Hindawi
Complexity
Volume 2021, Article ID 4667586, 9 pages
https://doi.org/10.1155/2021/4667586

Research Article

LSKE: Lightweight Secure Key Exchange Scheme in Fog Federation

Yashar Salami and Vahid Khajehvand

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Correspondence should be addressed to Vahid Khajehvand; vahidkhajehvand@gmail.com

Received 16 July 2021; Revised 13 August 2021; Accepted 29 September 2021; Published 25 October 2021

Academic Editor: Bo Xiao

The fog computing architecture allows data exchange with the vehicle network, sensor networks, etc. However, before exchanging data, the nodes need to know each other and key exchange. Yashar et al. recently proposed a secure key exchange scheme for the fog federation. However, their proposed scheme has a high computational overhead and is not suitable for fog federation. Therefore, we have proposed a lightweight, secure key exchange scheme for the fog federation to reduce computational overhead. To prove the lightweight, we have compared the proposed scheme with the Yashar design in terms of computing, and communication cost AVISPA Tool was used for the formal analysis of the proposed scheme. Then, we simulated the proposed scheme with the NS3 tool and compared it with Throughput, packet loss, Packet Delivery, and end-to-end delay with Yashar et al. scheme. The results show that the proposed design reduced 3.2457 ms of computational overhead and 1,024 transmitted data bits.

1. Introduction

The spread of distributed systems such as the cloud [1] has made it possible for users to access their data from anywhere and share or process their data. Furthermore, with the expansion of various branches of computer science and the relationship between these sciences, the development of distribution systems has accelerated. Today, the Internet of Things is connected to the fog layer and can generate thousands of data at any time that are sent to the fog layer for processing [2].

However, the fog layer needs to provide the necessary security for data processing between its nodes before processing the data. One of the most important challenges in maintaining security is how to exchange the key from the other side so that it is resistant to known attacks in the fog layer.

Novel remote user authentication and key agreement scheme for mobile client-server environment scheme in 2013 were proposed by Sun et al. [3]. This scheme was not secure and could not support the fog federation. In 2015, Li et al. [4] proposed smart card-based mutual authentication schemes in cloud computing. This scheme was not secure and could not support the fog federation. Security and privacy preservation scheme of face identification and resolution framework using fog computing in the Internet of things was presented by Hu et al. [5] in 2017; this scheme did not support fog federation and key exchange. The scheme proposed by Jia et al. [6], Wazid et al. [7], Chen et al. [8], Zheng and Chang [9], and Chen et al. [10] in 2019, 2020, and 2021 were all safe and supportive of mutual authentication and key exchange. However, they are not suitable for fog federation environments. Yashar et al. [11] proposed a secure key exchange scheme in the fog federation in 2021. This scheme supported mutual authentication and key exchange; however, this scheme is not lightweight. Table 1 shows a comparison of related work. Providing a secure and lightweight key exchange scheme in a fog federation environment is a challenge in this area.

1.1. Paper Contribution

(i) In this paper, we propose a secure lightweight key exchange scheme for the fog federation

(ii) For formal security analysis, the proposed scheme uses the AVISPA tool

(iii) The proposed scheme is compared with Yashar et al. regarding computing cost, communication cost, and security requirement

Table 1: Comparison of related work.

Related work	Fog federation	Secure	Mutual authentication	Key exchange	Lightweight
Sun et al. [3]	x	x	✓	✓	x
Li et al. [4]	x	x	✓	✓	x
Hu et al. [5]	x	✓	✓	x	x
Jia et al. [6]	x	✓	✓	✓	x
Wazid et al. [7]	X	✓	✓	✓	x
Chen et al. [8]	x	✓	✓	✓	x
Zheng et al. [9]	x	✓	✓	✓	x
Chen et al. [10]	x	✓	✓	✓	x
Yashar et al. [11]	✓	✓	✓	✓	x
Proposed scheme	✓	✓	✓	✓	✓

✓, the scheme is supported; X, the scheme is not supported.

(iv) The proposed scheme and Yashar et al. scheme are simulated with the NS3 tool and examined in terms of throughput, packet loss, packet delivery, and end-to-end delay criteria

1.2. Paper Organization. The rest of the paper is organized as follows. Section 2 reviews the Yashar et al. and network model. The proposed scheme has been presented in Section 3. Section 4 provides a security analysis of the proposed scheme with the AVISPA tool. Section 5 presents the performance analysis and security requirements. Section 6 compares the proposed scheme's simulation results with Yashar et al. Finally, conclusions have been presented in Section 7.

