Cryptanalysis and Countermeasures of "A Secure Lightweight Identity Authentication and Key Agreement Scheme for Internet of Drones"

Taehun Kim
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, South Korea
kimth028@knu.ac.kr

Sangjun Lee
School of Electronic and Electrical Engineering
Kyungpook National University
Daegu, South Korea
gumoning9010@knu.ac.kr

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

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

***Abstract*—In Internet of drones (IoD) environments, drones execute various tasks including search operations, delivery services, and farming. However, drones can be feasible targets for physical attacks. Moreover, communicating with users and the server through public channels can allow security attacks such as replay, impersonation, and man-in-the-middle (MitM) attacks. Therefore, secure authentication schemes to establish session keys are essential. In July 2025, Dong et al. designed an authentication scheme for IoD environments. In their paper, they claimed that users can establish a session key securely with drones through the server. Unfortunately, we have found that their scheme is vulnerable to user impersonation and session key disclosure attacks, and that drones cannot recover parameters received from the server. In this paper, we provide detailed review of Dong et al.'s scheme and cryptanalysis, and propose countermeasures for secure authentication in IoD environments.**

***Index Terms*—IoD, mutual authentication, cryptanalysis, countermeasures.**

I. Introduction

For the past few decades, drones have experienced significant evolution. The IoD is a networked architecture that allows access to drones in dedicated flight zones. In IoD environments, drones are used for many services, such as drone delivery [1], smart agriculture [2], and rescue mission [3]. Drones are equipped with sensors, collecting data from their surroundings. The information gathered by drones are sent to servers for analysis and can also be sent to users for real-time monitoring [4].

Nevertheless, as IoD utilizes public channel for message transmission, adversaries can perform security attacks to reveal private information [5]. Moreover, drones are especially vulnerable to capture due to physical characteristics [6]. If adversaries capture drones and extract stored parameters, adversaries can attempt to compute session keys. Furthermore, drones have limited batteries and computing power to perform complex cryptographic operations and data analysis. Therefore, secure and lightweight authentication schemes are essential for IoD environments.

In 2025, Dong et al. [7] proposed an identity authentication scheme for IoD environments to secure data transmission between drones and users. They utilized fuzzy extractor [8] for secure login phase and asserted their their scheme can prevent security attacks and ensure user anonymity. However, we found out that their scheme cannot resist user impersonation and session key disclosure attacks, and has drone authentication problem. Therefore, we suggest some countermeasures to supplement Dong et al.'s scheme.

This research was supported by the Regional Innovation System & Education(RISE) Global 30 program through the Daegu RISE Center, funded by the Ministry of Education(MOE) and the Daegu, Republic of Korea.(2025-RISE-03-001)

II. Preliminaries

A. System Model

The system model of IoD environments consists of users, server, and drones. Fig. 1 illustrates the system model of IoD, and details of each component are provided below.

1) User (U_i)*:* Users register themselves to server for authentication with server and drones. Users can access real-time or processed data captured by drones after authentication.

2) Server (S)*:* The server manages registration of users and drones, drone flights, processes data collected by drones, and supports mutual authentication. The server S is a fully-trusted entity, and is assumed to possess sufficient computing and storage resources to provide services to users.

3) Drone (DR_j)*:* In its dedicated flight zone, the drone captures and sends information to the server for analysis. After

