Cryptanalysis and Countermeasures of "LAAKA: Lightweight Anonymous Authentication and Key Agreement Scheme for Secure Fog-Driven IoT Systems"

DeokKyu Kwon
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, South Korea
kdk145@knu.ac.kr

Seunghwan Son
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, South Korea
sonshawn@knu.ac.kr

Youngho Park
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, South Korea
parkyh@knu.ac.kr

***Abstract*—Fog computing is a technology that fog servers cover the computational tasks of cloud server. Therefore, end devices can receive more real-time and localized services from fog servers. Therefore, researchers integrate fog computing and Internet of Things (IoT) to supplement the resource constraint problem of IoT devices and process data services in network edge. In 2024, Ali et al. proposed a mutual authentication and key agreement protocol to preserve anonymous and lightweight communications in fog-driven IoT environments. They utilized only hash functions and exclusive-OR (XOR) operators considering hardware specifications of IoT devices. In this work, we cryptanalysis Ali et al.'s authentication protocol to prove that "ephemeral secret leakage (ESL)" and "stolen verifier attacks" can be performed in their protocol. Moreover, we discover that Ali et al.'s protocol has a "desynchronization problem" where network entities cannot conduct authentication after initial communication. To supplement these security flaws, we conduct a discussion and present countermeasures, such as physically unclonable function (PUF), dynamic update of temporary identity, and usage of long-term secret parameters.**

***Index Terms*—Authentication, countermeasure, cryptanalysis, ephemeral secret leakage, stolen verifier.**

I. Introduction

With the development of communication and data processing technologies, Internet of Things (IoT) has applied to various network services, such as smart home [1], wireless medical sensor networks (WMSN) [2], and industrial IoT (IIoT) [3]. The widespread of IoT technology causes the explosion of information which are sent to cloud server. This can burden the storage and computation resources of cloud server because IoT devices collect and transmit surrounding circumstances in real-time. Moreover, physical distances between the cloud server and IoT devices are generally far, which can disrupt the real-time communications. If the information of IoT devices is stored in single cloud server, it can be vulnerable from service interruption, such as single point of failure (SPOF) problems.

Fog computing is a technology that can provide IoT services nearby the network edge [4]. In fog-driven environments, fog servers are deployed in specific regions with sufficient computing and storage capacities. IoT devices in this region collect the surrounding information and send it to the fog server. The fog server manages the deployed region and processes the information which are collected from IoT devices to generate useful services. Therefore, fog-driven IoT environments can provide localized and real-time services such as power prediction, weather forecasting, and road guidance.

Although fog-driven IoT environments have various advantages compared with the traditional cloud-based IoT, security challenges are still remained because messages are transmitted through open channel. If an adversary collects these messages, it can attempt to reveal sensitive information of fog servers and IoT devices. Moreover, the adversary can physically capture an IoT device and extract parameters to calculate secret information from that. If the database of a fog server is leaked, the adversary can try to compute network-critical information such as session key and master key. Generally, fog-driven IoT environments require lightweight computational loads because IoT devices have limited computation and storage capacities. To ensure security and preserve privacy in fog-driven IoT environments, designing a robust and lightweight authentication protocol is crucial.

In 2024, Ali et al. [5] proposed a mutual authentication protocol to prevent various security threats and enhance computation performance for fog-driven IoT environments. They

This study was supported by the "BK21 Four project funded by the Ministry of Education, Korea (4199990113966)."