2. The Background

This section provides the ECC and network model and problem statement and scheme of Yashar et al.

2.1. Review of ECC. The elliptic curve cryptography (ECC) is a public key encryption method, which has been designed based on an algebraic structure of elliptic curves on the finite fields. The curves of the elliptic equations are in the form of $y + axy + by = x^3 + cx^2 + dx + e$. In this equation, $\mathbb{R} = \{a, b, c, d, e\}$. These are real numbers that must satisfy simple conditions. In these curves, a point is zero or a point in infinity. For more information, you can refer to [12].

2.2. Network Model and Problem Statement. The network model presented in Figure 1 shows that cloud servers are at the top tier and can communicate with each other. In the network model, there is a middle layer of fog nodes. In this layer, there is a central fog whose main task is to manage other fog nodes. The middle layer can be connected to the top layer and the low layer. The purpose of developing the haze layer was to reduce latency for bottom layer processing. At the low layer are IOV, IOS, IOE, and M2M devices. If these devices require high processing, they send their data to the fog layer for processing. In the fog layer, the central node needs to be aware of the identity of the nodes so that they can exchange data with each other. Furthermore, because the central node is being processed and managed, a secure, lightweight key exchange scheme is needed that can withstand known attacks.

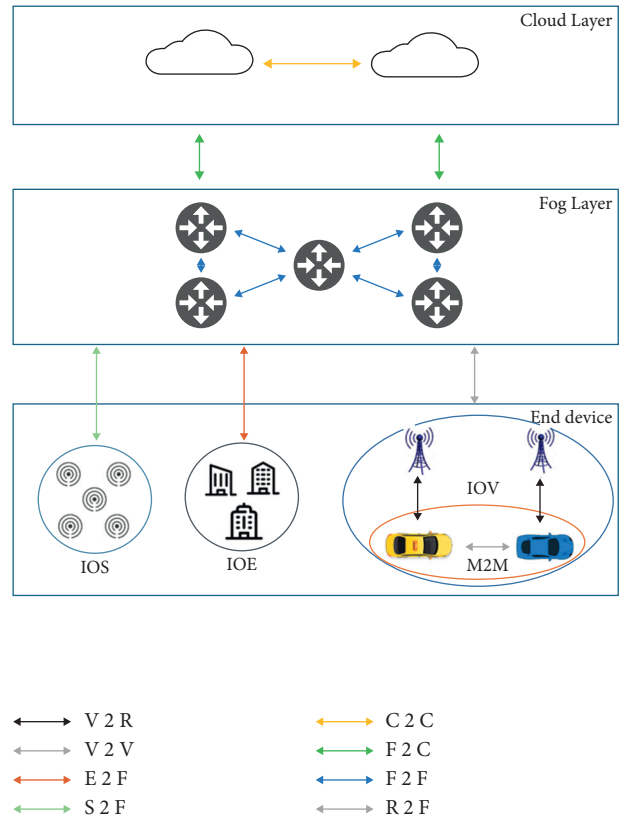

Figure 1: Network model [11].

2.3. Notations. The list of notations used in this paper is shown in Table 2.

2.4. Review of Yashar Et Al. The key exchange request steps are as follows.

Step 1: Bob generates an RB message to request the key exchange and transmits it to Alice. RB is calculated as follows:

Table 2: Notations used for the proposed work.

No.	Notations	Description
1	IDA	Identity of Alice (fog center)
2	IDB	Identity of Bob (fog node)
3	NCa	Nonce of Alice (fog center)
4	NCB	Nonce of Bob (fog node)
5	Cha	Challenging of Alice (fog center)
6	CHb	Challenging of Bob (fog node)
7	Kas	The public key of Alice (fog center)
8	Kbs	The public key of Bob (fog node)
9	TA	Timestamp of Alice (fog center)
10	TB	Timestamp of Bob (fog node)
11	ΔT	Expiration time
12	h	Hash
13	$\|\|$	Concatenation

Bi = IDB, NCb, NHa, Kbs, TB,
HBi = h(Bi),
RB = HBi‖Bi.