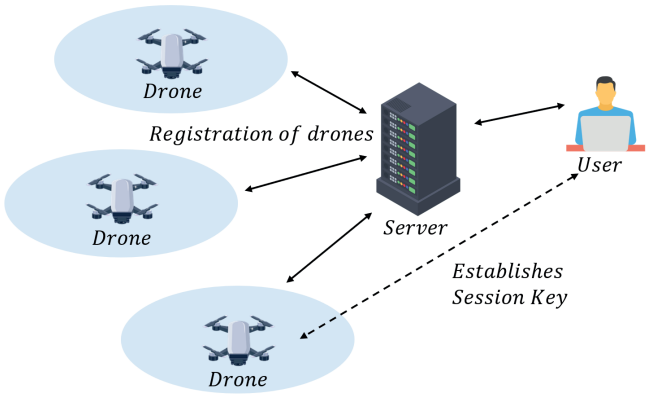

Fig. 1. System model of IoD environments.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
ΔT	Maximum transmission delay
T_U	Registration time of user
T_R	Registration time of drone
$T_i (i = 1, 2, 3)$	Timestamp
UID_i, $RUID_i$	Identity and pseudo-identity of user
SID_s, $RSID_s$	Identity and pseudo-identity of server
DID_j, $RDID_j$	Identity and pseudo-identity of drone
U_i, MD_i, S, DR_j	User, mobile device, server, and drone
$Gen(.)$, $Rep(.)$	Fuzzy extractor functions
h	One-way hash function
$\oplus$	Exclusive-OR operation

session key establishment, the drone can communicate with the user to provide real-time information.

B. Threat Model

We adopt widely-used Dolev-Yao (DY) model [9], [10] in this paper. Under the DY model, adversaries have following capabilities:

- An adversary $\mathcal{A}$ can intercept, insert, eavesdrop, and delete messages transmitted over public channels. Using collected messages, $\mathcal{A}$ can perform security attacks on users, the server, and drones.
- $\mathcal{A}$ can steal legitimate user's device and conduct power analysis to extract stored parameters.
- $\mathcal{A}$ can capture drones deployed in their flight zone and extract stored parameters using power analysis. However, adversary cannot replicate PUFs applied to drones.

III. Cryptanalysis of Dong et al.'s Scheme

We review and conduct cryptanalysis on Dong et al.'s scheme. Their scheme consists of "system initialization phase", "registration phase", "login and authentication phase", and "user password and biometric update phase". Table I explains notations used in Dong et al.'s scheme.

A. Review of Dong et al.'s Scheme

1) System Initialization Phase: Server S selects its identity SID_s and hash function $h(.)$. Then S chooses mask value m_i to each U_i, and stores $\{SID_s, m_i, T_R\}$.

2) Registration Phase: In this phase, user U_i and drone DR_j register to the server.

Step 1: U_i selects identity UID_i, generates registration timestamp T_U, and sends $\{UID_i, T_U, r_1^i\}$ to S through a secure channel.

Step 2: S receives the message, picks a random number K to calculate pseudo-identities $RUID_i = h(UID_i \| K)$, $RSID_s = h(SID_s \| K)$, and $RDID_j = h(DID_j \| K)$. Then S selects drone registration time T_R of drone DR_j, saves $\{K, RUID_i, RSID_s, RDID_j, r_1^i, T_U, T_R\}$ in database, and sends $\{RSID_s, RDID_j, RUID_i, m_i, T_R\}$ to MD_i of U_i, and $\{RSID_s, RDID_j, RUID_i, T_R\}$ to DR_j over a secure channel. DR_j stores the received message in its memory.

Step 3: After receiving the message, U_i inputs biometric B_i to compute $Gen(B_i) = (\sigma_i, \tau_i)$. Then, U_i inputs password PW_i, generates random number k_1^i, and calculates $RPW_i = h(PW_i \| k_1^i)$, $d_{reg} = h(RPW_i \| RUID_i \| T_U \| \sigma_i)$, $e_{reg} = h(RUID_i \| RPW_i \| T_U \| \sigma_i)$, $s_{reg} = (RPW_i + d_{reg}) \cdot r_1^i \mod q$, $\sigma_{reg} = s_{reg}(e_{reg} \cdot m_i + k_1^i)$, $A = h(RUID_j \| RPW_i \| \sigma_{reg})$ and stores $\{RUID_i, RSID_s, RDID_j, m_i, A, T_R, Gen(.), Rep(.), \tau_i, k_1^i, T_U, r_1^i, UID_i\}$ in MD_i.

3) Login and Authentication Phase: In this phase, U_i and DR_j establish a session key through the assistance of S. Detailed steps of Dong et al.'s login and authentication phase are as follows, and is summarized in Fig. 2.

