COMMENTARY

Comments on *CSAP-IoD: A Chaotic Map-Based Secure Authentication Protocol for Internet of Drones*

Akash Kumar[1,*] **and Saru Kumari**[1]

[1] Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, Uttar Pradesh, India

Abstract

Recently, Zahednejad and Gao (2025 *Journal of Information Security and Applications*, Elsevier, https://doi.org/10.1016/j.jisa.2025.104083,) identified key compromise impersonation (KCI) in both, a two-factor, and a three-factor authentication protocols for IoT devices. They further proposed a lightweight authentication scheme designed to mitigate the identified KCI threats. Subsequently, Zhang et al. (2025, *IEEE Transactions on Information Forensics and Security*, https://doi.org/10.1109/TIFS.2025.3599678) introduced a chaotic map-based authentication protocol for the Internet of Drones, claiming that their design is resilient to a wide range of security attacks. Earlier, Kumari et al. (2019) had emphasized the significance of KCI attacks, urging the research community to devise robust countermeasures to eliminate KCI and other vulnerabilities arising from the compromise of a trusted authority's keys. We observe that the protocol proposed by Zahednejad and Gao effectively resists KCI, even in scenarios where both the cloud server's database and the secret key are compromised. In this comment, we demonstrate that Zhang et al.'s protocol fails to withstand KCI attacks and is further susceptible to session-specific temporary random number leakage attack. These findings highlight that, despite continuous research efforts over the years, KCI remains a persistent challenge in the design of user authentication protocols, underscoring the urgent need for future schemes that are immune to KCI.

Keywords: authentication, key compromise impersonation, KCI attacks, Internet of Drones.

1 Introduction

The Internet of Drones (IoD) represents a cutting-edge paradigm that integrates unmanned aerial vehicles (UAVs) with Artificial Intelligence (AI) and next-generation communication networks to enable intelligent, human-centric services. Driven by rapid technological advancements, drones are now being deployed across diverse domains, including agriculture, defence, disaster management, environmental monitoring, and smart cities, to facilitate real-time surveillance, data exchange, and seamless human-drone collaboration.

Recently, Zahednejad and Gao [1] (2025) showed that Key Compromise Impersonation (KCI) attack can be launched on authentication protocols in different ways.

Submitted: 06 May 2026
Accepted: 12 June 2026
Published: 19 June 2026

Vol. 2, **No.** 2, 2026.
10.62762/JRSC.2026.503368

***Corresponding author:**
✉ Akash Kumar
akash79827@gmail.com

Citation
Kumar, A., & Kumari, S. (2026). Comments on *CSAP-IoD: A Chaotic Map-Based Secure Authentication Protocol for Internet of Drones*. *Journal of Reliable and Secure Computing*, 2(2), 156–160.

They revealed that Qiu et al.'s (2020) protocol [2] for mobile devices suffers from KCI attacks when both the cloud server's secret key and session nonce are compromised. Similarly, they also showed that He et al.'s (2023) protocol [3] for IoT environments becomes vulnerable to KCI attacks upon exposure of the cloud server's secret key.

We recall Kumari et al.'s (2019) work [4], which questioned the legitimacy of KCI attacks. Nevertheless, assuming KCI to be a valid attack, their analysis concluded that compromising a server's secret key could expose authentication protocols to various severe security risks. They emphasized the urgent need for developing practical countermeasures to mitigate KCI threats.

More recently, Zhang et al. [5] (2025) proposed a chaotic map-based authentication protocol for IoD and claimed its robust security properties. We observe that the protocol proposed by Zahednejad and Gao [1] exhibits strong resilience against KCI attacks, even under conditions where both the cloud server's database and its secret key are exposed. However, in this comment, we demonstrate that Zhang et al.'s protocol succumbs to KCI attacks and session-specific temporary random number leakage attack.

2 Notation and Review of Zhang et al.'s [5] Protocol

1. **Notation**

   Table 1 describes Notation.

2. **Review of Zhang et al.'s [5] Protocol**

   Tables 2 and 3 briefly display the registration and login/authentication phases of Zhang et al.'s [5] protocol.

Table 1. Notations.