Step 2: Alice separates the file contents upon receiving RB and then hashes Bi and compares the Bi` hash with HBi. Then, if the hash of Bi and HBi are the same, she checks the packet timestamp with the predetermined ΔT. If the timestamp of the received packet is smaller than ΔT, the packet is valid. Next, Alice generates RBi, and Ai sends it to Bob. RB and Ai are calculated as follows:

Bi‘ = h(Bi),
Check if Bi‘ = HBi,
Check if TB ≤ ΔT,
HRBi = h(IDB, NCb, CHb, Kbs, TA,),
RBi = (IDB, NCb, CHb, Kbs, TA, HRBi)Kbs.

Key generation by Alice is as follows:

(i) To generate a Galois field, Alice selects a large prime number and calls it p. The field Zp might have $p-1$ generators.
(ii) Alice selects one of the generators of Zp and calls it G.
(iii) Alice selects an arbitrary number and calls it a, and keeps it secrete. Then, the selected numbers are substituted in equation (1) to generate A:

$$\begin{aligned} A &= G^a \bmod P, \\ \text{HAi} &= h\,(\text{IDA, NCa, CHa, Kas, } (P, G, A), \text{ TA}), \\ \text{Ai} &= (\text{IDA, NCa, CHa, Kas, } (P, G, A), \text{TA, HAi})\text{Kbs}. \end{aligned} \tag{1}$$

Step 3: Upon receiving Ai and RBi, Bob separates the contents of RBi with his public key and hashes the contents except for HRBi and compares RBi` with HRBi. Then, continue the calculation as follows:

RBi decrypt by key Kbs,
RBi‘ = h(IDB, NCb, CHb, Kbs, TA),
Check if RBi‘ = HRBi,
Check if TA ≤ ΔT.

Key generation by Bob are as follows:

Bob uses equation (2) to generate B. Next, to obtain the shared key with Alice empowers A by b in the modulus of P, according to equation (3), the result would be the shared key agreed upon by Alice. In the next step, it calculate RAi from the following relation and sends it to Alice:

$$B = G^b \bmod P, \tag{2}$$

$$\begin{aligned} K &= A^b \bmod P = (G^a)^b \bmod P \\ &= G^{a.b} \bmod p, \text{ Ai decrypt by key Kbs, Ai‘} \\ &= h\left(\text{IDA, NCa, CHa, Kas, } G^{\text{Na}}, \text{ TA}\right), \text{Check if Ai‘} \\ &= \text{HAi, Check if TA} \le \Delta T, \text{HAi} \\ &= h\,(\text{IDA, NCa, CHa, Kas, } B, \text{ TB}), \text{RAi} \\ &= (\text{IDA, NCa, CHa, Kas, } B, \text{HAi, TB})\text{Kas}. \end{aligned} \tag{3}$$

Step 4: Alice opens RAi with her public key, hashes the packet contents except for HAi, and compares RAi' with HAi. Then, Alice empowers B by a in the modulus of P according to equation (4) to obtain the shared key. Figure 2 shows the key exchange scheme of Yashar et al. The shared key calculation is as follows:

$$\begin{aligned} K = B^a \bmod P = \left(G^b\right)^a \bmod P = G^{a.b} \bmod p \\ \text{RAi decrypt by key Kas} \\ \text{RAi‘} = h\,(\text{IDA, NCa, CHa, Kas, } B, \text{ TB}) \\ \text{Check if } RAi\text{‘} = RAi \\ \text{Check if TB} \le \Delta T. \end{aligned} \tag{4}$$

3. Proposed Scheme

This section presents the proposed scheme. The key exchange request steps are as follows:

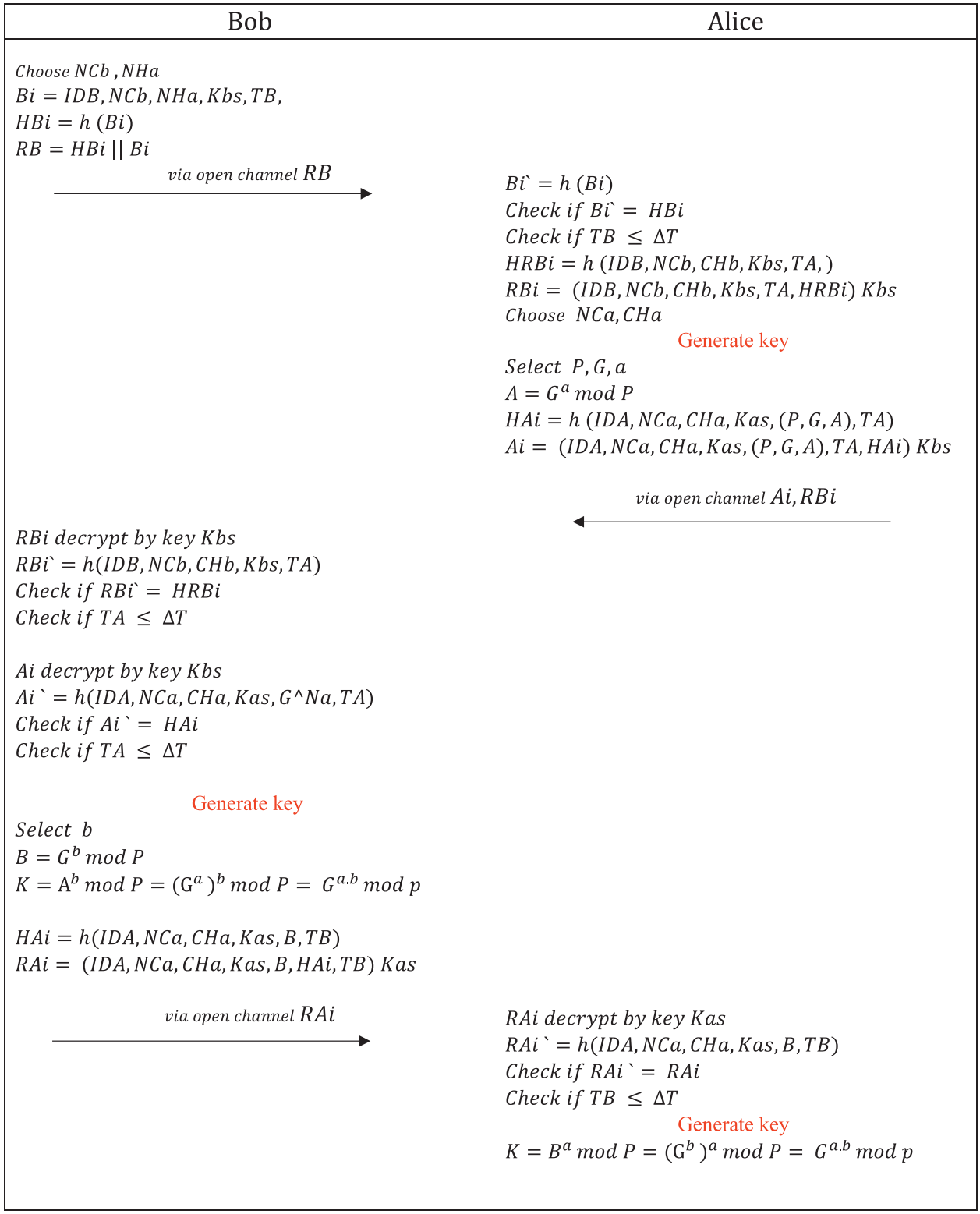

Figure 2: The key exchange scheme of Yashar et al.

Step 1: calculates the fog node of the equation $A1 = (\mathrm{IDA}, \mathrm{IDB}, \mathrm{TA}, \mathrm{Kas})$ and send it to the fog center.

Step 2: the Fog center first checks the time stamp with the expiration time; if the timestamp is shorter than the expiration time, it stores the Kas key.

In the next step, he chooses the numbers $a, b, p, R1, R2$, and NB and calculates them through equations $H1, B1, H2, B2, H3$, and PB, through equation (5). Finally, it sends B3 to the fog node:

$$\begin{aligned} H1 &= h(\mathrm{IDA}, \mathrm{IDB}, \mathrm{Kbs}),\\ B1 &= (\mathrm{IDA}, \mathrm{IDB}, \mathrm{Kbs}, H1),\\ H2 &= h(a, b, p, R1, R2),\\ B2 &= (a, b, p, R1, R2, H2),\\ \mathrm{PB} &= \mathrm{NB} * G(R1, R2),\\ H3 &= h(\mathrm{PB}),\\ B3 &= (\mathrm{PB}, \mathrm{TB}, H3, B1, B2)\mathrm{Kas}. \end{aligned} \tag{5}$$

Step 3: the Fog node first checks the time stamp with the expiration time; if the timestamp is shorter than the expiration time, it stores the Kas key. It then hashes B1, B2, and B3 and compares it to H1, H2, and H3 to ensure that the message is not tampered with. It then saves a, b, p, R1, R2, and PB. The fog node selects a random number in the next step, places it in equation (6), and obtains PA.