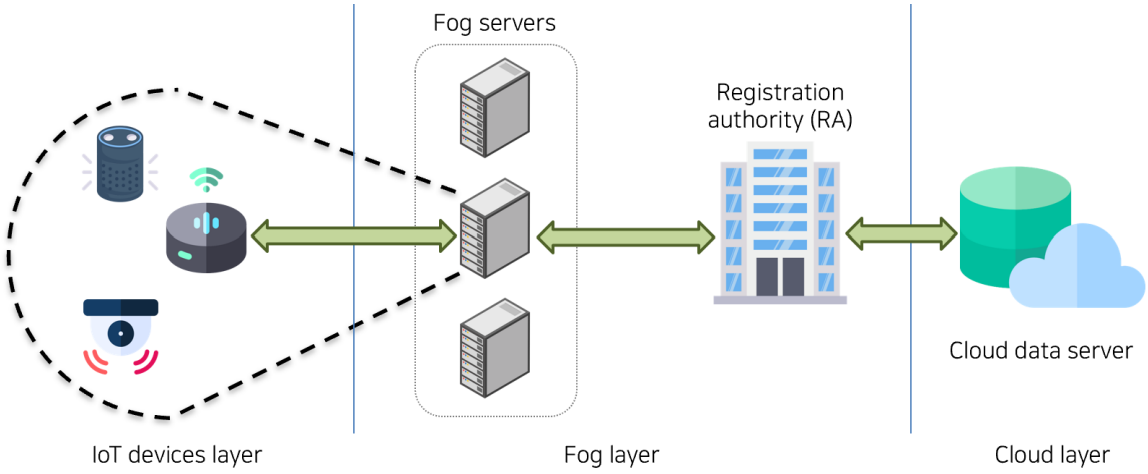

Fig. 1. System model for fog-driven IoT environments.

presented that the proposed authentication protocol can be lightweight by using only "hash functions" and "exclusive-OR (XOR) operators". Moreover, they argued that their protocol can provide anonymous authentication using temporary identities, which is updated in every session. Unfortunately, we found that Ali et al.'s protocol cannot prevent security attacks and suffers from update issues. In this work, we introduce that Ali et al.'s protocol cannot resist "ephemeral secret leakage (ESL)" and "stolen verifier attacks". Moreover, we show that Ali et al.'s protocol cannot ensure a smooth update of temporary identities. From that, we introduce the countermeasure to mitigate these security threats in Ali et al.'s protocol.

II. Preliminaries

A. System Model

This section outlines the system model designed for fog-driven IoT environments, consisting of four main entities: IoT devices, fog server, registration authority (RA), and cloud data server. As shown in Fig. 1, the system model is described with the following details.

- IoT device : IoT devices have various sensors to collect the surrounding circumstances. Then, IoT devices sends the information to a fog server because they have limited computation and storage resources. To join the proposed network, IoT devices must register to RA.
- Fog server : A fog server is deployed in specific area to manage IoT devices in this region. Moreover, the fog server can process data which is sent from IoT devices. Fog servers have sufficient computation and storage resources.
- Registration authority (RA) : RA initiates the fog-driven IoT environments and manage the sensitive information of IoT devices and fog servers. RA has enough computation and storage resources.
- Cloud data server : Cloud data server is located in cloud layer which has a large computation and storage capacities. Therefore, cloud data server can process a massive data processing, statistics, and analytics.

B. Threat Model

We employ "Dolev-Yao (DY) threat model [6]" in this paper, as it is widely recognized in authentication protocols [7], [8]. In DY threat model, an adversary has ability that can intercept, delete, eavesdrop, and modify messages transmitted through open channels. Moreover, we applied "Canetti-Krawczyk (CK) [9] threat model". In CK threat model, an adversary can obtain short-term secret parameters (e.g. ephemeral information) or long-term secrets (e.g. master key). Therefore, the adversary can execute various security exploits, including:

- Calculating the session key based on ephemeral secret parameters [10].
- Unveiling the verification table to extract sensitive data [11].
- Performing attacks including "forgery," "replay," "man-in-the-middle," "desynchronization," and "IoT device capture." [12]

III. Cryptanalysis of Ali et al.'s Protocol

We review and analyze Ali et al.'s protocol which consists of four phases : "Initialization", "Fog server registration", "IoT device registration", and "Authentication and key agreement phases". Table I shows the explanation of each notation used in Ali et al.'s protocol.

A. Review of Ali et al.'s Protocol

1) Initialization Phase: In this phase, the RA selects a master key K and hash function $h(.)$.

