Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4, 841-855
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i4.6113

Trustworthy mutual user authentication technique using id in fog computing

K. Sujatha[1*], D. Sasireka[2], M. Ayyadurai[3], G. Indumathi[4], M. Bharathi[5]
[1]SRM Institute of Science and Technology, Ramapuram, India; sujeekevi@gmail.com (K.S.).
[2]CSE, Mepco Schlenk Engineering College, Sivakasi, India; sasirekamtech@gmail.com (D.S.).
[3]CSE, Rajalakshmi Engineering College, India.
[4]CSE, SRM Institute of Science and Technology, Ramapuram, India.
[5]AI, K Ramakrishnan College of Technology, Samayapuram, Trichy, India; Bharathim.ai@krct.ac.in (M.B.).

Abstract: The extension of the cloud that provides services to a massive scale of end users to enhance efficient performance is coined as Fog Computing. Whenever a new paradigm arises, a challenging and vital issue is authentication for security. This paper proposes a new improvised Trustworthy Mutual Authentication Technique (TMAT) to achieve complete mutual authentication between end users and fog nodes using IDs in fog computing. Here, the trusted ID issuer is implemented between end users and fog nodes. They receive their authentication ticket while registering themselves with the Trusted ID Issuer (TID). It involves only simple XOR operations and one-way hash functions. The efficiency of the technique is analyzed by implementing various pseudo-random number generators (PRNG) and is obtained within a limited resource environment. This technique is more secure and efficient by avoiding the storage of a master key and repetitions of session keys. It ensures Dynamic Authentication with the usage of minimal infrastructure architecture.

Keywords: *Cloud computing, Cloud services, Insecure cloud servers, Mutual authentication protocol, Sharing files.*

1. Introduction

The efficiency of cloud computing spread through the high computational power, data processing, management tasks and data storage. Cloud computing plays a major and vital role in data supply. However, in real time applications, the centralized operating mode of cloud computing, has to endure more latency and efficiency drops when the processed data increases. Contemporarily, fog computing introduced to overcome the limitation [1, 2]. Fog computing pivots on the extension of cloud services to the end users while ensuring interaction with the cloud. So that, communication, computation, control and storage operations are performed closer to the edge network by pooling network local resources.

1.1. Architecture of Fog Computing

The objective of fog computing initiated from the outer edges where the data is collected to where it will be stored eventually. In fog computing, the data are collected from the multiple resources leads to get a vital role for fog nodes. They are distributed and deployed across the network.

Fog nodes are used in the process of data analysis. By the characteristics, decentralized local access the dependency on the cloud platform is reduced.

There are six layers in fog computing architecture.

History: Received: 27 January 2025; Revised: 10 March 2025; Accepted: 14 March 2025; Published: 10 April 2025
* Correspondence: sujeekevi@gmail.com

a. Physical and Virtualization Layer: There are two nodes such as physical node and virtual node. They are responsible for data collection. Nodes are equipped with sensors and the collected data sent via gateways.
b. Monitoring Layer: This layer involves in monitoring the time of work, the temperature and the battery life of the device.
c. Pre-processing Layer: collected data are thoroughly checked to derive meaningful data without errors and impurities.
d. Temporary Storage: By the help of virtual storage the data is stored before moving to the cloud.
e. Security Layer: It is responsible for the privacy of the data which enables cryptographic functions over the data.
f. Transport Layer: The scope of this layer is to upload the data to the cloud. The lightweight communication protocols are used to improve the efficiency.

The technical architecture of Fog Computing describes SaaS, PaaS, IaaS and Hardware layers. SaaS describes Application management systems. PaaS having Cisco DSX which acts as a bridge between SaaS and IoT devices. IaaS having Yocto Linux, Cisco IOS and middleware. Hardware Layer consists of some network edges and end user devices.

The diagrammatic representation of Fog computing is given below in Figure 1.

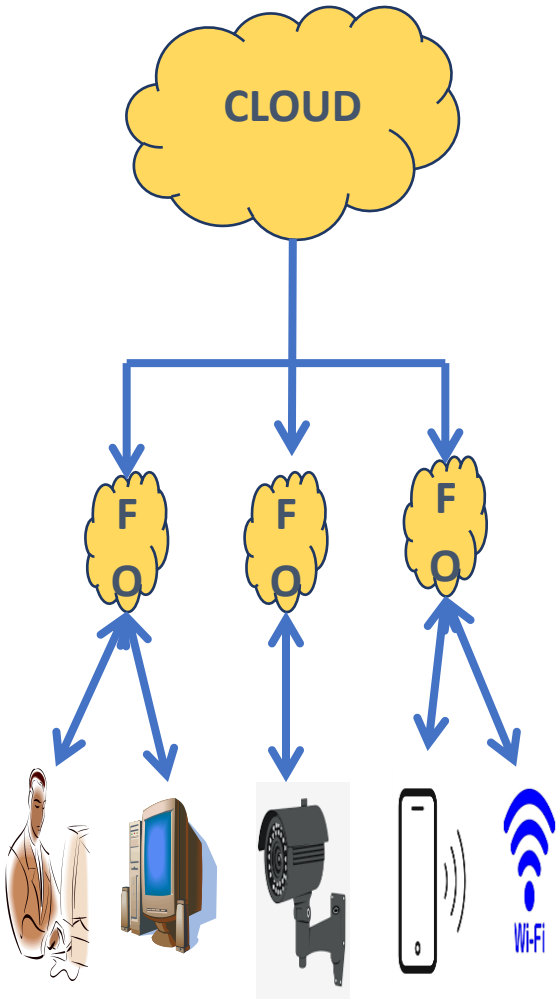

Figure 1.
Architecture of Fog Computing.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

1.2. Fog System Characteristics

The characteristics of the Fog System are given below [1].

