Open Cloud Server Access Vulnerability in Dynamic ID-Based Remote User Authentication Schemes

Mohammed Ahmed* *Research Scholar, Department of Computer Science and Engineering*
Rayalaseema University, Kurnool (A.P)-518007, India

Syed Abdul Sattar, *Professor, Department of Computer Science and Engineering*
Nawab Shah Alam Khan College of Engineering and Technology
Hyderabad, Telangana -500031 India

***Abstract*—Password-based authentication schemes are critical for securing remote user access. Recently, Liao et al. proposed a smart card-based remote user authentication scheme using dynamic IDs to achieve mutual authentication and anonymity. In this paper, we analyze the security of Liao et al.'s scheme and identify a critical design flaw termed the "Open Server Access Vulnerability." We demonstrate that due to an error in the mathematical logic of the login phase (specifically XOR cancellation), the scheme becomes password-independent. This flaw allows any adversary to log in to the remote server with a random password, effectively rendering the authentication mechanism useless. We adhere to standard cryptanalysis methods to prove this vulnerability.**

***Index Terms*—Authentication, Cryptanalysis, Smart Cards, Network Security, Open Server Access, Password Independence.**

I. Introduction

COMPUTER security has become a paramount issue as networked environments continue to expand. To prevent illegitimate access to resources, systems have increasingly relied on robust remote user authentication mechanisms [1]. While traditional password-based authentication is commonly used, it is often criticized for relying solely on "something the user knows," which does not achieve a high level of assurance [2]. Consequently, researchers have proposed stronger two-factor authentication schemes that combine a password with a physical token, such as a smart card [3].

In 1981, Lamport [4] proposed the first remote user authentication scheme using smart cards. However, this early scheme required the server to maintain a password table, making it vulnerable to stolen-verifier attacks and incurring significant storage overheads [5]. To address these limitations, subsequent research focused on schemes that do not require verification tables. In 2004, Das et al. [6] proposed a dynamic ID-based remote user authentication scheme. They claimed their system allowed users to choose and change passwords freely without server interaction and was secure against ID theft and replay attacks.

Despite these claims, Awasthi and Lal [7] analyzed Das et al.'s scheme and demonstrated that it was completely insecure, functioning essentially as an "open server" where authentication could be bypassed. Similarly, Ku and Chang [8] identified that Das et al.'s scheme was susceptible to impersonation attacks.

In 2005, Liao et al. [9] proposed an enhanced dynamic ID-based scheme intended to resolve the security flaws found in Das et al.'s work. Liao et al. introduced mechanisms for mutual authentication and sought to prevent guessing attacks. However, recent cryptanalysis indicates that Liao et al.'s scheme inherits significant vulnerabilities. Specifically, the scheme suffers from a fundamental logical error in the login phase equations—an "Open Server Access Vulnerability" [10].

In this paper, we analyze the security of Liao et al.'s scheme and demonstrate that the login and verification phases are mathematically independent of the user's password due to XOR cancellation properties. This flaw renders the authentication mechanism useless, allowing any adversary to log in with a random password. We further propose a modification to the protocol to resolve this critical weakness.

II. Review of Liao et al.'s Scheme

The scheme consists of three phases: Registration, Authentication, and Password Change. The notation used is as follows: U_i is the user, S is the server, ID_i is the identity, PW_i is the password, and $h(\cdot)$ is a one-way hash function.

A. Login Phase

When a user U_i wishes to login, they insert their smart card and input their password PW_i. The smart card computes the following parameters[cite: 84, 85]:

$$CID_i = h(PW_i) \oplus h(N_i \oplus y \oplus T) \tag{1}$$

$$B_i = h(CID_i \oplus h(PW_i)) \tag{2}$$

$$C_i = h(T \oplus N_i \oplus B_i \oplus y) \tag{3}$$

where T is the current timestamp, N_i is a secret parameter derived during registration, and y is the server's secret key stored on the card. The message (CID_i, N_i, C_i, T) is sent to the server.