Notation	Meaning
U_i, D_j, CC	ith user, jth drone, Control Center
$ID_i/ID_j/ID_C$	The real identity of $U_i/D_j/CC$
PID_i/PID_j	The pseudo identity of U_i/D_j
CID_i	The provisional identity of U_i
PW_i	The password of U_i
BIO_i	The biometric of U_i
$T_n(\cdot)$	Chebyshev polynomial
$h(\cdot)$	Hash function
$Gen(\cdot)/Rep(\cdot)$	Generate/Reconstruct in fuzzy extractor
$E_s(\cdot)/D_s(\cdot)$	Symmetric Encryption/Decryption
$r_1, r_2, r_3, r_4, r_5, r_2^*$	Random numbers
$\tau_1/\tau_2/\tau_3$	Timestamps
SK	Session key
A^T	Attacker
s	secret key of CC

3 Comment on Zhang et al.'s [5] Protocol

In this section, we demonstrate the KCI attack and session-specific temporary random number leakage attack on Zhang et al.'s [5] protocol. Table 4 is a brief (pictorial) expression of the KCI attack.

3.1 Key Compromise Impersonation Attack

Assume that the master key s of CC is compromised. Then an attacker A^T can impersonate a legitimate user of the system as described below.

Step-1 at A^T: A^T can compute $n_s^* = h(ID_c||s)$. A^T has $M1 = \{CID_i, C_1, C_3, C_4, \tau_1\}$ intercepted from the public channel. A^T decrypts $D_{T_{n_s^*}}(CID_i) = ID_i||n_s||r_1$ to obtain $< ID_i, n_s, r_1 >$ where ID_i is the identity of the user U_i and $r_1 \in Z_q^*$ is the random number generated by CC during system set up phase for user and drone registration purpose. A^T verifies if $n_s^* = n_s$. This verification builds A^T's confidence to proceed with the attack. Knowing $r_1 \& ID_i$, the attacker A^T computes $PID_i = T_{(r_1 ID_i)}(x)$ which is the pseudo identity of U_i. So far A^T has the following credentials: $\{ID_i, r_1, PID_i, n_s(= h(ID_c||s))\}$. These credentials are sufficient for A^T to impersonate U_i.

Step-2 at A^T: A^T randomly generates $r_2^{A^T} \in Z_q^*$ to compute $C_1^{A^T} = T_{r_2^{A^T}}(x), C_2^{A^T} = T_{r_2^{A^T}}(PID_i), C_3^{A^T} = PID_j \oplus h(C_2^{A^T}||h((PID_i)||n_s||r_1))$, and

$$C_4^{A^T} = h(ID_i||PID_j||C_2^{A^T}||C_3^{A^T}||\tau_1^{A^T})$$

where PID_j is the pseudo-identity of the jth drone and $(\tau_1^{A^T})$ is the current timestamp chosen at the attacker's side. A^T sends $M1^{A^T} = \{CID_i, C_1^{A^T}, C_3^{A^T}, C_4^{A^T}, \tau_1^{A^T}\}$ to CC.

Step-3 at CC: As CC receives $M1^{A^T}$, it first checks freshness of $\tau_1^{A^T}$. For fresh $\tau_1^{A^T}$, the control center CC decrypts CID_i as $D_{T_{n_s}(x)}(CID_i) = (ID_i||n_s||r_1)$, computes $PID_i = T_{(r_1 ID_i)}(x)$, $C_2^{A^{T^*}} = T_{(r_1 ID_i)}(C_1^{A^T})$, $Y_i = h(PID_i||n_s||r_1)$, $PID_j = C_3^{A^T} \oplus h(C_2^{A^{T^*}}||Y_i)$, and $C_4^{A^{T^*}} = h(ID_i||PID_j||C_2^{A^{T^*}}||C_3^{A^T}||\tau_1^{A^T})$. Then CC verifies if $C_4^{A^{T^*}} \stackrel{?}{=} C_4^{A^T}$. Success of the verification authenticates A^T to CC as the legitimate user U_i.

Step-4 at CC: As a consequence of step-3, CC computes the message $M2 = \{C_6, C_7, \tau_2\}$ exactly the same way as it does for the legitimate user U_i and sends $M2$ to D_j. Here τ_2 is freshly chosen timestamp at CC.