Step 1: U_i inputs $\{PW_i, UID_i, B_i\}$ to MD_i. MD_i calculates $\sigma_i = Rep(B_i, \tau_i)$, $RPW_i = h(PW_i \| k_1^i)$, $d_{reg} = h(RPW_i \| RUID_i \| T_U \| \sigma_i)$, $e_{reg} = h(RUID_i \| RPW_i \| T_U \| \sigma_i)$, $s_{reg} = (RPW_i + d_{reg}) \cdot r_1^i \mod q$, $\sigma_{reg} = s_{reg}(e_{reg} \cdot m_i + k_1^i)$, and $A' = h(RUID_i \| RPW_i \| \sigma_{reg})$. If $A = A'$, U_i successfully logins to MD_i, and MD_i generates the timestamp T_1. Then MD_i computes $d = h(RUID_i \| RSID_s \| RDID_j \| T_U \| m_i)$, $e = h(RDID_j \| RSID_s \| RUID_i \| T_U \| m_i)$, $s_{U_i} = [h(r_1^i \| T_1) + d] \cdot m_i \mod q$, $\sigma_{U_i} = s_{U_i}(e \cdot m_i + r_1^i)$, $M_1 = RDID_i \oplus h(RSID_s \| T_1)$, $M_2 = h(\sigma_{U_i} \| T_U \| RUID_i \| T_1)$, and sends $Msg_1 = \{M_1, M_2, T_1\}$ to S by public channel.

Step 2: Receiving the message from U_i, S first verifies the freshness of T_1. After the verification, S computes $RDID_i = M_1 \oplus h(RSID_s \| T_1)$ and checks for its existence in the database. If $RDID_i$ is in the database, S retrieves T_R and

User U_i	**Server** S	**Drone** DR_j
Inputs PW_i, UID_i, B_i Calculates $\sigma_i' = Rep(B_i, \tau_i)$ $RPW_i' = h(PW_i \| k_1^i)$ $d_{reg}' = h(RPW_i' \| RUID_i \| T_U \| \sigma_i')$ $e_{reg}' = h(RUID_i \| RPW_i' \| T_U \| \sigma_i')$ $s_{reg}' = (RPW_i' + d_{reg}') \cdot r_1^i \bmod q$ $\sigma_i' = s_{reg}'(e_{reg}' \cdot m_i + k_1^i)$ $A' = h(RUID_i \| RPW_i' \| \sigma_{reg}')$		
Checks if $A' \stackrel{?}{=} A$ Generates T_1 Calculates $d = h(RUID_i \| RSID_s \| RDID_j \| T_U \| m_i)$ $e = h(RDID_j \| RSID_s \| RUID_i \| T_U \| m_i)$ $s_{U_i} = [h(r_1^i \| T_1) + d] \cdot m_i \bmod q$ $\sigma_{U_i} = s_{U_i}(e \cdot m_i + r_1^i)$ $M_1 = RDID_i \oplus h(RSID_s \| T_1)$ $M_2 = h(\sigma_{U_i} \| T_U \| RUID_i \| T_1)$ $\xrightarrow{Msg_1 = \{M_1, M_2, T_1\}}$	If so, retrieve T_R and $RDID_j$ that corresponds to $RDID_i$. $d' = h(RUID_i \| RSID_s \| RDID_j \| T_U \| m_i)$ $e' = h(RDID_j \| RSID_s \| RUID_i \| T_U \| m_i)$ $s_s = [h(r_1^i \| T_1) + d'] \cdot m_i \bmod q$ $\sigma_s = s_s(e' \cdot m_i + r_1^i)$ $M_2' = h(\sigma_s \| T_U \| RUID_i \| T_1)$ Checks if $M_2' \stackrel{?}{=} M_2$ Generates r_2 and T_2 Calculates $M_3 = h(T_R \| RDID_j) \oplus h(\sigma_s \| r_1^i \| r_2)$ $M_4 = h(RDID_j \| T_2) \oplus h(\sigma_s)$ $M_5 = h(h(\sigma_s \| r_1^i \| r_2) \| T_2)$ $\xrightarrow{Msg_2 = \{M_3, M_4, M_5, T_2\}}$	
		Checks if $\|T - T_2\| \le \Delta T$ $M_6 = M_3 \oplus h(T_R \| RDID_j)$ $M_5' = h(M_6 \| T_2)$ Checks if $M_5' \stackrel{?}{=} M_5$ Generates r_3 and T_3 Calculates $M_7 = h(\sigma_s' \| RDID_j) \oplus r_3$ $h(\sigma_s)' = M_4 \oplus h(RDID_j \| T_2)$ $M_8 = h(RDID_j \| h(\sigma_s)' \| r_3) \oplus M_7$ $SK_{ij} = h(M_7 \| T_R \| RDID_j \| r_3)$ $M_9 = h(SK_{ij} \| T_3)$ $\xleftarrow{Msg_3 = \{M_7, M_8, M_9, T_3\}}$
Checks if $\|T - T_3\| \le \Delta T$ Calculates $r_3' = M_7 \oplus h(\sigma_s' \| RDID_j)$ $M_7' = M_8 \oplus h(RDID_j \| h(\sigma_s)' \| r_3')$ $SK_{ij}' = h(M_7' \| T_R \| RDID_j \| r_3')$ $M_9' = h(SK_{ij}' \| T_3)$ Checks if $M_9' \stackrel{?}{=} M_9$		