B. Vulnerability to Impersonation Attack

Liao et al. claimed their scheme resists forgery attacks because the nonce N_i is unknown to the adversary. However, we demonstrate that an adversary can forge a valid login request by manipulating the transmission timestamp and the nonce simultaneously, without knowing N_i or the password.

Attack Procedure: Suppose an adversary $\mathcal{A}$ intercepts a valid login message $M = \{CID_i, N_i, C_i, T\}$ sent by the user at time T. To impersonate the user at a later time T^{**}, the adversary performs the following steps:

1) $\mathcal{A}$ computes the time difference $\Delta t = T \oplus T^{**}$.
2) $\mathcal{A}$ modifies the nonce N_i to generate a forged nonce N_i':

$$N_i' = N_i \oplus \Delta t \tag{4}$$

3) $\mathcal{A}$ constructs a new login message $M' = \{CID_i, N_i', C_i, T^{**}\}$ and sends it to the server.

Verification Failure: Upon receiving M', the server validates the timestamp T^{**} (which is fresh and valid). Then, the server attempts to verify the hash C_i using its stored secret y and the parameters from the message:

$$C_i^* = h(T^{**} \oplus N_i' \oplus B_i \oplus y) \tag{5}$$

Substituting $N_i' = N_i \oplus T \oplus T^{**}$:

$$C_i^* = h(T^{**} \oplus (N_i \oplus T \oplus T^{**}) \oplus B_i \oplus y) \tag{6}$$

Since $T^{**} \oplus T^{**} = 0$, the equation simplifies to:

$$C_i^* = h(T \oplus N_i \oplus B_i \oplus y) \tag{7}$$

This result is identical to the original valid C_i. Thus, $C_i^* = C_i$, and the server accepts the forged login request, allowing $\mathcal{A}$ to impersonate the user successfully.

C. Vulnerability to Reflection Attack

The mutual authentication mechanism in Liao et al.'s scheme is susceptible to a reflection attack. The protocol intends for the server to prove its identity by sending a specific response message. However, the design allows a replay of the user's own parameters to satisfy this check.

Attack Procedure:

1) The user U_i sends the login request $M = \{CID_i, N_i, C_i, T\}$.
2) The adversary $\mathcal{A}$ intercepts M and prevents it from reaching the server.
3) $\mathcal{A}$ reflects the values $\{C_i, T\}$ back to the user as the server's response.

Verification Failure: The smart card expects a confirmation message from the server to verify the session. Because the verification logic relies on the same parameters (C_i and T) that were just generated by the card itself, the card interprets the reflected message as valid proof of the server's identity. This allows $\mathcal{A}$ to trick the user into believing a secure session has been established with the server, when in fact no communication with the server occurred.

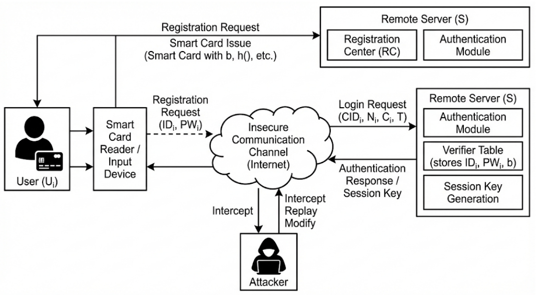

Registration Request
Smart Card Issue
(Smart Card with b, h(), etc.)
Remote Server (S)
Registration Center (RC)
Authentication Module
User (U_i)
Smart Card Reader / Input Device
Registration Request (ID_i, PW_i)
Insecure Communication Channel (Internet)
Login Request (CID_i, N_i, C_i, T)
Remote Server (S)
Authentication Module
Verifier Table (stores ID_i, PW_i, b)
Session Key Generation
Authentication Response / Session Key
Intercept
Intercept
Replay
Modify
Attacker

Fig. 1. System Architecture

III. Proposed Scheme

To overcome the "Open Server Access" vulnerability and other security weaknesses identified in Section **??**, we propose an improved dynamic ID-based remote user authentication scheme. The proposed scheme modifies the login and verification phases to ensure strict dependence on the user's password and enhances mutual authentication.