- Fog node is positioned at the edge with enormous support by the end users with heterogeneity.
- Instant response allows fog to support the broad range of the industrial applications.
- Fog has its own computing, networking and storage services.
- It runs natively from the edge.
- It provides economic, flexible and portable deployment.
- It is an extremely virtualized system.

The Fog system has its own advantages and disadvantages [3, 4].

- Scalability: Compared to cloud, Fog has small component resources and increases in demand.
- Heterogeneity: Data from a diverse set of devices can be processed.
- Support geographic Distribution: On the basis of geographical location, the data can be distributed either dense and sparse.
- Wireless Access: It supports wireless connectivity and machine to machine communication.
- Real time applications: In switches and IP cameras, it is possible to install a fog system.
- End Devices Mobility: The fog system is mainly used for the mobile and portable devices.
- It provides low latency and location awareness and improves quality of service.

1.3. Problem Statement

The main concern arises in this fog computing, is data security. The decentralized and distributed architecture of fog computing concerns the users about data security. The entry and primary point of any security system is authentication. It verifies the identity of the user.

In several applications, certificate-based authentication plays a vital role to verify the identity of the user. In the fog computing, the end users and fog nodes get their certificates from different authorities, due to heterogeneity.

To achieve mutual authentication both of them have to claim their certificates from intermediate authorities, which are connected to the trusted authority. Thus, defines the process of claiming certificates, obviously ensures mutual authentication while increases latency.

The major concern in certificate-based authentication is revocation management. The users have to update and store recent certificates frequently to validate the signature. It affects the scalability of the system due to the large number of verification request by the central trusted authority.

2. Related Works

The legacy works have been discussed in this section related to the authentication protocol in fog computing.

Ibrahim [5] proposed a mutual authentication scheme, in which the fog users have to mutually authenticate with any fog under the guidance of a cloud service provider. Each and every fog server needs to possess a secret key for each fog user. It is mainly designed for smart card devices. This scheme is still susceptible to masquerading server threat by repeating the master password is unsafe.

Daniel and Mohammad [6] discusses the security challenges of IoT in a resource constrained environment. Public key infrastructure is considered very secure for IoT devices, which is limited by storage, computing, power and memory. Using symmetric key for communication is still not considered as a trustworthy in the system.

Mukherjee, et al. [7] deals with malfunctioning fog nodes attacks to support data security, the process of locating illegitimate access and malicious user inside, challenges of mutual authentication between fog nodes and end users, trustworthy connection between new fog user and fog server.

Hong [8] introduced a programming model for large scale applications on IoT. It is mainly focused on large scale and geo spatially distributed internet applications. It implements mobile fog with the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

high-level programming model. An application can perform can utilize the functions and set of event handlers which is provided by this programming model. Mobile fog is not projected as a structured framework which deals with technical challenges.

Li, et al. [9] introduced Lightweight mutual authentication for IoT. The Public Key Encryption scheme is applied over a couple of light weight devices. To achieve mutual authentication, challenge and response by encryption is used. It implements the scheme on the assumption of the awareness of the public keys and identities of the respected participants. It is being noticed that good performance rather than RSA and ECC.

Alshamrani and Traore [10] discusses a secure mutual authentication for IoT using cumulative Keyed hash chain. They proposed lightweight key exchange mutual authentication which uses temporary updatable ID. Every session is provided with each session key by an anonymous and lightweight key changed scheme. Each and every session is equipped with a session key and a temporary identity by using an anonymous and lightweight methods. A division of devices can be achieved by virtual domain segregation. This system is resisted over insider attacks, outsider attacks and identity theft.

Xiao, et al. [11] proposed a hybrid scheme fine grained search and authorized access with four entities data owner, fog nodes, cloud server and end users in fog computing environment. It ensures keyword privacy, data confidentiality. Attribute based encryption and secure index generation is implemented in the system. It resists over swapping attack and chosen keyword attack. Efficiency is measured over complexity and actual implementation process. Forward secrecy is not considered in this scheme.

Wang, et al. [12] proposed anonymous and secure aggregation scheme with four types of entities such as, system manager, a fog node, terminal devices, and a public cloud server. It addresses anonymity and identity privacy and to achieve those elliptic curves public key cryptography and Castognos-Laguillaumie cryptosystem. In terms of communication and computation the efficiency is improved. Location privacy is not considered in this scheme and adversary model is limited.

Chen, et al. [13] proposed the secure authenticated and key exchange scheme which is specially designed for fog devices. Even though it has some limitations with IoT user input, it achieves various security services.

Alrawais, et al. [14] proposed an attribute-based encryption in fog computing for secure communications. It implements the Ciphertext Policy–Attribute Based Encryption (CP – ABE) algorithm for efficient key management. The digital signature is enabled to obtain authentication, confidentiality and verifiability.

Rahman and Wen [15] proposed a mutual authentication scheme is in fog computing. This scheme provides improved mutual authentication by implementing the Advanced Encryption Standard and the Hashed Message authentication Code (HMAC). The secure mutual authentication between end devices and fog nodes is achieved by Burrows Abdi Needham Logic (BAN logic) [16]. It provides better efficiency with reduced communication and computation overheads.