Step-5 at D_j: The drone D_j checks the validity of $M2$ which will hold similarly as for a legitimate user. D_j

Table 2. Registration Phase of Zhang et al.'s [5] Protocol.

User(U_i)	**Control Center**(CC)
Chooses ID_i, PW_i	
Imprints BIO_i	
Computes $(\sigma_1, \sigma_2) = Gen(BIO_i)$	Generates $r_1 \in Z_q^*$
Computes $X_i = h(ID_i \| PW_i \| \sigma_1)$	Computes $n_s = h(ID_c \| s)$
$\xrightarrow{\{M_1 = ID_i, X_i\}}$	$PID_i = T_{(r_1 ID_i)}(x)$
	$Y_i = h(PID_i \| n_s \| r_1)$
$Z_i^* = Z_i \oplus PID_i$	$Z_i = X_i \oplus Y_i \oplus PID_i$
$W_i = h(h(PID_i \| X_i \oplus Z_i^* \| \sigma_1) \bmod n)$	$SC_i = \{PID_i, CID_i, Z_i\}$
Updates Smart card	$CID_i = E_{T_{n_s}(x)}(ID_i \| n_s \| r_1)$
$SC_i = \{PID_i, CID_i, Z_i^*, W_i, \sigma_2\}$	$\xleftarrow{\{M_2 = \{PID_i, CID_i, Z_i\}}$

Table 3. Login and Authentication Phase of Zhang et al.'s [5].

User (U_i)	**Control Center (CC)**	**Drone** (D_j)
Inputs ID_i, PW_i		
Imprints BIO_i		
Computes $\sigma_1 = Rep(BIO_i, \sigma_2)$	Check $\|\tau_1' - \tau_1\| < \triangle T$	
Computes $X_i = h(ID_i \| PW_i \| \sigma_1)$	$(ID_i \| n_s \| r_1) = D_{T_{n_s}}(x)(CID_i)$	
$W_i^* = h(h(PID_i \| X_i \oplus Z_i^* \| \sigma_1) \bmod n)$	$PID_i = T_{(r_1 ID_i)}(x)$	
Verifies $W_i^* \stackrel{?}{=} W_i$	$C_2^* = T_{(r_1 ID_i)}(C_1)$	
Generates $r_2 \in Z_q^*$ and timestamp τ_1	$Y_i = h(PID_i \| n_s \| r_1)$	
Computes: $C_1 = T_{r_2}(x), C_2 = T_{r_2}(PID_i)$	$PID_j = C_3 \oplus h(C_2^* \| Y_i)$	
$C_3 = PID_j \oplus h(C_2 \| X_i \oplus Z_i^*)$	$C_4^* = h(ID_i \| PID_j \| C_2^* \| C_3 \| \tau_1)$	Check $\|\tau_2' - \tau_2\| < \triangle T$
$C_4 = h(ID_i \| PID_j \| C_2 \| C_3 \| \tau_1)$	Verifies $C_4^* \stackrel{?}{=} C_4$	Computes
$\xrightarrow{M1 = \{CID_i, C_1, C_3, C_4, \tau_1\}}$	Generates $r_3 \in Z_q^*$ and timestamp τ_2	$(PID_j^{new} \| r_3 \| C_1) = C_6 \oplus h(PID_j \| \tau_2)$
	Computes $PID_j^{new} = T_{(r_3 ID_j)}(x)$	$C_5^* = T_{r_3}(PID_j)$
	$C_5 = T_{r_1}(PID_j^{new})$	$C_7^* = h(ID_i \| PID_j^{new} \| C_5^* \| C_6 \| \tau_2)$
	$C_6 = (PID_j^{new} \| r_3 \| C_1) \oplus h(PID_j \| \tau_2)$	Verifies $C_7^* \stackrel{?}{=} C_7$
	$C_7 = h(ID_i \| PID_j^{new} \| C_5 \| C_6 \| \tau_2)$	Generates $r_4, r_5 \in Z_q^*$ and
	$\xrightarrow{M2 = \{C_6, C_7, \tau_2\}}$	timestamp τ_3
		Computes $C_8 = T_{r_4}(x), C_9 = T_{r_4}(C_1)$
Check $\|\tau_3' - \tau_3\| < \triangle T$		$C_{10} = r_5 \oplus C_9$
Computes $C_9^* = T_{r_2}(C_8), r_5^* = C_{10} \oplus C_9^*$		$SK = h(PID_j \| C_9 \| r_5 \| \tau_3)$
$SK = h(PID_j \| C_9^* \| r_5^* \| \tau_3)$		$C_{11} = h(SK \| C_9 \| r_5 \| \tau_3)$
$C_{11}^* = h(SK \| C_9^* \| r_5^* \| \tau_3)$		$\xleftarrow{M3 = \{C_8, C_{10}, C_{11}, \tau_3\}}$
Verifies $C_{11}^* \stackrel{?}{=} C_{11}$		

