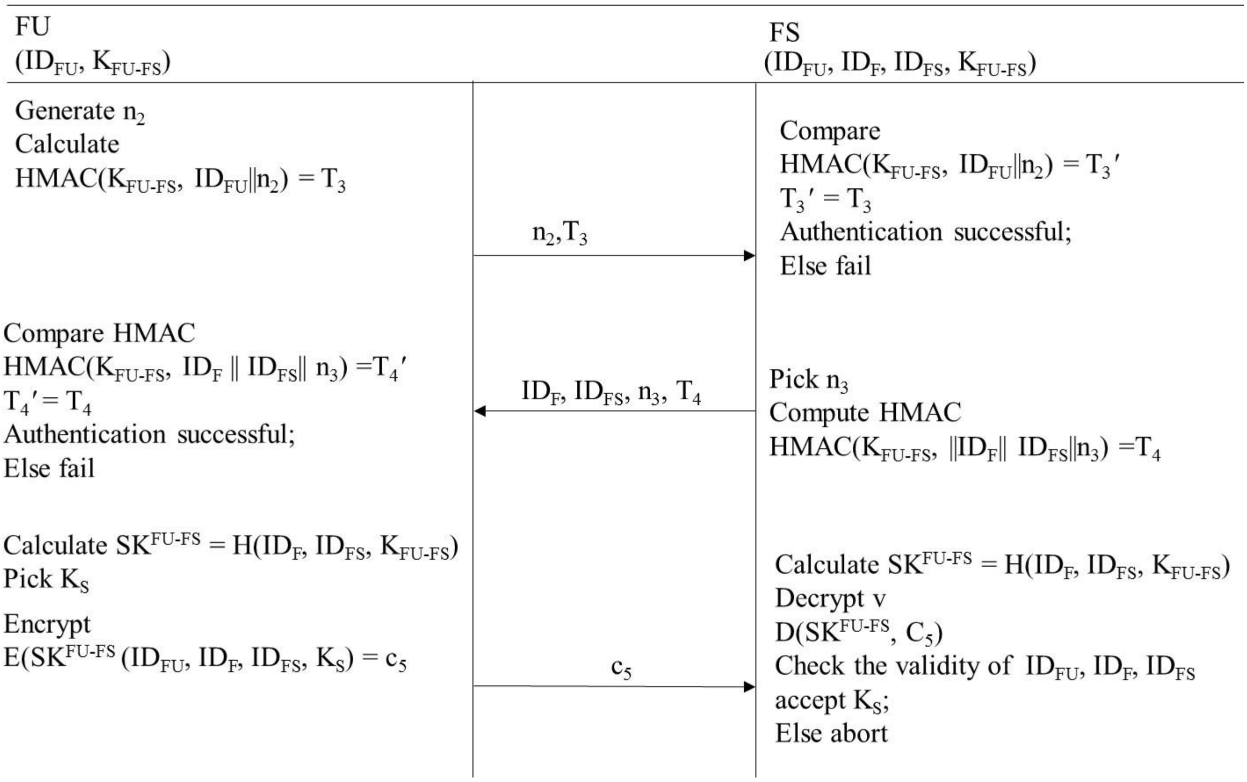
FU
(ID_{FU}, K_{FU-FS})
FS
$(ID_{FU}, ID_F, ID_{FS}, K_{FU-FS})$
Generate n_2
Calculate
$HMAC(K_{FU-FS}, ID_{FU}||n_2) = T_3$
n_2,T_3
Compare
$HMAC(K_{FU-FS}, ID_{FU}||n_2) = T_3'$
$T_3' = T_3$
Authentication successful;
Else fail
Compare HMAC
$HMAC(K_{FU-FS}, ID_F || ID_{FS}|| n_3) =T_4'$
$T_4' = T_4$
Authentication successful;
Else fail
ID_F, ID_{FS}, n_3, T_4
Pick n_3
Compute HMAC
$HMAC(K_{FU-FS}, ||ID_F|| ID_{FS}||n_3) =T_4$
Calculate $SK^{FU-FS} = H(ID_F, ID_{FS}, K_{FU-FS})$
Pick K_S
Encrypt
$E(SK^{FU-FS} (ID_{FU}, ID_F, ID_{FS}, K_S) = c_5$
c_5
Calculate $SK^{FU-FS} = H(ID_F, ID_{FS}, K_{FU-FS})$
Decrypt v
$D(SK^{FU-FS}, C_5)$
Check the validity of ID_{FU}, ID_F, ID_{FS}
accept K_S;
Else abort

Fig. 5 Mutual authentication phase of the proposed improved mutual authentication scheme

VI. SECURITY ANALYSES OF THE PROPOSED IMPROVED SCHEME

Man in the Middle Attack

In man in the middle attack the active attacker attacks in the middle when two entities share secret information with each other. The attacker secretly relays and get the secret parameters and messages. In our proposed improved model when the malicious attacker wish to relay and manipulate the transmitted parameters, the attacker must needs to know the identity of the fog user and must have knowledge about the session key.

Since in our proposed improved model during the authentication phase the attacker cannot know about the secret key used by message authentication code. While during fog user registration the identity is encrypted by using shared symmetric key K_{FU-RA}. Moreover the random nonce and identity of fog user is authenticated by HMAC function. The attacker cannot deduce the HMAC Tag and encrypted identity because the attacker have no knowledge about the shared symmetric key between the fog user, registration authority and fog server Thus in our proposed improved model the man in the middle attack will fails.