Wazid, et al. [17] proposed a secure mutual authentication and key management in fog computing. This scheme is enhanced with formal security verification. It uses only lightweight operations such that one-way hash functions and bitwise XOR operations. It achieves reduced overheads on computation and communication cost.

Guo and Zhang [18] proposed an authenticated key agreement scheme in a fog environment without the help of trusted third parties. This scheme does not require trusted cloud server involvement. This scheme is considered as an attractive model and designed with DY adversary model. It protects the system under the Canetti and Krawczyk (CK) adversary model against Ephemeral Secret Leakage (ESL) attacks.

Hamada, et al. [19] proposed a lightweight anonymous mutual authentication in fog computing environment. It achieves various level of security with user anonymity and user untraceability. It needs higher computation costs at fog server and registration authority side.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

Sujatha and Ceronmani Sharmila [20]; Sujatha and Ceronmani [21] and Sujatha and Ceronmani [22] proposed a Trustworthy Mutual Authentication Using ID in cloud and fog computing. It achieves exactly trustworthy protocol by avoiding storage of master keys and repetitions of session keys.

Kalaria, et al. [23] proposed Fog Based Mutual Authentication Scheme using ECC and one way hash functions.

Some papers discuss security challenges, issues and threats [11, 24-26] authorization and trust is discussed in Ko, et al. [27]; Khan and Malluhi [28] and Sujatha and Ceronmani [22] and privacy preserving based techniques are discussed in Koo and Hur [29]; Lu, et al. [30] and Wang, et al. [31].

3. Proposed Methodology

The proposed system TMAT discusses a complete and improvised mutual authentication using the ID between fog users and fog nodes. It is proposed to overcome above addressed security threats and achieve reduced communication and computational overheads.

The general architecture of TMAT has given below in figure 2.

The TMAT involves in the implementation of the Trusted ID Issuer (TID) along with fog users and fog nodes. TID is responsible for fog user registration for mutual authentication [21].

The main scope of the proposed system Trustworthy Mutual Authentication Technique (TMAT) is given as below:

i. To Obtain efficient performance for the limited resource environment.
ii. Dynamic authentication – No need to store any secret keys.
iii. Usage of Minimum Infrastructure.
iv. Different techniques for security are implemented such as a different PRNG for each level of mutual authentication in Fog.
v. Unique keys are used in mutual authentication in Fog.

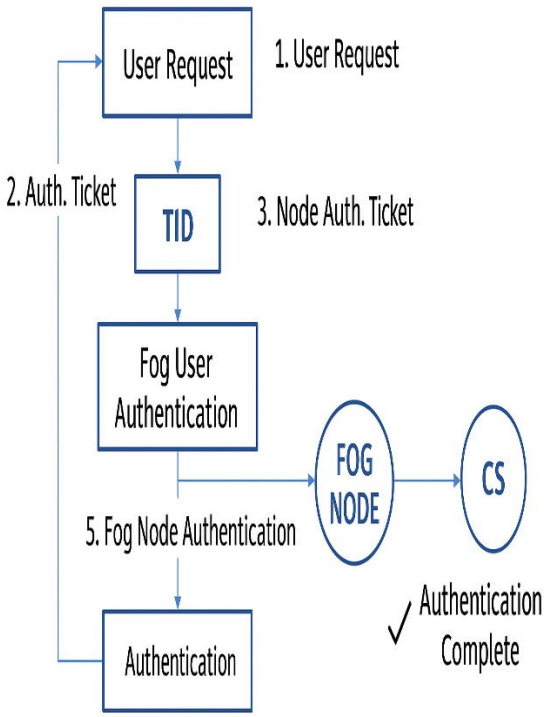

Figure 2.
General Architecture.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

The general framework in fog computing is given below in Figure 3.

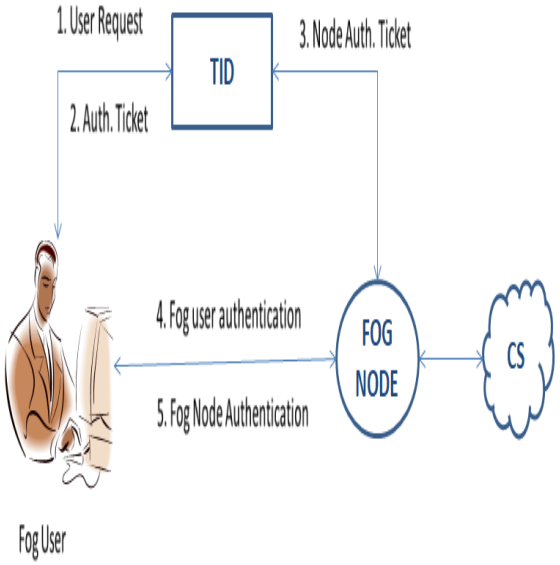

Figure 3.
General Framework.

The notations used in the scheme:
SK_{TID} - Secret key of the TID
FU-ID - Fog/End User ID
FN-ID - Fog Node ID
PR_{FU} - Random number generated by fog user
PR_{FN} - Random number generated from fog node
H(.) - One way hash function
SK_{FU} - Fog User Session Key
SK_{FN} - Fog Node Session Key
XOR - XOR operation
FU_{AT} - Authentication Ticket to Fog User
FN_{AT} - Authentication Ticket to Fog Node
TS - Timestamp

In the proposed system, mutual authentication accomplished by two steps. Fog user Registration is responsible for Registering the end users with TID and gets authentication tickets from TID for mutual authentication. Fog node is also getting notified with the authentication ticket from the TID.