2) Fog Server Registration Phase: To generate and process convenient services, a fog server must register to RA. We show the "fog server registration phase" in Fig. 2-(a). The following outlines the detailed procedure:

Step 1 : The fog server F_i selects its identity ID_f. Then, F_i generates a random number r_1 and computes

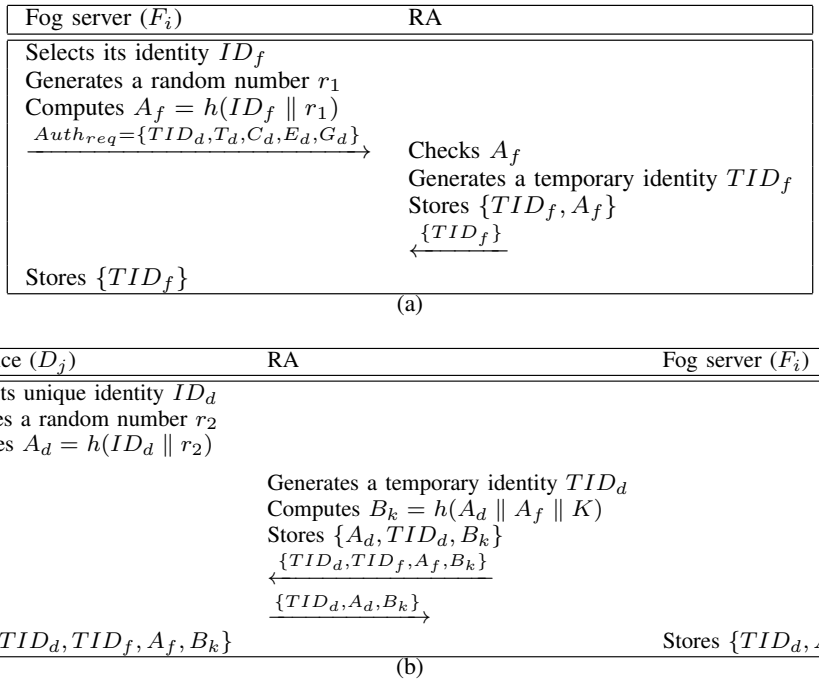

Fig. 2. Fog server (a) and IoT device (b) registration phases.

TABLE I
NOTATIONS AND DESCRIPTIONS USED IN ALI ET AL.'S PROTOCOL

Notation	Description
F_i, D_j	"Fog server" and "IoT device"
RA	"Registration authority"
ID_d, ID_f	Real identity of "D_j" and "F_j"
TID_d, TID_f	Temporary identity of "D_j" and "F_i"
r	"Random number"
T	"Timestamp"
SK	"Session key"
$h(.)$	"Hash function"
$\oplus$	"XOR operator"
$\parallel$	"Concatenation operator"

$A_f = h(ID_f \parallel r_1)$. F_i sends $\{A_f\}$ to RA via a secure channel.

Step 2 : The RA checks the validity of A_f and generates a temporary identity TID_f. Then, the RA stores $\{TID_f, A_f\}$ in secure database. After that, the RA transmits $\{TID_f\}$ to F_i through a secure channel.

Step 3 : F_i stores TID_f in its database.

3) IoT Device Registration Phase: To connect with the proposed network system, each IoT device must conduct a registration process. We show the "IoT device registration phase" in Fig. 2-(b). The steps in detail are outlined below:

Step 1 : An IoT device D_j selects its unique identity ID_d and generates a random number r_2. Then, D_j computes $A_d = h(ID_d \parallel r_2)$. Through a secure channel, the IoT device transmits $\{A_d\}$ to RA.

Step 2 : The RA checks $\{A_d\}$ and generates a temporary identity TID_d. After that, the RA computes $B_k = h(A_d \parallel A_f \parallel K)$ using its master key K. The RA stores $\{A_d, TID_d, B_k\}$ in its secure database and sends $\{TID_d, TID_f, A_f, B_k\}$, $\{TID_d, A_d, B_k\}$ to D_j, F_i, respectively.

Stem 3 : D_j stores $\{TID_d, TID_f, A_f, B_k\}$ in its memory. Moreover, F_i stores $\{TID_d, A_d, B_k\}$ in its database.

4) Authentication and Key Agreement Phase: The IoT device and fog server establish a secure channel using the registration parameters. Fig. 3 shows the "authentication and key agreement phase" and the following outlines the detailed procedure:

Step 1 : The IoT device D_j generates a random nonce r_d and timestamp T_d. Then, D_j computes $C_d = h(T_d \parallel r_d)$, $E_d = r_d \oplus h(B_k \parallel A_f)$, $TID_d^{new} = TID_d \oplus r_d$, $G_d = h(A_d \parallel TID_d^{new} \parallel B_k \parallel r_d)$. After that, D_j sends an authentication request message $Auth_{req} = \{TID_d, T_d, C_d, E_d, G_d\}$ to fog server F_i via an open channel.