Fig. 2. Login and AKA phase of Dong et al.'s scheme.

$RDID_j$ corresponding to $RDID_i$. Then, S calculates $d_i' = h(RUID_i \| RSID_s \| RDID_j \| T_U \| m_i)$, $s_s = [h(r_1^i \| T_1) + d'] \cdot m_i \bmod q$, $\sigma_s = s_s(e' \cdot m_i + r_1^i)$, $M_1 = RDID_i \oplus h(RSID_s \| T_1)$, $M_2' = h(\sigma_s \| T_U \| RUID_i \| T_1)$, and checks whether $M_2' = M_2$. If equal, S generates T_2 and r_2, and computes $M_3 = h(T_R \| RDID_j) \oplus h(\sigma_s \| r_1^i \| r_2^i)$, $M_4 = h(RDID_j \| T_2) \oplus h(\sigma_s)$, $M_5 = h(h(\sigma_s \| r_1^i \| r_2) \| T_2)$ and sends the message $Msg_2 = \{M_3, M_4, M_5, T_2\}$ to DR_j through public channel.

Step 3: After DR_j receives Msg_2 from S, DR_j checks the freshness of T_2. If T_2 is fresh, DR_j calculates $M_6 = M_3 \oplus h(T_R \| RDID_j)$, $M_5' = h(M_6 \| T_2)$, and checks $M_5' \stackrel{?}{=} M_5$. If equal, DR_j generates r_3, and calculates $M_7 = h(\sigma_s' \| RDID_j) \oplus r_3$, $h(\sigma_s)' = M_4 \oplus h(RDID_j \| T_2)$, $M_8 = h(RDID_j \| h(\sigma_s)' \| r_3) \oplus M_7$, $SK_{ij} = h(M_7 \| T_R \| RDID_j \| r_3)$, $M_9 = h(SK_{ij} \| T_3)$. Then DR_j sends $Msg_3 = \{M_7, M_8, M_9, T_3\}$ to U_i through the public channel.

Step 4: U_i receives Msg_3, and verifies T_3. If T_3 is fresh, U_i calculates $r_3' = M_7 \oplus h(\sigma_s' \| RDID_j)$, $M_7' = M_8 \oplus h(RDID_j \| h(\sigma_s)' \| r_3')$, $SK_{ij}' = h(M_7' \| T_R \| RDID_j \| r_3')$, $M_9' = h(SK_{ij}' \| T_3)$, and checks $M_9' \stackrel{?}{=} M_9$. If equal, U_i and DR_j have established session key SK_{ij} for secure communication.

4) User Password and Biometric Update Phase: Legitimate users can update their passwords and biometrics.