In Mutual Authentication Phase, fog user and fog node mutually authenticated themselves with the help of authentication ticket which is received from the TID.

3.1. Fog User Registration

Figure 4 describes the workflow for the fog user registration phase.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

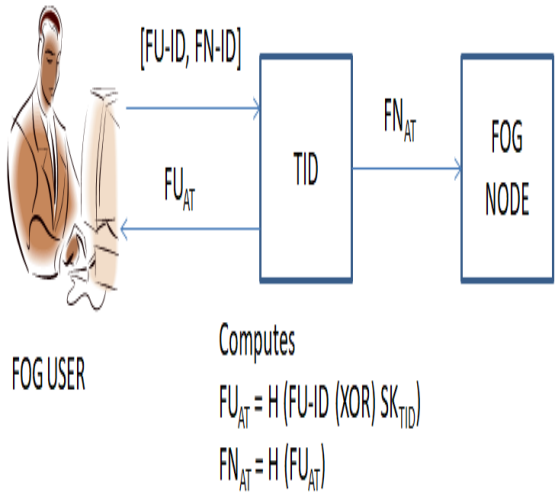

Figure 4.
Fog User Registration.

To start any collaboration with the server the end users have to register themselves with TID.

i. Fog users send their ID and Server's ID which they want to communicate to the TID.

FU ----- (FU-ID, FN-ID) ---→ TID

ii. TID computes hash function of fog user ID XOR with secret key TID. This computed authentication Ticket sent to the fog user.
FU_{AT} = H (FU-ID (XOR) SK_{TID})

iii. TID computes the hash function having the components of fog user authentication ticket and sends it to the respected fog node.
FN_{AT} = H (FU_{AT})

3.2. Mutual Authentication

Mutual Authentication involves, both the fog users and fog nodes authenticate themselves mutually with the help authentication message received from the TID.

3.2.1. Authentication of Fog User

i. In the very initial stage, Fog user generates a session key for the communication by concatenating fog user authentication ticket, Fog user Identity and Timestamp, then it is XORed with Pseudo Random number generated from the Fog user side. Finally, the hash function is performed over the calculated yield.

SK_{FU}=H(FU_{AT}||FU-ID||TS) XOR (PR_{FU})

ii. The fog user takes the pseudo random number for the hash operation which is generated from the fog user side.

A = H (PR_{FU})

iii. The request made from the Fog user side, having session key SK_{FU}, A, Fog user Identity and Timestamp to the fog node.

iv. The timestamp is validated at the fog node side for the freshness.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

v. The fog node computes Pseudo random number of Fog user with the use of received session key value. Then computes 'A1' for the comparison of the received 'A' value. The liaison of Fog user is achieved while matching these values.

3.2.3. Authentication of Fog Node

i. The Fog node generates a session key for fog node authentication. It performs concatenation operation of the Fog user authentication ticket, Identity of Fog Node and Timestamp then it is XORed with the Pseudo random number generated from the fog node side. Finally hash function is applied over the computed value.

$SK_{FN}= H(FU_{AT}||FN\text{-}ID||TS)XOR\ (PR_{FN})$

vi. The fog user takes the pseudo random number for the hash operation which is generated from the fog node side.

$B = H\ (PR_{FN})$

ii. Computed values Session Key SK_{FN} and 'B' to the fog/end user for liaison.
iii. At the fog user side, it verifies the timestamp and computes pseudo random number with the help of receiving session key value if the timestamp is valid.
iv. The Fog user computes 'B1' value from the fog user side and if it matches with the received 'B' value, then the mutual authentication is achieved between fog users and fog node.

Figure 5 represents the flow of work in the mutual authentication phase.

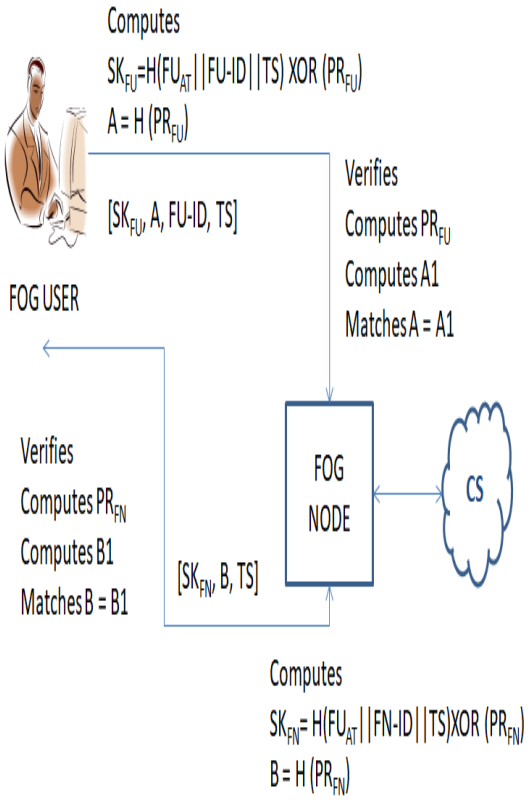

Figure 5.
Flow of Mutual Authentication.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

In mutual authentication phase, two different pseudo random number generators are used in fog user and fog node side. This effects in each level key generation differs and the efficiency improved.