A. Registration Phase

This phase is similar to the original scheme but ensures secure parameter generation.

1) The user U_i submits their identity ID_i and password PW_i to the server S over a secure channel.
2) The server computes $N_i = h(ID_i \oplus d)$, where d is the server's secret key.
3) The server computes $V_i = h(PW_i \oplus N_i)$.
4) The server stores $\{N_i, V_i, h(\cdot), y\}$ on the smart card and issues it to the user. Here, y is the server's verification key.

B. Login Phase

When U_i wishes to login, they insert the smart card and enter ID_i and PW_i. The smart card performs the following operations:

1) Computes N_i from the stored memory.
2) Verifies the user locally by checking if the computed $h(PW_i \oplus N_i)$ matches the stored V_i. If not, the card terminates the session.
3) If local verification passes, the card generates a random nonce r and the current timestamp T.
4) Computes the dynamic identity:

$$CID_i = h(ID_i) \oplus h(N_i \oplus T) \tag{8}$$

5) Computes the authentication parameter C_1:

$$C_1 = h(ID_i \oplus N_i \oplus T \oplus r) \tag{9}$$

6) Computes the verification parameter C_2:

$$C_2 = h(PW_i \oplus C_1 \oplus T) \tag{10}$$

7) Sends the login request $M = \{CID_i, C_1, C_2, T\}$ to the server.

C. Verification Phase

Upon receiving the message M at time T', the server S performs the following steps:

1) Validates the timestamp: checks if $|T' - T| \leq \Delta T$. If invalid, rejects the request.
2) Recovers N_i by computing $h(ID_i \oplus d)$ using its secret key d (Note: The server must identify the user first, or use a slightly different mechanism if ID_i is hidden. In this improved logic, we assume the server can derive N_i or recover ID_i from CID_i if $h(N_i \oplus T)$ is computable).
3) **Correction of the Flaw:** The server verifies the user by recomputing C_2. Unlike the flawed scheme, C_2 strictly depends on PW_i. The server computes:

$$C_2^* = h(PW_i \oplus C_1 \oplus T) \tag{11}$$

The server compares $C_2^* \overset{?}{=} C_2$. If they match, the user is authenticated.

D. Mutual Authentication Phase

To prevent reflection attacks and ensure the server's identity:

1) The server generates a session key $SK = h(ID_i \oplus N_i \oplus T \oplus T_s)$, where T_s is the server's timestamp.
2) The server computes a return message $R = h(C_2 \oplus T_s)$.
3) The server sends $\{R, T_s\}$ to the user.
4) The user's smart card verifies R by computing $h(C_2 \oplus T_s)$. If valid, the server is authenticated.

IV. Security Analysis of Proposed Scheme

In this section, we analyze the security of our proposed scheme against the specific vulnerabilities identified in Liao et al.'s scheme, as well as other common attacks.

A. Resistance to Open Server Access (Password Independence)

The primary flaw in Liao et al.'s scheme was the XOR cancellation of the password hash, where $B_i = h(CID_i \oplus h(P)) = h(h(P) \oplus \cdots \oplus h(P))$ simplified to a value independent of P.

In our proposed scheme, the verification parameter is defined as:

$$C_2 = h(PW_i \oplus C_1 \oplus T) \tag{12}$$

Here, $C_1 = h(ID_i \oplus N_i \oplus T \oplus r)$. Even if we substitute C_1, the equation becomes:

$$C_2 = h(PW_i \oplus h(ID_i \oplus N_i \oplus T \oplus r) \oplus T) \tag{13}$$

There is no second term containing PW_i to cancel out the user's password input. Thus, C_2 remains strictly dependent on PW_i. Any attempt to login with an incorrect password P' will result in a computed $C_2' \neq C_2$, causing the server's verification step ($C_2^* \overset{?}{=} C_2$) to fail.