computes the message $M3 = \{C_8, C_{10}, C_{11}, \tau_3\}$ and sends it. Here τ_3 is freshly chosen timestamp at D_j.

Step-6 at U_i: On receiving $M3$ with fresh timestamp τ_3, the attacker A^T computes $C_9^{A^T} = T_{r_2^{A^T}}(C_8)$, $r_5^{A^T} = C_{10} \oplus C_9^{A^T}$ exactly the same way as U_i does. Then A^T uses $C_9^{A^T}$ and $r_5^{A^T}$ to compute the session key $SK^{A^T} = h(PID_j \| C_9^{A^T} \| r_5^{A^T} \| \tau_3)$, and $C_{11}^{A^T} = h(SK^{A^T} \| C_9^{A^T} \| r_5^{A^T} \| \tau_3)$. Then A^T verifies if $C_{11}^{A^T} \stackrel{?}{=} C_{11}$. If so, it validates the correctness of the session key SK^{A^T}.

In this way, the attacker can authenticate itself and establish a session key with D_j and hence successfully impersonate the legitimate user U_i without knowing the user's password or biometrics.

Session Specific Temporary Random Number Leakage Attack

During the authentication phase, user U_i uses the pseudo identity PID_j of jth drone D_j to compute $C_3 = PID_j \oplus h(C_2 \| X_i \oplus Z_i^*)$. However, CC never provides PID_j to U_i. Thus, PID_j is a public

Table 4. Demonstration of Key Compromise Impersonation Attack on Zhang et al.'s [5] Protocol.