$$\mathrm{PA} = \mathrm{NA} * G(R1, R2). \tag{6}$$

After the following calculations, it sends A2 and H4 to the fog center:

$$\begin{aligned} H4 &= h(\mathrm{PA}),\\ A2 &= (\mathrm{PA}, \mathrm{TA}, H4)\mathrm{Kbs}. \end{aligned}$$

Fog node through equation (7) calculates the common key:

$$K = \mathrm{NA} * \mathrm{PB}. \tag{7}$$

Step 4: the Fog center first checks the time stamp with the expiration time; if the timestamp is shorter than the expiration time, it first hashes A2 and compares it to H4. Then, it checks the time stamp with the expiration time; if the timestamp is shorter than the expiration time, the fog center calculates the common key through equation (8). Figure 3 shows the proposed scheme.

$$K = \mathrm{NB} * \mathrm{PA}. \tag{8}$$

4. Security Analysis

This section presents the simulation results of the proposed scheme with the AVISPA tool.

The AVISPA tool is a formal simulation to assess whether a secure or insecure protocol [13]. AVISPA uses an HLPSL language to describe and display the security specifications of protocols. HLPSL is a role-oriented language in which each entity plays an independent role during the protocol implementation [14]. In HLPSL, a legal role is conceived for the attacker, modeled by Dolew-yao [15]. AVISPA has four built-in tools OFMC (On-the-Fly Model-Checker) [16], CL-AtSe (Constraint Logic-based Attack Searcher) [17], SATMC (SAT-based Model-Checker) [18], and TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) [19] that are used for security analysis. After parsing, the output results indicate whether the protocol is secure or insecure.

4.1. Analysis of Simulation Results. Figures 4 and 5 show the simulation results showing the proposed design with security tools OFMC and CL-AtSe. The simulation results in the OFMC show that the total number of nodes visited for the proposed scheme was 17 and with a depth of 4 in 0.14 seconds. The simulation results in the CL-AtSe show that the total number of analyzed and reachable for the proposed scheme was four states in translation time was 0.04 seconds. Furthermore, the security analysis results with tools OFMC and CL-AtSe show that the proposed scheme is secure.

5. Performance Analysis

In this section, the performance analysis of the proposed scheme and security requirements are compared with Yashar et al. The following symbols are defined to evaluate the computing cost of the proposed scheme. Th is the execution number of a hash operation. Pm is the execution number of Point Multiplication. Pe is the execution number of public key encryption. Pd is the execution number of public key decryption. Se is the execution number of symmetric key encryption. Sd is the execution number of symmetric key decryption. The execution time to perform the computation is as follows. Th ≈ **0.0023**, Pm ≈ **2.226**, Pe ≈ **3.8500**, Pd ≈ **3.8500**, Se ≈ **0.0046**, and Sd ≈ **0.0046.** The proposed scheme uses 1024 bit RSA.

5.1. Computation Cost. Table 3 shows a comparison of the computing cost of the proposed scheme and the Yashar et al. scheme. Our observations show that the Yashar scheme consists of 7 Th, 3Pe, and 3Pd; the total cost is 23.1161 ms. On the contrary, our proposed scheme consists of 8Th, 2Pm, 2Pe, and 2Pd; the total cost is 19.8704 seconds. Thus, compared to Yashar et al., the proposed scheme reduced the calculation by 3.2457.

5.2. Communication Cost. Table 4 shows a comparison of the communication cost of the proposed scheme, and the Yashar et al. scheme has a communication cost of 3, and the total number of bits used is 3072. In our proposal, the communication cost is three, and the total number of bits used is 2048. Thus, we reduced 1,024 bits sent over the scheme of Yashar et al.

5.3. Security Requirements' Comparison. Our observations show that the proposed scheme is resistant to defined attacks. However, our proposal also cannot support device