4. Evaluation of Proposed Scheme

The proposed system, the Trustworthy Mutual Authentication Technique is implemented by front end by angular, Node JS on the server side, mongodb for storage and uses the AWS cloud computing environment.

4.1. Efficiency

The efficiency of the mutual authentication technique has been evaluated based on hardness of Pseudo Random Number Generators (PRNG) and session key generation in fog computing. To implement this, random numbers have been evaluated using different random number generators such as Mersenne Twister algorithm on the end user side and Linear Congruential Generator algorithm on the fog node side in the phase of mutual authentication.

To ensure the trustworthiness in the system different random number generators is used on client and server side of authentication phase. This concept involves and effects in the session key generation.

Figure 6 explains the pseudo random numbers of 100 values using the Mersenne Twister PRNG. The 100 values of pseudo random number values are plotted over in 10^9 which takes a high range value.

Figure 7 explains the pseudo random numbers of 100 values using Linear Congruential generator LCG PRNG. The 100 values of pseudo random number values are plotted over in 10^9 which takes a high range value.

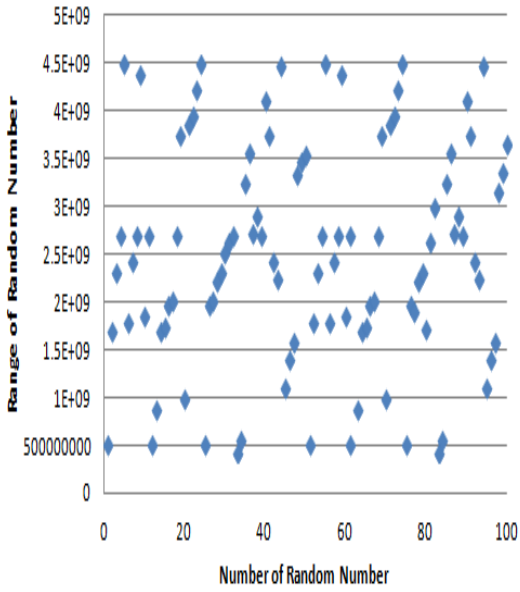

Figure 6.
Pseudo Random Number Values using Mersenne Twister.

Figure 8 describes the generation of session key SK_{FU} with 100 values. These values are based on a SHA-256-bit algorithm which involves Fog User Identity and Fog User Authentication message XORed with Pseudo random number which is generated by Mersenne Twister.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

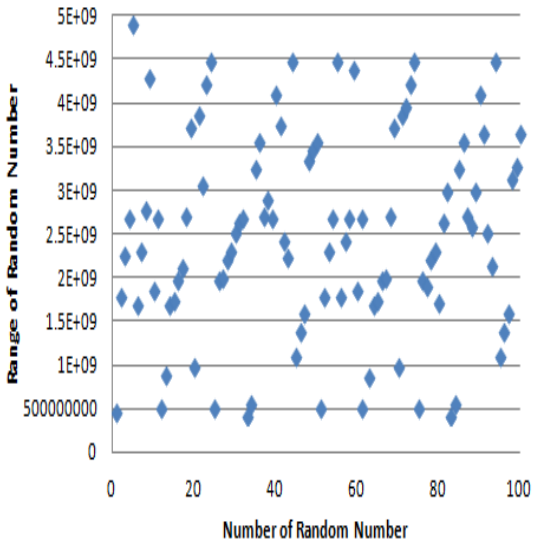

Figure 7.
Pseudo Random Number Values using LCG.

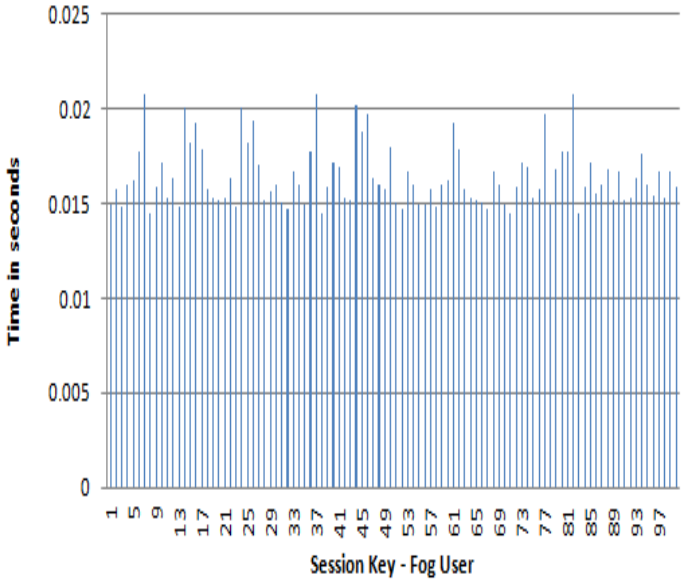

Figure 8.
Session Key Generation at User side.