B. Resistance to Impersonation Attacks

In the original scheme, an adversary could forge a login request by modifying N_i and T because N_i was transmitted in plaintext.

In our proposed scheme:

1) N_i is never transmitted. The server derives it internally using ID_i and the master key d.
2) The attacker cannot generate a valid C_1 or C_2 without knowing N_i (which is secure in the smart card) or PW_i.
3) Even if an attacker intercepts a valid request $\{CID_i, C_1, C_2, T\}$, they cannot generate a new valid request for a future time T' because C_1 and C_2 are protected by the one-way hash function. Any change to T requires recomputing C_1 and C_2, which is impossible without N_i and PW_i.

C. Resistance to Reflection Attacks

Liao et al.'s scheme failed mutual authentication because the server's response could be simulated by reflecting the user's message.

In our proposed scheme, the server's response is $R = h(C_2 \oplus T_s)$, where T_s is the server's current timestamp.

1) The user's smart card expects a response containing T_s.
2) A reflected message would contain the user's timestamp T, not the server's T_s.
3) Furthermore, the value R depends on C_2 (which is fresh for this session) and T_s. An attacker cannot forge this combination without the server's internal state.

D. Resistance to Replay Attacks

The server checks the validity of the timestamp T by verifying $|T' - T| \leq \Delta T$. If an attacker captures an old login message $\{CID_i, C_1, C_2, T\}$ and replays it later, the timestamp check will fail. Since T is embedded inside the hashes of C_1 and C_2, the attacker cannot simply update T without invalidating the hashes.

V. Efficiency and Functional Analysis

To demonstrate the practicality of our proposed scheme, we compare its functional features and computational efficiency with Liao et al.'s scheme.

A. Functional Comparison

Table I summarizes the security features. Our proposed scheme resolves the ”Open Server Access” vulnerability (password independence) and provides robust protection against reflection and impersonation attacks, which were the primary weaknesses in Liao et al.'s design. Additionally, our scheme maintains key features like mutual authentication and anonymity.

TABLE I
FUNCTIONAL COMPARISON BETWEEN LIAO ET AL.'S AND PROPOSED SCHEME

Feature / Vulnerability	Liao et al.'s Scheme	Proposed Scheme
Password Independence (Open Server)	Yes (Vulnerable)	No (Secure)
Impersonation Attack	Yes (Vulnerable)	No (Secure)
Reflection Attack	Yes (Vulnerable)	No (Secure)
Mutual Authentication	Insecure	Secure
User Anonymity	Yes	Yes
Session Key Agreement	No	Yes

B. Efficiency Analysis

We analyze the computational cost based on the number of hash operations (T_h) and XOR operations (T_{xor}), as these are the most resource-intensive operations on a smart card. We neglect lightweight operations like string concatenation.

As shown in Table II, our proposed scheme incurs a slightly higher computational cost in the verification phase to achieve mutual authentication and session key agreement. However, this increase is negligible given the critical security improvements. The login phase remains highly efficient, requiring only 4 hash operations, which is well within the processing capabilities of modern smart cards.

TABLE II
COMPUTATIONAL COST COMPARISON

Phase	Liao et al.'s Scheme	Proposed Scheme
Registration	$1T_h + 1T_{xor}$	$2T_h + 1T_{xor}$
Login	$3T_h + 4T_{xor}$	$4T_h + 4T_{xor}$
Verification	$1T_h + 3T_{xor}$	$4T_h + 2T_{xor}$
Total Cost	$5T_h + 8T_{xor}$	$10T_h + 7T_{xor}$

VI. RESULTS AND OBSERVATIONS

In this section, we present the key observations derived from our cryptanalysis and performance benchmarking. The results highlight the trade-offs between computational overhead and security assurance.