Step 2 : F_i firstly checks the validity of TID_d and timestamp $T_c - T_d < \Delta T$. Then, F_i computes $r_d^* = E_d \oplus h(B_k \parallel A_f)$ and $C_d^* = h(T_d \parallel r_d^*)$ to check the legitimacy of r_d. If it is valid, F_i computes $TID_d^{new} = TID_d \oplus r_d^*$ and $G_d^* = h(A_d \parallel TID_d^{new} \parallel B_k \parallel r_d^*)$. If $C_d^* \stackrel{?}{=} C_d$ and $G_d^* \stackrel{?}{=} G_d$, F_i selects a random nonce r_f and timestamp T_f. Then, F_i computes $C_f = h(T_f \parallel r_f)$ and selects an additional timestamp T_s. F_i computes a session key $SK = h(r_d \parallel r_f \parallel T_s)$, $E_f = r_f \oplus h(TID_d^{new})$, $TID_f^{new} = TID_f \oplus r_f$, $G_f = h(TID_f^{new} \parallel B_k \parallel r_f \parallel SK \parallel T_s)$. After that, F_i sends an authentication response message $Auth_{rep} = \{TID_f, T_f, T_s, C_f, E_f, G_f\}$ to D_j through an open channel.

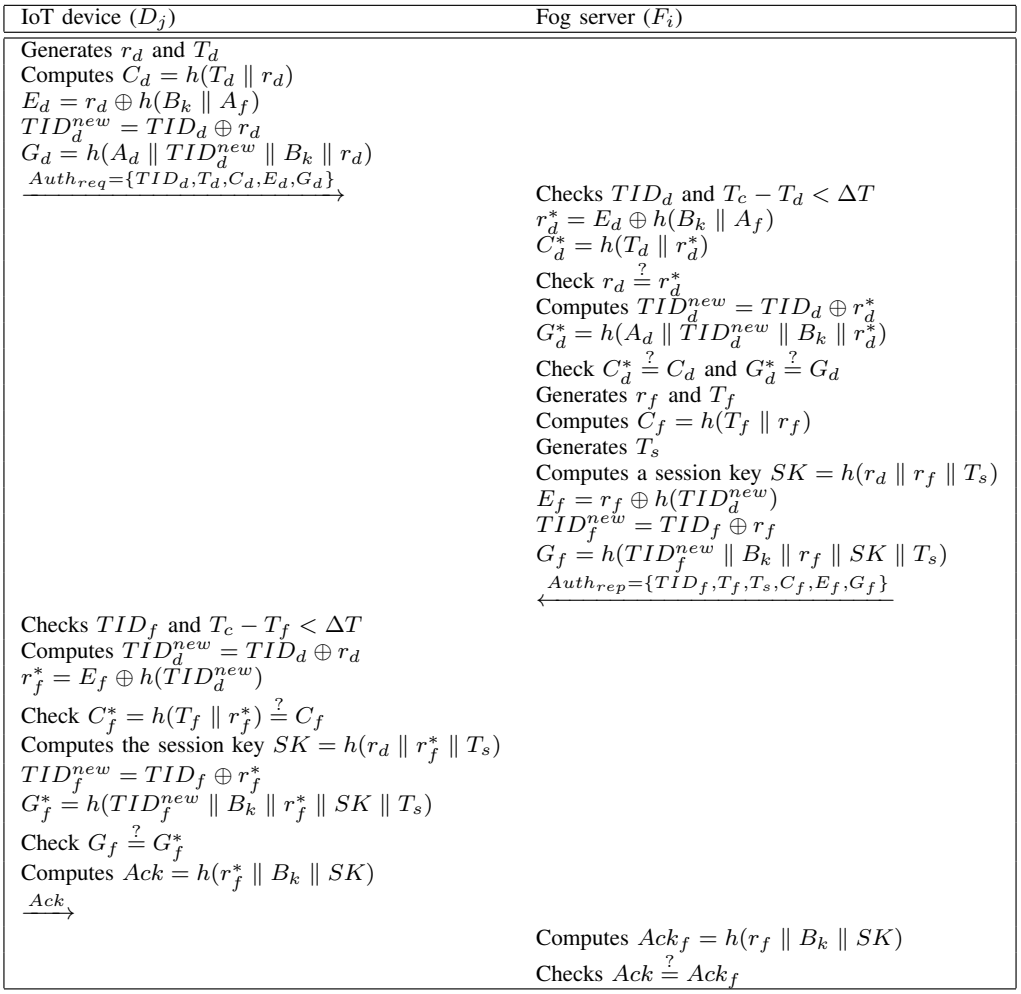

Fig. 3. Authentication and key agreement phase.

Step 3 : D_j checks the validity of TID_f and timestamp $T_c - T_f < \Delta T$. Then, D_j computes $TID_d^{new} = TID_d \oplus r_d$ and $r_f^* = E_f \oplus h(TID_d^{new})$. If $C_f^* = h(T_f \parallel r_f^*) \stackrel{?}{=} C_f$, D_j computes the session key $SK = h(r_d \parallel r_f^* \parallel T_s)$, $TID_f^{new} = TID_f \oplus r_f^*$, $G_f^* = h(TID_f^{new} \parallel B_k \parallel r_f^* \parallel SK \parallel T_s)$. If G_f^* and G_f are equal, D_j can ensure the authentication with F_j. Finally, D_j computes $Ack = h(r_f^* \parallel B_k \parallel SK)$ and sends it to F_i through an open channel.

Step 4 : F_j computes $Ack_f = h(r_f \parallel B_k \parallel SK)$ and checks $Ack_f \stackrel{?}{=} Ack$. If it checks out, the authentication and key agreement phase succeeds.

B. Security Weaknesses of Ali et al.'s Protocol

We prove that ESL and stolen verifier attacks can be valid in the above protocol. The specifics are outlined below.