Attacker ($\mathcal{A}$)	**Control Center (CC)**	**Drone** (D_j)
Computes $n_s^* = h(ID_c\|\|s)$		
Has $M1 = \{CID_i, C_1, C_3, C_4, \tau_1\}$		
Decrypts $T_{n_s(x)}(CID_i) = (ID_i\|\|n_s\|\|r_1)$	Check $\|\tau_1' - \tau_1^{\mathcal{A}}\| < \Delta T$	
Verifies $n_s^* \stackrel{?}{=} n_s$	$(ID_i\|\|n_s\|\|r_1) = D_{T_{n_s}}(x)(CID_i)$	
Computes $PID_i = T_{(r_1 ID_i)}(x)$	$PID_i = T_{(r_1 ID_i)}(x)$	
Randomly generate $r_2^{\mathcal{A}} \in Z_q^*$ and timestamp $\tau_1^{\mathcal{A}}$	$C_2^{\mathcal{A}^*} = T_{(r_1 ID_i)}(C_1^{\mathcal{A}})$	
Computes: $C_1^{\mathcal{A}} = T_{r_2^{\mathcal{A}}}(x), C_2^{\mathcal{A}} = T_{r_2^{\mathcal{A}}}(PID_i)$	$Y_i = h(PID_i\|\|n_s\|\|r_1)$	
$C_3^{\mathcal{A}} = PID_j \oplus h(C_2^{\mathcal{A}}\|\|h(PID_i\|\|n_s\|\|r_1))$	$PID_j = C_3^{\mathcal{A}} \oplus h(C_2^{\mathcal{A}^*}\|\|Y_i)$	
$C_4^{\mathcal{A}} = h(ID_i\|\|PID_j\|\|C_2^{\mathcal{A}}\|\|C_3^{\mathcal{A}}\|\|\tau_1^{\mathcal{A}})$	$C_4^{\mathcal{A}^*} = h(ID_i\|\|PID_j\|\|C_2^{\mathcal{A}^*}\|\|C_3^{\mathcal{A}}\|\|\tau_1^{\mathcal{A}})$	Check $\|\tau_2' - \tau_2\| < \Delta T$
$\xrightarrow{M1^{\mathcal{A}}=\{CID_i, C_1^{\mathcal{A}}, C_3^{\mathcal{A}}, C_4^{\mathcal{A}}, \tau_1^{\mathcal{A}}\}}$	Verifies $C_4^{\mathcal{A}^*} \stackrel{?}{=} C_4^{\mathcal{A}}$	Computes $(PID_j^{new}\|\|r_3\|\|C_1) = C_6 \oplus h(PID_j\|\|\tau_2)$
	Generates $r_3 \in Z_q^*$ and timestamp τ_2	$C_5^* = T_{r_3}(PID_j)$
	Computes $PID_j^{new} = T_{(r_3 ID_j)}(x)$	$C_7^* = h(ID_i\|\|PID_j^{new}\|\|C_5^*\|\|C_6\|\|\tau_2)$
	$C_5 = T_{r_1}(PID_j^{new})$	Verifies $C_7^* \stackrel{?}{=} C_7$
	$C_6 = (PID_j^{new}\|\|r_3\|\|C_1) \oplus h(PID_j\|\|\tau_2)$	Generates $r_4, r_5 \in Z_q^*$ and timestamp τ_3
	$C_7 = h(ID_i\|\|PID_j^{new}\|\|C_5\|\|C_6\|\|\tau_2)$	Computes $C_8 = T_{r_4}(x), C_9 = T_{r_4}(C_1)$
	$\xrightarrow{M2=\{C_6, C_7, \tau_2\}}$	$C_{10} = r_5 \oplus C_9$
Check $\|\tau_3' - \tau_3\| < \Delta T$		$SK = h(PID_j\|\|C_9\|\|r_5\|\|\tau_3)$
Computes $C_9^{\mathcal{A}} = T_{r_2^{\mathcal{A}}}(C_8), r_5^{\mathcal{A}} = C_{10} \oplus C_9^{\mathcal{A}}$		$C_{11} = h(SK\|\|C_9\|\|r_5\|\|\tau_3)$
$SK^{\mathcal{A}} = h(PID_j\|\|C_9^{\mathcal{A}}\|\|r_5^{\mathcal{A}}\|\|\tau_3)$		$\xleftarrow{M3=\{C_8, C_{10}, C_{11}, \tau_3\}}$
$C_{11}^{\mathcal{A}} = h(SK\|\|C_9^{\mathcal{A}}\|\|r_5^{\mathcal{A}}\|\|\tau_3)$		
Verifies $C_{11}^{\mathcal{A}} \stackrel{?}{=} C_{11}$		

parameter. Suppose the random number r_4 is leaked, and $\mathcal{A}$ has $M1 = \{CID_i, C_1, C_3, C_4, \tau_1\}$ and $M3 = \{C_8, C_{10}, C_{11}, \tau_3\}$ intercepted from public channel. Then $\mathcal{A}$ uses r_4 to compute $C_9^{\mathcal{A}} = T_{r_4}(C_1)$ exactly the same way as D_j does, and obtains $r_5^{\mathcal{A}} = C_{10} \oplus C_9^{\mathcal{A}}$ exactly the same way as U_i does. Attacker $\mathcal{A}$ uses $C_9^{\mathcal{A}}$ and $r_5^{\mathcal{A}}$ to compute the session key $SK^{\mathcal{A}} = h(PID_j||C_9^{\mathcal{A}}||r_5^{\mathcal{A}}||\tau_3)$ and henceforth it can read all the messages of the session which the random number belongs to. Therefore, Zhang et al.'s protocol is not secure against session-specific temporary random number leakage attack.

4 Findings and Conclusion

In this paper, we demonstrate that Zhang et al.'s authentication protocol for the Internet of Drones is vulnerable to key compromise impersonation (KCI) attack and session-specific temporary random number leakage attack. However, Zahednejad and Gao's authentication protocol for IoT devices does resist the KCI attack. The control center and the cloud server play the roles of the server in Zhang et al.'s and Zahednejad and Gao's protocols, respectively.