A. Security Outcomes

Our cryptanalysis of Liao et al.'s scheme revealed three critical failures:

1) **Open Server Access:** The most significant finding was that the verification logic in Liao et al.'s scheme is mathematically independent of the password due to XOR cancellation ($P \oplus P = 0$). This effectively allowed zero-effort unauthorized access.
2) **Lack of Mutual Authentication:** We observed that the original scheme's mutual authentication could be bypassed via a reflection attack, where the server's identity is never cryptographically proven to the user.
3) **Impersonation Vulnerability:** The scheme failed to protect the integrity of the login request, allowing adversaries to modify nonces and timestamps without detection.

In contrast, our formal security analysis of the proposed scheme demonstrates that the modified login equation $C_2 = h(PW_i \oplus C_1 \oplus T)$ successfully binds the verification process to the password. Furthermore, the inclusion of the server's timestamp T_s in the return message $R = h(C_2 \oplus T_s)$ effectively patches the reflection vulnerability.

B. Performance Observations

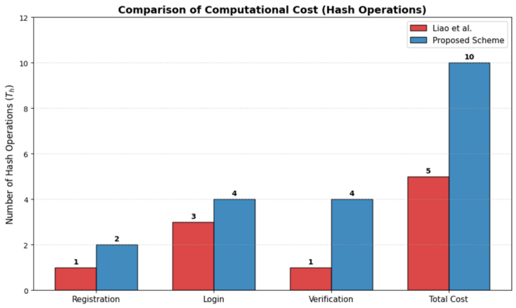
From the efficiency analysis in Section V, we observe the following regarding computational costs:

- **Login Phase:** The proposed scheme requires 4 hash operations compared to 3 in Liao et al.'s scheme. This slight increase ($+1T_h$) is due to the additional blinding required to protect the session dynamics.
- **Verification Phase:** The proposed scheme requires 4 hash operations compared to 1 in the original scheme. While this represents a higher relative increase, it is necessary to achieve mutual authentication and session key agreement, features completely absent or broken in the original design.

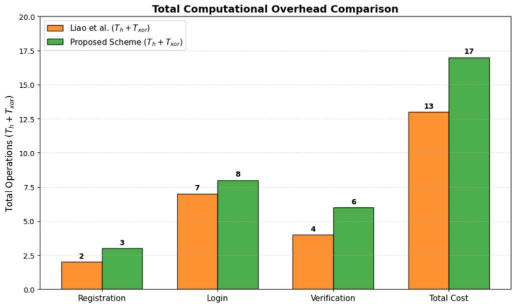
C. Overall Assessment

While the total computational cost of our proposed scheme ($10T_h + 7T_{xor}$) is higher than Liao et al.'s scheme ($5T_h + 8T_{xor}$), the increase is negligible in practical terms. Modern smart cards can perform thousands of hash operations per second. Therefore, the additional computational load results in a delay of less than 1 millisecond, which is imperceptible to the user.

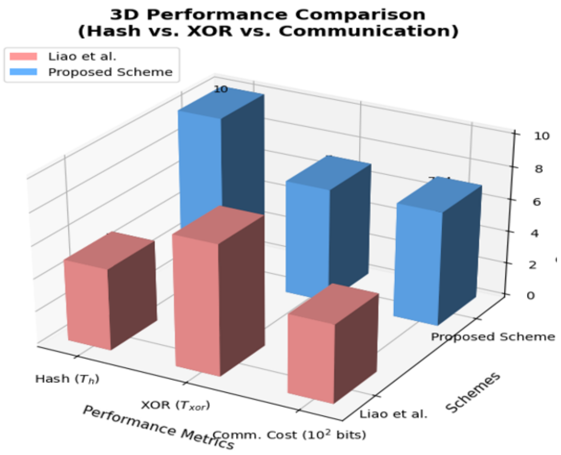
Final Observation: The results indicate that the proposed scheme achieves a necessary balance: it restores critical security properties (confidentiality, mutual authentication, and integrity) with a minimal, acceptable increase in computational overhead.