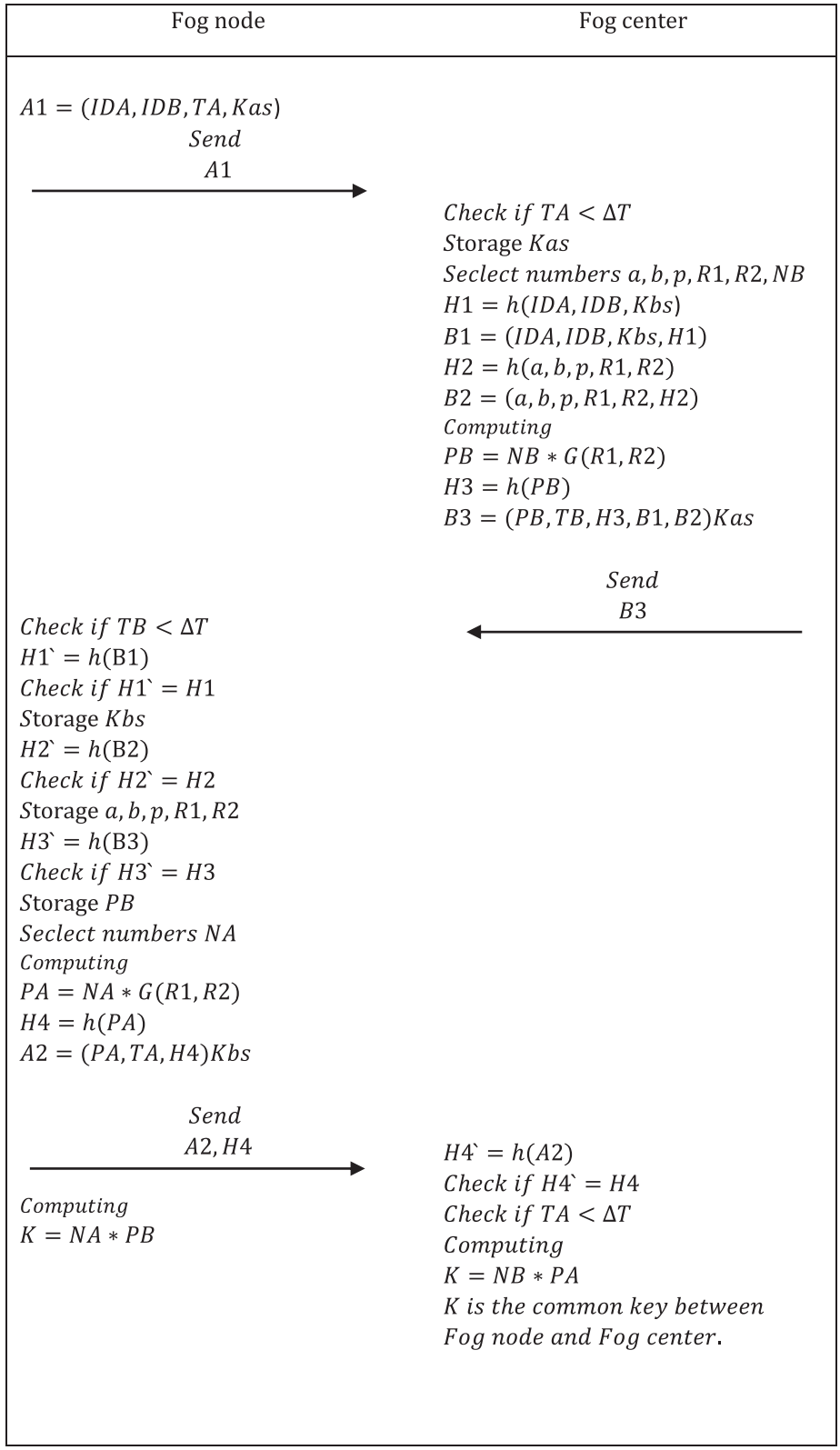

Figure 3: The proposed scheme.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/hlpslGenFile.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  Parse Time: 0.00s
  Search Time: 0.14s
  Visited Nodes: 17 nodes
  depth: 4 plies
```

Figure 4: Simulation results of the proposed scheme under OFMC.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  home/span/span/testsuite/results/hlpslGenFile.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed : 4 states
  Reachable : 4 states
  Translation: 0.04 seconds
  Computation: 0.00 seconds
```

Figure 5: Simulation results of the proposed scheme under CL-ATSe.

Table 3: Comparison of computation cost.

No.	Schemes	Hash function	Point multiplication (Pm)	Public key encryption	Public key decryption	Symmetric key encryption (Se)	Symmetric key decryption	Total cost	TC (ms)
1	Yashar et al.	7Th	0	3Pe	3Pd	0	0Sd	7Th + 3Pe + 3Pd	23.1161
2	Proposed	8Th	2	2Pe	2Pd	0	0Sd	8Th + 4Pm + 2Pe + 2Pd	19.8704

anonymity and session key agreement. Table 5 shows the security requirements' comparison of the proposed scheme with the Yashar et al. scheme. Note: AF1: replay attack; AF2: man-in-the-middle attack; AF3: insider attack; AF4: impersonation attack; AF5: brute force attack; AF6: offline password guessing attack; AF7: device anonymity; AF8: mutual authentication; AF9: session key agreement; AF10: key exchange, AF11: fog federation; AF12: OFMC; AF13: CL-ATSE; ✓: the scheme is supported; X: the scheme is not supported.