Step 1: U_i inputs PW_i^{old}, B_i^{old}, calculates $\sigma_i = Rep(B_i^{old}, \tau_m)$, $RPW_i' = h(PW_i^{old} \| k_1^i)$, $d_{reg}' = h(RPW_i' \| RUID_i \| T_U \| \sigma_i)$, $e_{reg}' = h(RUID_i \| RPW_i' \| T_U \| \sigma_i)$, $s_{reg}' = (RPW_i' + d_{reg}') \cdot r_1^i \bmod q$, $\sigma_{reg}' = s_{reg}'(e_{reg}' \cdot m_i + k_1^i)$, $A' = h(RUID_i \| RPW_i' \| \sigma_{reg}')$ and checks $A' \stackrel{?}{=} A$. If equal, U_i can continue update phase. U_i enters PW_i^{new}, B_i^{new}, calculates $Gen(B_i^{new}) = (\sigma_i', \tau_m')$, $RPW_i' = h(PW_i^{new} \| k_1^i)$, $d_{reg}' = h(RPW_i' \| RUID_i \| T_U \| \sigma_i')$, $s_{reg}' = (RPW_i' + d_{reg}') \cdot r_1^i \bmod q$, $\sigma_{reg}' = s_{reg}'(e_{reg}' \cdot m_i + k_1^i)$, $A^{new} = h(RUID_i \| RPW_i \| \sigma_{reg}')$, and stores updated values.

B. Security vulnerabilities of Dong et al.'s Scheme

In Dong et al.'s scheme, we found that user impersonation and session key disclosure attacks can be performed to calculate session keys. Moreover, DR_j cannot recover necessary parameter for authentication.

1) User Impersonation Attacks: According to DY threat model, an adversary $\mathcal{A}$ can intercept, eavesdrop, modify,

and delete messages transmitted through open channels [11]. Moreover, $\mathcal{A}$ can extract stored parameters from user's mobile device using power analysis. Using extracted parameters, $\mathcal{A}$ can successfully forge Msg_1 to impersonate the user. Detailed steps are as follows.

- **Step 1:** $\mathcal{A}$ steals U_i's mobile device and performs power analysis to extract $\{T_R, T_U, RUID_i, RSID_s, RDID_j, r_1^i, m_i\}$.
- **Step 2:** $\mathcal{A}$ generates fresh timestamp $T_1^{\mathcal{A}}$, and computes $d = h(RUID_i \| RSID_s \| RDID_j \| T_U \| m_i)$, $e = h(RDID_j \| RSID_s \| RUID_i \| T_U \| m_i)$, $s_{U_i} = [h(r_1^i \| T_1^{\mathcal{A}}) + d] \cdot m_i \bmod q$, $\sigma_{U_i} = s_{U_i}(e \cdot m_i + r_1^i)$, $M_1^{\mathcal{A}} = RDID_i \oplus h(RSID_s \| T_1^{\mathcal{A}})$, $M_2^{\mathcal{A}} = h(\sigma_{U_i} \| T_U \| RUID_i \| T_1^{\mathcal{A}})$, and send $Msg_1^{\mathcal{A}} = \{M_1^{\mathcal{A}}, M_2^{\mathcal{A}}, T_1^{\mathcal{A}}\}$ to the server.
- **Step 3:** S receives $Msg_1^{\mathcal{A}} = \{M_1^{\mathcal{A}}, M_2^{\mathcal{A}}, T_1^{\mathcal{A}}\}$, and follows the steps of the scheme because $T_1^{\mathcal{A}}$ is a fresh timestamp value, and $M_1^{\mathcal{A}}$ and $M_2^{\mathcal{A}}$ are created using legitimate user's parameters and $T_1^{\mathcal{A}}$.

Therefore, Dong et al.'s scheme cannot prevent user impersonation attacks.

2) Session Key Disclosure Attacks: $\mathcal{A}$ can calculate the session key after performing user impersonation attacks. Following is the procedures for this attack.

- **Step 1:** $\mathcal{A}$ receives $Msg_3 = \{M_7, M_8, M_9, T_3\}$ from public channel, and calculates $r_3 = M_7 \oplus h(\sigma_s \| RDID_j)$.
- **Step 2:** With collected parameters $\{T_R, T_U, RUID_i, RSID_s, RDID_j, r_1^i, M_i, M_7, M_8, M_9, T_3\}$, $\mathcal{A}$ can calculate the session key $SK_{ij} = h(M_7 \| T_R \| RDID_j \| r_3)$.

Therefore, Dong et al.'s scheme is insecure to session key disclosure attacks.