Although both protocols utilise the concept of chaotic maps to achieve security, one falls prey to KCI while the other stands out as a robust authentication framework. We have analyzed the reason for the different behaviors of these two protocol designs towards KCI. In Zhang et al.'s protocol, the keys used for chaotic map operation are either random numbers or a combination of a session-specific random number with the identity of some entity, or a combination of an entity's identity and the secret key of the control center protected with a one-way hash function. Generally, identities are either not hidden or are easily guessed. Therefore, in Zhang et al.'s protocol, an attacker possessing the master secret key of the control server can obtain the keys required for chaotic map operations involved in computing the session key. In Zahednejad and Gao's protocol, the keys used for chaotic map operation differ from those in Zhang et al.'s protocol in the sense that, instead of using the identity of the user, they have used a secret key of the user, which is accessible only to the user itself and is not vulnerable to guessing. Therefore, an attacker cannot apply KCI attacks on Zahednejad and Gao's protocol.

Kumari et al.'s paper published in 2019 questioned the validity of the KCI attack and also highlighted its impact on the authentication and key agreement protocols. Even after years of extensive research to counter KCI attacks and their countermeasures, the KCI attack is still applicable in newly designed protocols. Zhang et al.'s protocol is susceptible, while Zahednejad and Gao's protocol is resistant to KCI attack under the same hard problem deployed to impart security to their protocols. The message is that it is not only the hard problem, but also the manner of using the hard problem which affects the security of the designed authentication protocol. The issue of KCI attack continues to demand special

attention while designing authentication and key agreement frameworks. It becomes very important for researchers to look at the design of their newly proposed protocols from various perspectives in order to truly and effectively mitigate the KCI attack. The gist is that the researchers should be attentive towards the KCI attack and its remedies so that the future protocols become free from KCI vulnerability.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

Saru Kumari served as a Co-Editor-in-Chief of the *Journal of Reliable and Secure Computing* at the time of manuscript submission. To ensure the integrity of the peer-review process, Saru Kumari was not involved in the editorial handling, peer review, or decision-making process for this manuscript, which was handled independently by another editor. The remaining authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Zahednejad, B., & Gao, C. Z. (2025). Mitigating server key compromise impersonation: a secure and efficient authentication and key agreement protocol for IoT devices using chaotic maps. *Journal of Information Security and Applications*, *92*, 104083. [CrossRef]

[2] Qiu, S., Wang, D., Xu, G., & Kumari, S. (2020). Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 1338-1351. [CrossRef]

[3] He, D., Cai, Y., Zhu, S., Zhao, Z., Chan, S., & Guizani, M. (2023). A lightweight authentication and key exchange protocol with anonymity for IoT. *IEEE Transactions on Wireless Communications*, *22*(11), 7862-7872. [CrossRef]

[4] Kumari, S., Chaudhary, P., Chen, C. M., & Khan, M. K. (2019). Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications. *IEEE Access*, *7*, 39717-39720. [CrossRef]

[5] Zhang, J., Cheng, Q., Chen, X., & Luo, X. (2025). CSAP-IoD: A Chaotic Map-Based Secure Authentication Protocol for Internet of Drones. *IEEE Transactions on Information Forensics and Security*, *20*, 8848-8862. [CrossRef]

Saru Kumari , Assoc. Prof., Dept. of Maths, CCS University, Meerut, U.P., India. She received her PhD in Maths in 2012 from the same University. She is the recipient of the India Research Excellence-Citation Awards: Women in Research 2023, presented by Clarivate Analytics. She has published more than 380 research papers in reputable international journals and conferences, including over 330 research papers in various SCIE-indexed journals. Her research interests encompass security, Applied Cryptography, Blockchain Technology, and Information Security. (Email: saryusiirohi@gmail.com)

Akash Kumar is a research scholar, Dept. of Maths, CCS University, Meerut, U.P., India. He is enthusiastic and hard-working, and a keen learner. His research interests include Authentication and Blockchain Technology. (Email: akash79827@gmail.com)