6. Simulation and Result

In this section, a simulation of the proposed design with the Yashar design is provided. In addition, simulation by network simulation tool (NS 3 2.29 simulator) on the Ubuntu-20.04.1 platform is provided. The hardware environment for carrying out NS3 simulation [20] was on Dell Inspiron 5110 machine with Intel Core i5 2410 M/ 2.30 GHz processor having 4 GB RAM and 1 TB HDD (Hard Disk Drive).

Table 4: Comparison of communication cost and the number of bits.

No.	Schemes	No. of messages	Total cost (in bits)
1	Yashar et al.	3	3072
2	Proposed	3	2048

Table 5: Comparison of security requirements.

Security requirements	Scheme	
	Yashar et al.	Proposed
AF1	✓	✓
AF2	✓	✓
AF3	✓	✓
AF4	✓	✓
AF5	✓	✓
AF6	✓	✓
AF7	X	x
AF8	✓	✓
AF9	X	x
AF10	✓	✓
AF11	✓	✓
AF12	✓	✓
AF13	✓	✓

Table 6: Simulation parameters.

Parameters	Description
Platform	Ubuntu-20.04.1
Hardware platform	Dell 5110, Intel Core i5, 4 GB RAM, 1 TB HDD
Tool used	NS 3 2.29
Number of fog node	20
Number of fog center	10
Mobility of fog node	0
Mobility of fog center	0
Simulation environment area	300 ∗ 1500 M
Loss model	Friis loss
Transmit power	7.5 dB
Routing protocol	OLSR
Medium access control type	IEEE 802.11
Wireless protocol	802.11 p
Communication range of fog node to fog center	100 M
Simulation time	1800 seconds

6.1. Simulation Environment and Settings. The various parameters used in the NS3 simulations are provided in Table 6. The simulation time of the proposed scheme was 1800 seconds. The number of fog centers is ten, and the fog node is 20. Other parameters are as follows: the mobility of the fog centers and fog node is 0 m/s, loss model is Friis loss, transmit power is 7.5-dBm, medium access control type IEEE 802.11, wireless protocol 802.11 *p*, routing protocol: OLSR, and Simulation Environment Area is 300 ∗ 1500M.

6.2. Simulation Results. The simulation results show that the proposed scheme performs better in terms of throughput than the Yashar et al. scheme. Figure 6 shows a comparison

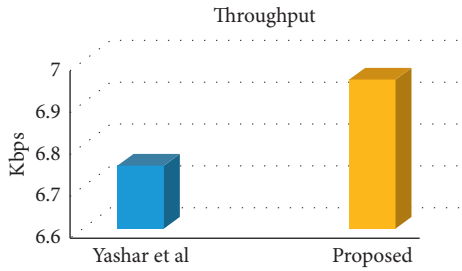

Figure 6: Comparison of throughput.

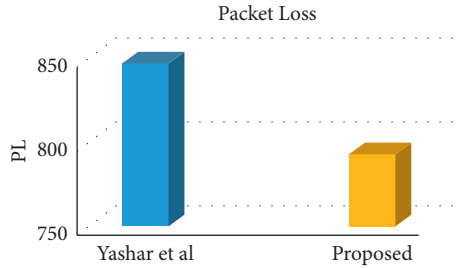

Figure 7: Comparison of packet loss.

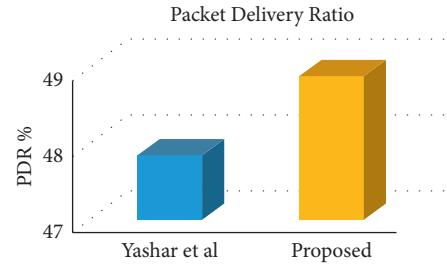

Figure 8: Comparison of packet delivery.