1) ESL Attacks: An adversary can obtain the short term secret parameters according to Section II-B. Using these parameters, the adversary can calculate SK. Detailed steps are as below:

Step 1 : The adversary obtains ephemeral secret parameters r_d and r_f. Moreover, the adversary eavesdrops a message $Auth_{rep} = \{TID_f, T_f, T_s, C_f, E_f, G_f\}$.

Step 2 : The adversary calculates the session key $SK = h(r_d \parallel r_f \parallel T_s)$ using r_d, r_f, and a timestamp T_s.

Accordingly, Ali et al.'s protocol has difficulty mitigating stolen verifier attacks.

2) Stolen Verifier Attacks: The adversary gets the verification data which is leaked from RA in this attack. From that, the adversary can compute SK. The following are the detailed steps:

Step 1 : The adversary obtains the verification table $\{A_d, TID_d, B_k\}$ and $\{TID_f, A_f\}$ from RA. Moreover, the adversary intercepts messages $Auth_{req} = \{TID_d, T_d, C_d, E_d, G_d\}$ and $Auth_{rep} = \{TID_f, T_f, T_s, C_f, E_f, G_f\}$.

Step 2 : The adversary computes $r_d = E_d \oplus h(B_k \parallel A_f)$, $TID_d^{new} = TID_d \oplus r_d$, and $r_f = E_f \oplus h(TID_d^{new})$ using $\{E_d, E_f\}$ and $\{TID_d, B_k, A_f\}$.

Step 3 : The adversary computes the session key $SK = h(r_d \parallel r_f \parallel T_s)$ using the timestamp T_s from the message $Auth_{rep}$.

Consequently, stolen verifier attacks can be valid to Ali et al.'s protocol.

3) Desynchronization Problem of Temporary Identity: In Section III-A4, the IoT device validates TID_f from the message $Auth_{rep} = \{TID_f, T_f, T_s, C_f, E_f, G_f\}$. Then, temporary identities TID_d and TID_f are updated to TID_d^{new} and TID_f^{new}. To authenticate to the fog server F_i in another session, each IoT devices must know TID_f^{new}. However, other IoT devices (e.g. TID_{d-2}, TID_{d-3} ... TID_{d-n}) cannot authenticate with F_i because they do not receive notification of the updated fog server's temporary identity TID_f^{new}.