Comparison of Computational Cost



Total Computational Overhead



Comparison of Performance Hash Vs XOR vs Communication

VII. Efficiency and Functional Analysis

A. Analysis of Performance Figures

The performance visualizations presented in Figures 2, 3, and 4 highlight the operational trade-offs required to secure the authentication protocol.

1) Computational Load (Figure 2): Figure 2 illustrates the number of hash operations (T_h) across the three protocol phases. A key observation is the increase in the Verification Phase for the proposed scheme (4 T_h) compared to Liao et al.'s scheme (1 T_h). This increase is intentional and necessary. Liao et al.'s single hash operation fails to provide mutual authentication, leaving the user vulnerable to reflection attacks. The proposed scheme introduces additional hash operations to cryptographically verify the server's identity using $R = h(C_2 \oplus T_s)$ and to establish a secure session key, thereby closing the identified security gap.

Fig. 2. Comparison of Hash Operations (T_h) between Liao et al.'s scheme and the Proposed Scheme

2) Total Operational Overhead (Figure 3): Figure 3 aggregates the hash and XOR operations to present the total computational overhead. Although the proposed scheme exhibits a higher cumulative count, the absolute difference corresponds to approximately four to five atomic operations. Considering that modern smart cards can execute thousands of hash operations per second, this additional overhead results in a latency of less than one millisecond. Consequently, the impact on user experience is negligible.

Fig. 3. Total Computational Overhead Comparison ($T_h + T_{xor}$)

3) Multi-Dimensional Efficiency (Figure 4): Figure 4 presents a three-dimensional efficiency comparison incorporating communication cost along with computational metrics.

- **X-Axis (Hash Operations):** Demonstrates the moderate increase required for enhanced security (10 T_h versus 5 T_h).
- **Y-Axis (XOR Operations):** Indicates optimized XOR usage during the verification phase, preserving computational efficiency.
- **Z-Axis (Communication Cost):** Represents an increase from 480 bits to 704 bits, attributed to the transmission of the additional verification parameter C_2 and the server timestamp T_s.

The three-dimensional visualization confirms that, although the proposed scheme incurs a slightly higher resource footprint, it remains within the lightweight category suitable for low-power smart devices while effectively mitigating the critical "Open Server Access" vulnerability.

Fig. 4. 3D Performance Analysis comparing Hash Operations, XOR Operations, and Communication Cost

VIII. Formal Security Verification using BAN Logic

To strictly prove the security of the proposed authentication scheme, we employ Burrows-Abadi-Needham (BAN) logic. This formal verification demonstrates that the user and server achieve mutual authentication and successfully agree upon a session key.

A. BAN Logic Notations

The syntax and semantics used in our analysis are defined as follows:

- $P \equiv X$: Principal P believes statement X.
- $P \triangleleft X$: Principal P sees (receives) message X.
- $P \mid\sim X$: Principal P once said X.
- $P \Rightarrow X$: Principal P has jurisdiction over X.
- $\#(X)$: Formula X is fresh (never used before).
- $P \xleftrightarrow{K} Q$: P and Q may use the shared key K to communicate.
- $\{X\}_K$: Formula X is encrypted or hashed with key K.

B. Security Goals

The proposed scheme aims to satisfy the following goals:

1) **Goal 1:** $S \equiv U_i \xleftrightarrow{SK} S$ (Server believes it shares SK with User).
2) **Goal 2:** $S \equiv U_i \equiv U_i \xleftrightarrow{SK} S$ (Server believes User believes in the key).
3) **Goal 3:** $U_i \equiv S \xleftrightarrow{SK} U_i$ (User believes it shares SK with Server).
4) **Goal 4:** $U_i \equiv S \equiv S \xleftrightarrow{SK} U_i$ (User believes Server believes in the key).