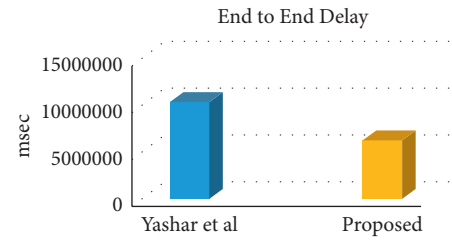

Figure 9: Comparison of end-to-end delay.

of the proposed scheme and Yashar et al. scheme in terms of throughput. The proposed scheme has a much better performance in packet loss than Yashar et al. Figure 7 compares the proposed scheme and Yashar et al. scheme in packet loss. In terms of packet delivery rate, the proposed scheme has shown better performance than Yashar et al. Figure 8 compares the proposed scheme and Yashar et al. scheme in terms of the packet delivery rate. Finally, in terms of end-to-end delay, the performance of the proposed design is better than that of Yashar et al. Figure 9 shows a comparison of the proposed scheme and the Yashar et al. scheme in terms of end-to-end delay.

7. Conclusion

The secure key exchange in fog federation environments is a major challenge. This paper presents a lightweight, secure key exchange scheme based on ECC for fog federation environments. The results of the AVISPA tool show that the proposed scheme is safe, and the proposed scheme is compared with Yashar et al. Comparison results show that the proposed scheme has a lower computational and byte cost. The proposed scheme is then simulated with the NS3 tool. The simulation results show that the proposed scheme performs better in terms of throughput, packet loss, packet delivery, and end-to-end delay than Yashar et al. In future work, our goal is to provide a three-way key exchange scheme in fog federation.

Data Availability

Data used to support this novel scheme are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. Zhang, Y. Li, and L. Lu, "Privacy-preserving attribute-based keyword search with traceability and revocation for cloud-assisted IoT," *Security and Communication Networks*, vol. 2021, Article ID 9929663, 13 pages, 2021.

[2] B. Zheng, Z. Mei, L. Hou, and S. Qiu, "Application of internet of things and edge computing technology in sports tourism services," *Security and Communication Networks*, vol. 2021, Article ID 9980375, 10 pages, 2021.

[3] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel remote user authentication and key agreement scheme for mobile client-server environment," *Applied Mathematics & Information Sciences*, vol. 7, no. 4, pp. 1365–1374, 2013.

[4] H. Li, F. Li, C. Song, and Y. Yan, "Towards smart card based mutual authentication schemes in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 7, pp. 2719–2735, 2015.

[5] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1143–1155, 2017.

[6] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[7] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019.

[8] 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 Systems*, vol. 75, pp. 1–16, 2020.

[9] Y. Zheng and C.-H. Chang, "Secure mutual authentication and key-exchange protocol between PUF-embedded IoT endpoints," in *Proceedings of the 2021 IEEE International Symposium On Circuits And Systems (ISCAS)*, pp. 1–5, Daegu, South Korea, May 2021.

[10] Y. Chen, J. Yuan, and Y. Zhang, "An improved password-authenticated key exchange protocol for VANET," *Vehicular Communications*, vol. 27, p. 100286, 2021.

[11] Y. Salami, Y. Ebazadeh, and V. Khajehvand, "CE-SKE: cost-effective secure key exchange scheme in Fog Federation," *Iran Journal of Computer Science*, vol. 4, no. 3, pp. 1–13, 2021.

[12] S. Kalra and S. K. Sood, "Secure authentication scheme for IoT and cloud servers," *Pervasive and Mobile Computing*, vol. 24, pp. 210–223, 2015.

[13] L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, no. 1, pp. 61–86, 2006.

[14] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of the APPSEM 2005 Workshop*, pp. 1–17, Frauenchiemsee, Germany, September 2005.

[15] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[16] D. Basin, S. Mödersheim, and L. Viganò, "An on-the-fly model-checker for security protocol analysis," *Computer Security-ESORICS 2003*, vol. 2808, no. 3, pp. 253–270, 2003.

[17] M. Turuani, "The CL-Atse protocol analyser," in *Term Rewriting and Applications*, F. Pfenning, Ed., pp. 277–286, Springer, Berlin Germany, 2006.

[18] A. Armando and L. Compagna, "SATMC: a SAT-based model checker for security protocols," in *Logics in Artificial Intelligence*, J. J. Alferes and J. Leite, Eds., Springer, Berlin Germany, 2004.

[19] Y. Boichut, N. Kosmatov, and L. Vigneron, "Validation of Prouvé protocols using the automatic tool TA4SP," *TFIT*, vol. 6, pp. 467–480, 2006.

[20] "NS3." https://www.nsnam.org.