Mutual authentication

In the authentication phase the authenticity of fog users and fog servers must be authentic, this process is called mutual authentication where both entities authenticate each other. In our proposed improved model the attacker cannot determine the secret key of the hash message authentication code. The fog server authenticate the fog user by verifying the HMAC on fog server side. Similarly, fog user also authenticate the fog server by verifying the HMAC on fog user side. Moreover, during communication both, entities using random nonce so the attacker cannot rely on messages. Therefore in this step the man in the middle attack also failed.

Anonymity

In our proposed scheme the anonymity of the fog user is protected from the attacker by using random number and HMAC. The hash message authentication code are used to authenticate the actual identity of each fog user. The attacker cannot calculate the actual identity from the HMAC. Therefore, our scheme will protect the anonymity of the fog user.

VII. CONCLUSION

In this paper we have proposed an improved mutual authentication security scheme for fog computing. We have analyzed the Man in the Middle attack in the existing protocol. In a future work we aim to bring the formal security proof for our proposed mutual authentication scheme and compare with the existing security model, in order to achieve better results on computation storage and time consumption.

ACKNOWLEDGMENT

This research was supported by Research Fund, H082 Tier 1/2018, RMC UTHM and Gates IT Solutions Sdn. Bhd

REFERENCES

1. R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing*:* A taxonomy, survey and future directions*”, In Internet of everything*, pp. 103-130. 2018, Springer, Singapore
2. M. Ketel, “Fog-cloud services for iot”, In *Proceedings of the SouthEast Conference*, pp. 262-264. 2017, .2April
3. J. Ni, K. Zhang, X. Lin, and X. S. Shen, “Securing fog computing for internet of things applications Challenges and solutions. *IEEE Communications Surveys & Tutorials*, vol, 20, pp. 601-628, 2017.
4. G. Rahman, and C. Wen, “Fog Computing, Applications, Security and Challenges, Review, *International Journal of Engineering & Technology*, vol. 7, no.3, pp.1615-1621, 2018.
5. N.A. Abubaker, L. Dervishi., and E. Ayday, “Privacy-preserving fog computing paradigm”, In *Communications and Network Security (CNS), 2017 IEEE Conference on* pp. 502-509. IEEE
6. K.A.Fakeeh,“Privacy and security problems in fog computing”. *Common. Appl. Electron.(CAE)*, vol.4, no.6, 2016.
7. M. Azam, H, S. Zedeally, and K. A. Harass, “offloading in fog computing in IoT: Review enabling and research opportunities”. *Future Generation and Computer System*, 2018.
8. G. Rahman, and C. Wen, “Fog Computing, Applications, Security and Challenges, Review, *International Journal of Engineering & Technology*, vol. 7, no.3, pp.1615-1621, New York 2018.
9. F.Y. Okay, and S. Ozdemir. “A fog computing based smart grid model”. In *Networks, Computers and Communications (ISNCC), International Symposium,* 2016, May, pp. 1-6.
10. K.R. Barik, S. K. Gudey, G. G Reddy, M. Pant, H. Dubey, K. Mankodiya, and V. Kumar, “FogGrid: Leveraging Fog Computing for Enhanced Smart Grid Network”, *14TH IEEE India Council International Conference 2017,* Dec 15-17, IIT Roorkee India.
11. R. Brzoza-Woch, M. Konieczny, B. Kwolek, P. Nawrocki, T. Szydlo, and K. Zielinski, “Holistic Approach to Urgent Computing for Flood” Decision Support. In *ICCS* 2015, January, pp. 2387-2396.
12. X. M. Bruin, E. M. Tordera, G. Tashakor, A. Jukan. and G. J. Ren, “Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems”. *IEEE Wireless Communications*, vol.23, no.5, pp. 120-128, 2016.
13. H. F. Atlam, R. J. Walters. And C. B. Wills, “Fog Computing and the Internet of Things: A Review”. *Big Data and Cognitive Computing*, , vol.2, no. 2, PP.10, 2018.
14. A. V .Dastjerdi, and R. Buyya, “Fog computing: Helping the Internet of Things realize its potential. *Computer*, vol.49, no.8, pp. 112-116, 2016.
15. P. More, “Review of implementing fog computing. *International Journal of Research in Engineering and Technology*, vol.4, no. 6, pp. 335-338, 2015.
16. I. Stojmenovic, S. Wen, X. Huang. and H. Luan, “An overview of fog computing and its security issues. Concurrency and Computation”: *Practice and Experience*, vol. 28, no.10, pp.2991-3005, 2016.
17. C. Dsouza, G. J .Ahn, and M .Taguinod, “Policy-driven security management for fog computing: Preliminary framework and a case study”. In *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference* on, 2014, August pp. 16-23. IEEE.
18. M. H. Ibrahim, “Octopus: An Edge-fog Mutual Authentication Scheme”, *IJCN Network Security*, vol. 18, no.6, pp.1089-1101, 2016.
19. A. B. Amor, M. Abid, and A. Meddeb, “A Privacy-Preserving Authentication Scheme in an Edge-Fog Environment. In *Computer Systems and Applications*” 2017, October.
20. Padmapriya, T., Manikanthan, S.V. LTE-A intensified voice service coder using TCP for efficient coding speech. International Journal of Innovative Technology and Exploring Engineering. 2019