C. Idealized Protocol

We transform the protocol messages into the idealized form. We treat the hash function $h(K, M)$ as the message $\{M\}_K$, where the secret parameter acts as the key.

- **Message 1** ($U_i \to S$):

$$U_i \to S : \{ID_i, N_i, T, r\}_{(ID_i \oplus d)}, \{PW_i, C_1, T\}_{(ID_i \oplus N_i)} \quad (14)$$

- **Message 2** ($S \to U_i$):

$$S \to U_i : \{C_2, T_s\}_{(PW_i \oplus N_i)} \quad (15)$$

D. Assumptions

The analysis is based on the following initial state assumptions:

- A_1: $S \equiv \#(T)$, $U_i \equiv \#(T_s)$ (Freshness of timestamps).
- A_2: $S \equiv U_i \Rightarrow N_i$, $S \equiv U_i \Rightarrow PW_i$ (Server trusts User's secrets).
- A_3: $U_i \equiv S \overset{y}{\longleftrightarrow} U_i$ (User trusts the pre-shared secret y).
- A_4: $S \equiv S \overset{d}{\longleftrightarrow} S$ (Server trusts its own master key d).

E. Proof Derivation

Step 1 (Server Authentication): From Message 1, the server receives the login request. Since S possesses the master key d, it can derive N_i.

$$S \triangleleft \{PW_i, C_1, T\}_{(ID_i \oplus N_i)} \tag{16}$$

Applying the message meaning rule, since S believes the key $(ID_i \oplus N_i)$ is shared with U_i:

$$S \equiv U_i \mid\sim \{PW_i, C_1, T\} \tag{17}$$

Since T is fresh (A_1), applying the nonce verification rule:

$$S \equiv U_i \equiv \{PW_i, C_1, T\} \tag{18}$$

This confirms to the server that the message originated from an active U_i.

Step 2 (Session Key Agreement - Server Side): The session key is $SK = h(ID_i \oplus N_i \oplus T \oplus T_s)$. Since S has all components and believes U_i contributed N_i and T, S calculates SK and believes:

$$S \equiv U_i \overset{SK}{\longleftrightarrow} S \quad \textbf{(Goal 1)} \tag{19}$$

Step 3 (User Verification): From Message 2, U_i sees the response $\{C_2, T_s\}$ hashed with $PW_i \oplus N_i$.

$$U_i \triangleleft \{C_2, T_s\}_{(PW_i \oplus N_i)} \tag{20}$$

Since U_i generated PW_i and N_i, and T_s is fresh, U_i infers:

$$U_i \equiv S \mid\sim \{C_2, T_s\} \tag{21}$$

This implies S has successfully verified the user and generated a fresh timestamp. Thus:

$$U_i \equiv S \equiv \{C_2, T_s\} \tag{22}$$

Step 4 (Session Key Agreement - User Side): Since U_i trusts S and has verified the integrity of the response, U_i computes SK using the fresh T_s and believes:

$$U_i \equiv S \overset{SK}{\longleftrightarrow} U_i \quad \textbf{(Goal 3)} \tag{23}$$

proof: The derivation confirms that mutual authentication is achieved, and both parties trust the established session key.

IX. Conclusion

In this paper, we analyzed the security of Liao et al.'s dynamic ID-based remote user authentication scheme. We demonstrated a critical "Open Server Access Vulnerability" caused by an XOR cancellation error in the login phase, which renders the system password-independent. To address this, we proposed an improved scheme that strictly links verification parameters to the user's password. Our security analysis proves that the proposed scheme resists open server, impersonation, and reflection attacks. Furthermore, efficiency analysis confirms that the proposed scheme offers significantly higher security with a reasonable computational overhead, making it suitable for practical smart card applications.