3) Drone Authentication Problem: According to Dong et al.'s scheme, DR_j generates r_3 and calculates $M_7 = h(\sigma_s' \| RDID_j) \oplus r_3$. However, DR_j needs $\{T_U, m_i, T_1, r_1^i\}$ to calculate $\sigma_s = s_s(e' \cdot m_i + r_1^i)$, but DR_j does not know these values. Therefore, DR_j cannot calculate M_7, M_8, and SK_{ij}, and send message $Msg_3 = \{M_7, M_8, M_9, T_3\}$ to U_i.

IV. Countermeasures

Dong et al.'s scheme cannot prevent user impersonation and session key disclosure attacks which leads to leakage of session key SK_{ij}. Moreover, drone cannot calculate SK_{ij} and send Msg_3 to U_i. To cope with these issues, we suggest countermeasures for secure authentication and key agreement in IoD environments.

- **Physically unclonable function (PUF)** : PUF is a one-way function that is based on the randomness during the manufacturing process of integrated circuits. PUF provides the following features: output unpredictability, impossibility of replication, and uniqueness. Therefore, adversaries cannot copy PUFs or guess outputs. PUF can be expressed as $Response = PUF(Challenge)$. We can utilize PUF in drones to mask r_3.
- **Encryption of stored parameters** : User devices store $\{T_R, T_U, RUID_i, RSID_s, RDID_j, r_1^i, m_i, A, \tau_i, k_1^i, UID_i\}$ in plaintext. Among these values, $RDID_j$ and T_R are used directly in calculation of session keys $SK_{ij} = h(M_7 \| T_R \| RDID_j \| r_3)$. r_3 can be recovered with σ_s which can be also calculated using stored parameters such as r_1^i, T_U and m_i. Therefore, we suggest encrypting stores parameters using PW_i and σ_i. User can generate new parameter $MPW_i = h(PW_i \| \sigma_i)$ and use it to encrypt stored parameters. Because $\mathcal{A}$ cannot calculate MPW_i, this can prevent leakage of stored parameters.

V. Conclusions

In this paper, we reviewed Dong et al.'s authentication scheme for IoD environments. We analyzed that their scheme cannot prevent user impersonation and session key disclosure attacks, and has drone authentication problem. Therefore, we suggested countermeasures of security vulnerabilities for Dong et al.'s scheme in this paper. Use of PUFs and encrypting stored parameters can make the scheme more secure. In the future, we plan to devise authentication scheme for IoD environments.

References

[1] G. Attenni, V. Arrigoni, N. Bartolini, and G. Maselli, "Drone-based delivery systems: A survey on route planning," *IEEE Access*, vol. 11, pp. 123476-123504, 2023.

[2] P. K. R. Maddikunta, S. Hakak, M. Alazab, S. Bhattacharya, T. R. Gadekallu, W. Z. Khan, and Q. V. Pham, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, vol. 21, no. 16, pp.17608-17619, 2021.

[3] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp.1-10, 2020.

[4] M. Bakirci, "Performance evaluation of low-power and lightweight object detectors for real-time monitoring in resource-constrained drone systems," *Engineering Applications of Artificial Intelligence*, vol. 159, pp. 111775-111795, 2025.

[5] 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.

[6] J. Choi, S. Son, D. Kwon, and Y. Park, "A puf-based secure authentication and key agreement scheme for the internet of drones," *Sensors*, vol. 25, no. 3, pp. 982, 2025.

[7] W. Dong, X. Wang, and J. Li, "A secure lightweight identity authentication and key agreement scheme for internet of drones," *Computer Networks*, vol. 270, pp.111503-111416, 2025.

[8] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the the theory and applications of cryptographic techniques*, pp. 523-540, 2004.

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

[10] M. Wazid, J. Singh, C. Pandey, R. S. Sherratt, A. K. Das, D. Giri, and Y. Park, "Explainable deep Learning-Enabled malware attack detection for IoT-Enabled intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 5, pp. 7231-7244, 2025.

[11] D. Gautam, G.Thakur, P. Kumar, A. K. Das, and Y. Park, "Blockchain assisted intra-twin and inter-twin authentication scheme for vehicular digital twin system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 15002-15015, 2024.