C. Discussion and Countermeasures

In Ali et al.'s protocol, the session key SK can be revealed by using ephemeral secret parameters and timestamps. Moreover, each parameter in $Auth_{req}$ and $Auth_{rep}$ can be easily decrypted using RA's verification table. Ali et al.'s protocol has desynchronization problem from the perspective of the entire network. Therefore, we propose several countermeasures to ensure a high level of security for fog-driven IoT environments.

- **Physically unclonable function (PUF)** : PUF is a digital fingerprint using the difference of molecular structure in manufacturing of semiconductors. In cryptographic aspects, PUF can retrieve a unique private parameter because even the same product can produce different results. Therefore, the adversary cannot guess or extract secret parameters using PUF. The idealized function of PUF is written as $Response = PUF(Challenge)$. In Ali et al.'s protocol, we can utilize PUF in IoT devices when it generates B_k.
- **Dynamic update of temporary identity** : In authentication and key agreement phase, F_i and D_j compute an updated temporary identity TID_f^{new} and stores it in only their memory. This can cause desynchronization problem because another IoT device D_k do not have TID_f^{new} of F_i. Therefore, we suggest to publish TID_f in the entire network.
- **Usage of long-term secret and short-term secret parameters in session key** : In authentication and key agreement phase, F_i and D_j establish a session key $SK = h(r_d \parallel r_f \parallel T_s)$ which are composed of random nonces and timestamp. Therefore, we suggest to establish SK using long-term secret parameters, such as B_k and A_f.

IV. Conclusions

We analyzed Ali et al.'s protocol [5] to prove that their protocol cannot prevent ESL, stolen verifier attacks. Moreover, Ali et al.'s protocol has desychonization problem that other IoT devices cannot access to the fog server. Therefore, we presented countermeasures to solve these flaws, such as PUF, dynamic update of TID, and usage of long-term secret parameters in session key. In future works, we will propose an authentication protocol applying these countermeasures.

References

[1] K. Maswadi, N. B. A. Ghani, and S. B. Hamid, S. B. "Systematic literature review of smart home monitoring technologies based on IoT for the elderly," *IEEE Access*, vol. 8, pp. 92244-92261, 2020.

[2] K. Ghoumid, D. Ar-Reyouchi, S. Rattal, R. Yahiaoui, and O. Elmazria, "Protocol wireless medical sensor networks in IoT for the efficiency of healthcare," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10693-10704, 2021.

[3] I. T. Christou, N. Kefalakis, J. K. Soldatos, and A. M. Despotopoulou, "End-to-end industrial IoT platform for Quality 4.0 applications," *Computers in Industry*, vol. 137, pp. 103591-103603, 2022.

[4] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of network and computer applications*, vol. 98, pp. 27-42, 2017.

[5] H. Ali, and I. Ahmed, "LAAKA: Lightweight anonymous authentication and key agreement scheme for secure fog-driven IoT systems," *Computers & Security*, vol. 140, pp. 103770–103787. 2024.

[6] D. Dolev, and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, pp. 198–208, 1983.

[7] S. Son, J. Lee, Y. Park, Y. Park, and A. K. Das, "Design of blockchain-based lightweight V2I handover authentication protocol for VANET," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1346–1358, 2022.

[8] M. Kim, J. Lee, J. Oh, K. Park, Y. Park, and K. Park, "Blockchain based energy trading scheme for vehicle-to-vehicle using decentralized identifiers," *Applied Energy*, vol. 322, pp. 119445–119453 , 2022.

[9] R. Canetti, and H. Krawczyk, "Universally composable notions of key exchange and secure channels," In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT'02), Amsterdam, The Netherlands, May 2002, pp. 337–351.

[10] S. Yu, and Y. Park, "A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20214–20228, 2022.

[11] D. Kwon, S. Son, M. Kim, J. Lee, A. K. Das, and Y. Park, "A secure self-certified broadcast authentication protocol for intelligent transportation systems in UAV-assisted mobile edge computing environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 19004 – 19017, 2024.

[12] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K. K. R. Choo, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8739-8752, 2019.