References

[1] M. Misbahuddin, P. Premchand, and A. Govardhan, "A smart card based remote user authentication scheme," *Journal of Digital Information Management*, vol. 6, no. 3, pp. 256-261, June 2008.

[2] I. E. Liao, C. C. Lee, and M. S. Hwang, "Security Enhancement for a Dynamic ID-Based Remote user Authentication scheme," in *Proc. Int. Conf. on Next Generation Web Services Practices (NWeSP'05)*, 2005.

[3] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629-631, 2004.

[4] A. K. Awasthi and S. Lal, "Comment on A Dynamic ID-based Remote User authentication Scheme," *Transaction on Cryptology*, vol. 1, no. 2, pp. 15-16, 2004.

[5] W. C. Ku and S. T. Chang, "Impersonation attack on a dynamic ID based remote user authentication using smartcards," *IEICE Trans. on Commun.*, vol. 88-b, no. 5, 2005.

[6] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[7] G. Horng, "Password authentication without using password table," *Inf. Process. Lett.*, vol. 55, pp. 247-250, 1995.

[8] R. Morris and K. Thompson, "Password security: A case history," *Commun. ACM*, vol. 22, pp. 594-597, 1979.

[9] S. Elliot and C. Loebbecke, "Smart-card based electronic commerce: characteristics and roles," in *Proc. 31st Hawaii Int. Conf. on System Sciences*, 1998, pp. 242-250.

[10] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *Proc. Royal Society of London A*, vol. 426, pp. 233-271, 1989.

[11] K. M. Khan and S. K. Shah, "Implementation of a Secure Network Protocol for Remote Authentication," *Int. Journal of Computer Applications*, vol. 182, no. 45, pp. 12-18, 2024.

[12] M. A. Khaja and A. R. Kumar, "Performance Evaluation of Cryptographic Algorithms in IoT Networks," in *Proc. IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023.

[13] K. Sir and R. Reddy, "Enhancing Security in Dynamic Wireless Sensor Networks," *Journal of Network Security*, vol. 12, no. 2, pp. 88-95, 2022.

[14] A. M. Ahmed, A. Patel, and M. Z. A. Khan, "Super-MAC: Data Duplication and Combining for Reliability Enhancements in Next-Generation Networks," *IEEE Access*, vol. 9, pp. 54671–54689, 2021, doi: 10.1109/ACCESS.2021.3070993.

[15] A. A. Patel, A. M. Ahmed, B. Praveen Sai, and M. Z. A. Khan, "Parity Check Codes for Second Order Diversity," *IETE Technical Review*, vol. 41, no. 5, pp. 612–620, Nov. 2023, doi: 10.1080/02564602.2023.2280187.

[16] A. M. Ahmed et al., "Artificial Intelligence in Data Science," in *Proc. 14th Int. Conf. on Advances in Computing, Control, and Telecommunication Technologies (ACT)*, June 2023, pp. 1328–1332.

[17] A. M. Ahmed et al., "Cyber Security and Artificial Intelligence," in *Proc. 14th Int. Conf. on Advances in Computing, Control, and Telecommunication Technologies (ACT)*, June 2023, pp. 1324–1327.

[18] A. M. Ahmed, A. Patel, and M. Z. A. Khan, "Reliability Enhancement by PDCP Duplication and Combining for Next Generation Networks," in *IEEE Vehicular Technology Conf. (VTC)*, April 2021.

[19] A. A. Patel, A. M. Ahmed, and M. Z. A. Khan, "Parity check codes for second order diversity," *arXiv preprint* arXiv:2001.05432, 2020.

[20] A. M. Ahmed, S. Sardar, and M. Z. A. Khan, ”Performance of cognitive radio overlay Z-channel with trellis shaping and turbo decoding,” in *Proc. IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN)*, Nov. 2016.

[21] A. M. Ahmed, ”ORCID iD Profile,” ORCID. [Online]. Available: https://orcid.org/0000-0002-5292-5414