Figure 9 describes the generation of session key SK_{FN} with 100 values. These values are based on a SHA-256-bit algorithm which involves Fog Node Identity and Authentication message XORed with Pseudo random number which is generated by Linear Congruential Generator.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

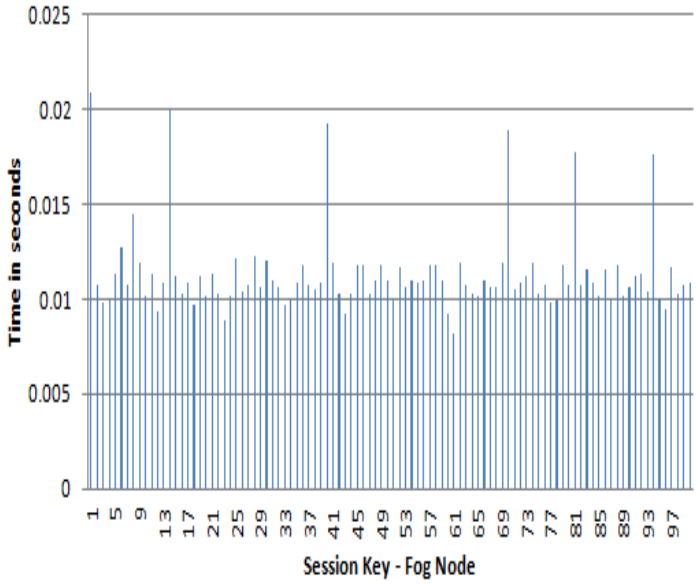

Figure 9.
Session Key Generation at Fog Node side.

In Figure 10, the values have reflected that, the comparison of session key generation over time SK_{FU}, SK_{FN} with each other. In the different stage, different generators are used to generate Pseudo random numbers. Even though different generators are implemented to generate session keys, the time taken to generate each session key is almost near to the each other. No time variance is recorded. The efficiency of the system is improvised via PRNG and ensured trustworthiness.

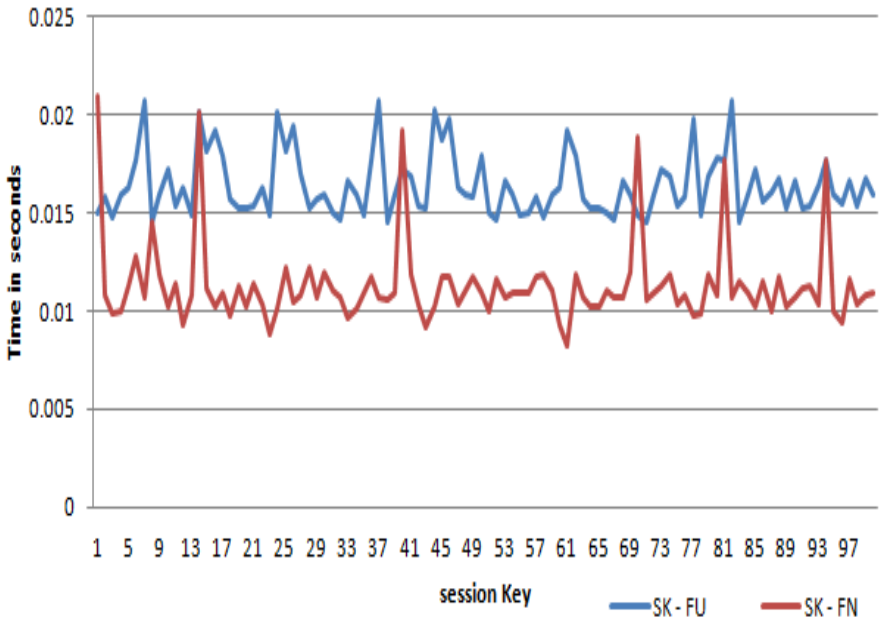

Figure 10.
Session Key Analysis.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

Table 1 depicts the average time taken for each session key set which is calculated respectively for fog user and the fog node.

Table 1.
Session Key Generation Time.

SHA 256 Algorithm	Average Time Taken for Each Set in Seconds
SK-FU	0.016429
SK-FN	0.011318

Table 2 describes the computation complexity of the proposed TMAT for each entity and stage in the mutual authentication process.

The notations used for complexity is given below:

T_V - Time for Verification
T_X - Time for XOR Operation
T_H - Time for Hash Function

Table 2.
Computation Complexity of TMAT.

ENTITY	FU	FS	TID	Total
Registration	-	-	$1T_X+ 2T_H$	$1T_X+ 2T_H$
Authentication	$T_V+2T_X+ 4T_H$	$T_V+2T_X+ 4T_H$	-	$2T_V+ 4T_X+ 8T_H$
Total	$T_V+2T_X+ 4T_H$	$T_V+2T_X+ 4T_H$	$1T_X+ 2T_H$	$2T_V+ 5T_X+ 10T_H$

Table 3 bestows the storage comparison of each entity for the existing methods over TMAT. In the storage compuataion phase, the user identity consists of 32 bits, user password consists of 32 bits, fog server identity consists of 8 bits. The hash function imposed in the algorithm is SHA-256, so that the output size is 160 bits.

The random values generated at fog user side and fog node side by Mersenne twister and Linear Congruential Generator are in size 256 bits. The secret key of the TID measures 256 bits.

Table 3.
Storage computation of TMAT.

ENTITY	Chen, et al. [13]	Sujatha and Ceronmani [21]	TMAT
FU	Elliptic Curve Parameters, R,P,G. User ID. User Password Biometric Features Two Parameters A,D.	User Identity (ID-EU). User Password. User Authentication Ticket (UA). Random Number for EU (RU).	User Identity (FU-ID). User Password User Authentication Ticket (FU_{AT}) Random Number for FU (PR_{FU}).
FS	Elliptic Curve Parameters, R,P,G. One Identity FSID. Two Parameters, FSID, X	Identity of CS (ID-S). Server Authentication Ticket (SA). Random Number for CS (SU).	Identity of FS (FN-ID) Server Authentication Ticket (FN_{AT}). Random Number for FS (PR_{FN}).
TID	Elliptic Curve Parameters, R,P,G. Secret Key S. One Identity for FU One identity for FS One parameter for FU One parameter for FS	Secret Key of IDI (X). One ID for EU (ID-EU). One ID for S (ID-S).	Secret Key of TID (SK_{TID}). One ID for FU (FU-ID). One ID for FS (FN-ID).

Table 4 confers the required storage size in fog user, fog node and TID for the existing methods [13, 21] and the proposed method TMAT in fog computing environment.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

Table 4.
Comparison of Required Storage size (in bits) for each entity.

Entity	Chen, et al. [13]	Sujatha and Ceronmani [21]	TMAT
FU	1840	576	480
FS	1016	520	424
TID	1336	296	296
Total	4192	1392	1200

Figure 11 explains the time taken for a complete mutual authentication process using various algorithms. The algorithm TPUA [21] has taken the authentication time from 4.5 seconds to 5.1 seconds varies over increasing number of request in cloud computing environment. Lightweight Mutual Authentication Protocol LMAP [9] takes time from 1.5 seconds to 1.8 seconds varies over increasing number of requests in fog computing. The proposed method Trustworthy Mutual Authentication Technique TMAT takes time from 1.1 seconds to 1.3 seconds varies over increasing number of requests in fog computing environment.

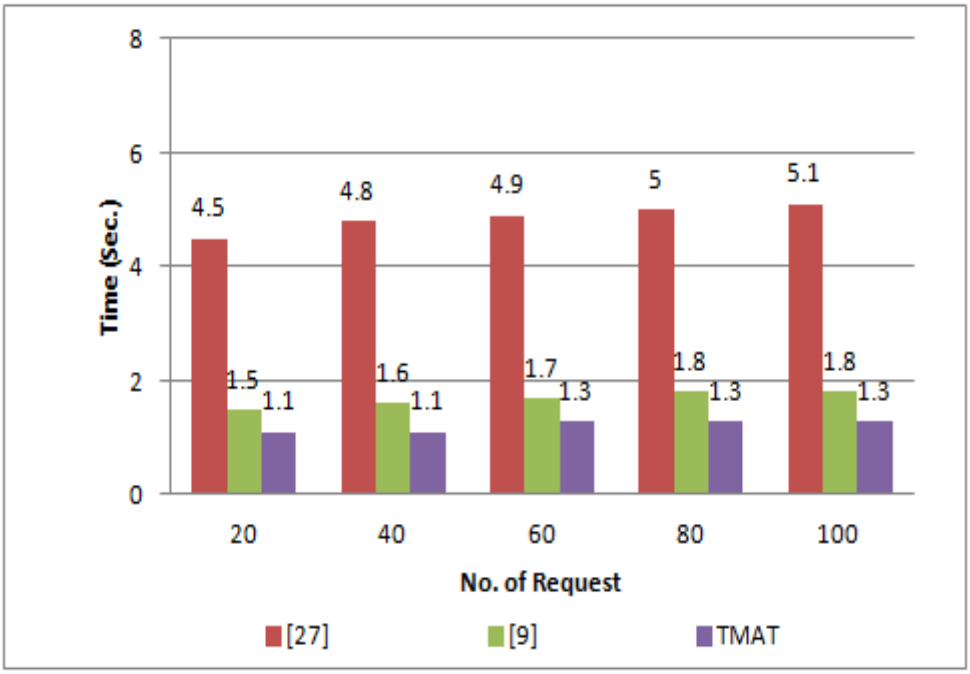

Figure 11.
Comparison over other algorithm.

4.2. Security

The security of the system has been evaluated under criteria such as mutual authentication, stolen verifier attacks, replay attacks, and password-based attacks.

i. Mutual Authentication: The fog user sends the request to any one of the fog nodes that contains a session key (SK_{FU}), a hash function of random number generated for the fog user by using the Mersenne Twister algorithm (A), the identity of the fog user (FU-ID), and a timestamp (TS). On the fog node side, initially it checks the freshness of the request. While it is fresh, it computes a random number from its side. If it matches with the received request, then the fog user has been authorized as a genuine user. The same process has been continued from the fog node to the fog user.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 841-855, 2025
DOI: 10.55214/25768484.v9i4.6113

ii. Stolen verifier attack: This proposed technique does not contain any table for passwords either at the TID or the fog node. At the request time, depending upon the random number, the values have been computed and sent in the request. So, the stolen verifier attack is impossible.
iii. Replay attacks: If an attacker succeeds in getting the request message from the fog user to the fog node (SK_{FU}, A, FU-ID, TS), then the attacker cannot hit the fog node as an authorized person. Because the random number varies every time the request has been made, and timestamp has been added to the request. The initial stage of verification was to check the freshness of the request.
iv. Password based attacks: The proposed scheme does not rely on the password authentication scheme. Either the fog node or TID does not contain any table to store the passwords. So, the proposed scheme TMAT has been protected against the password-based attacks.

5. Conclusion

The proposed scheme achieved the following objectives, minimum infrastructure, unique keys, and different PRNG for each session proven by performing less operations on limited resource devices. The TMAT model is suitable for resisting over various security attacks and exhibits the efficient execution results over various algorithms. The complete mutual authentication and efficiency is achieved using hardness of the PRNG and session key generation. The implementation of different Pseudo Random Number Generators in each session ensures the trustworthiness of the system.

The future work involves the experimentation of multiple high-end servers. The backup of TID is executed in case of any calamity on a single point.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," presented at the Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12), ACM, 2012.
[2] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2014, pp. 1-8.
[3] P. Sareen and K. P., "The fog computing paradigm," *International Journal of Emerging Technology Engineering, Reserch*, vol. 4 pp. 55 – 60, 2016.
[4] K. Sahran and A. Kumar, "Fog in comparison to cloud: A survey," *International Journal of Computer Applications*, vol. 122, no. 3, pp. 10 – 12, 2015.
[5] M. H. Ibrahim, "Octopus: An edge fog mutual authentication scheme," *International Journal of Network Security*, vol. 18, no. 6, pp. 1089 – 1101, 2016.
[6] K. Daniel and H. Mohammad, "Optimization of the public key encryption infrastructure for the internet of things," in *Proceedings of International Conference on Future Networks and Distributed Systems (ICFNDS' 18), Amman, Jordan*, 2018.
[7] M. Mukherjee *et al.*, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017. https://doi.org/10.1109/ACCESS.2017.2757955
[8] K. Hong, "Mobile Fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing (MCC '13), ACM*, 2013, pp. 15–20, doi: https://doi.org/10.1145/2491288.2491291.
[9] N. Li, D. Liu, and S. Nepal, "Lightweight mutual authentication for IoT and its applications," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 193–205, 2017. https://doi.org/10.1109/TSUSC.2017.2760845
[10] M. Alshamrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative approach," *Journal of Information Security and Applications*, vol. 45, pp. 156–175, 2019. https://doi.org/10.1016/j.jisa.2019.02.001

[11] M. Xiao, J. Zhou, X. Liu, S. Guo, and M. Jiang, "A hybrid scheme for fine-grained search and access authorization in fog computing environment," *Sensors*, vol. 17, no. 6, p. 1342, 2017. https://doi.org/10.3390/s17061342

[12] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712–719, 2018. https://doi.org/10.1016/j.future.2017.03.017

[13] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise information system*, vol. 15, no. 9, pp. 1200–1215, 2021. https://doi.org/10.1080/17517575.2020.1712746

[14] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, and X. Cheng, "An attribute-based encryption scheme to secure fog communications," *IEEE Access*, vol. 5, pp. 9131–9138, 2017. https://doi.org/10.1109/ACCESS.2017.2706260

[15] G. Rahman and C. C. Wen, "Mutual authentication security scheme in fog computing," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, pp. 443–451, 2019. https://doi.org/10.14569/IJACSA.2019.0101159

[16] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990. https://doi.org/10.1145/77648.77649

[17] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019. https://doi.org/10.1016/j.future.2018.09.007

[18] Y. Guo and Z. Zhang, "Authenticated key agreement scheme without trusted parties," *IEEE Systems Journal*, vol. 15, no. 1, pp. 708–717, 2021. https://doi.org/10.1109/JSYST.2020.2967075

[19] M. Hamada, S. A. Salem, and F. M. Salem, "LAMAS: Lightweight anonymous mutual authentication scheme for securing fog computing environments," *Ain Shams Engineering Journal*, vol. 13, p. 101752, 2022. https://doi.org/10.1016/j.asej.2021.10.023

[20] K. Sujatha and V. Ceronmani Sharmila, "Efficient mutual user authentication protocol to share files using ID in cloud storage," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 20, pp. 3362–3373, 2020.

[21] K. Sujatha and V. S. Ceronmani, "Enhanced TPUA mechanism with data protection using ID in cloud computing," *Journal of Engineering and Applied Sciences*, vol. 14, no. 4, pp. 1189–1197, 2019.

[22] K. Sujatha and V. S. Ceronmani, "Enhanced mutual authentication technique using ID (MATID) in fog computing," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 1728–1733, 2019. https://doi.org/10.35940/ijrte.B2817.078219

[23] R. Kalaria, A. S. M. Kayes, W. Rahayu, and E. Pardede, "A secure mutual authentication approach to fog computing environment," *Computers & Security*, vol. 111, p. 102490, 2021. https://doi.org/10.1016/j.cose.2021.102490

[24] S. Sarkar, S. Chatterjee, and S. Mishra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, pp. 1–14, 2015. https://doi.org/10.1109/TCC.2015.2485206

[25] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey. In R. Bestak, J. Platos, & P. Ruiz (Eds.)," in *Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015)*, *Springer*, 2015, pp. 685–695, doi: https://doi.org/10.1007/978-3-319-21837-3_60.

[26] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017. https://doi.org/10.1109/MIC.2017.37

[27] R. K. L. Ko *et al.*, "TrustCloud: A framework for accountability and trust in cloud computing," presented at the 2011 IEEE World Congress on Services, IEEE, 2011.

[28] K. M. Khan and Q. Malluhi, "Establishing trust in cloud computing," *IT Professional*, vol. 12, no. 5, pp. 20–27, 2010. https://doi.org/10.1109/MITP.2010.122

[29] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 421–432, 2018. https://doi.org/10.1016/j.future.2017.02.010

[30] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017. https://doi.org/10.1109/ACCESS.2017.2671342

[31] T. Wang *et al.*, "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, 2017. https://doi.org/10.1109/ACCESS.2017.2688382