Secure Lightweight Authentication Scheme for IoT-Enabled Smart Farming

Raveendra Babu Ponnuru
Dept. of CIS
Virginia Military Institute
Lexington, VA, USA
raveendrap@vt.edu

Mohammed Azab
Dept. of CIS
Virginia Military Institute
Lexington, VA, USA
mazab@vt.edu

Denis Gracanin
Dept. of Computer Science
Virginia Tech
Blacksburg, VA, USA
gracanin@vt.edu

***Abstract*—The integration of Internet of Things technology into agricultural systems has enabled the development of advanced farming platforms that improve productivity, sustainability, and food security through real-time monitoring, automation, and data analytics. However, this extensive connectivity also introduces significant security vulnerabilities and privacy concerns. To mitigate these risks, this work presents a novel authentication framework specifically designed for resource-constrained agricultural IoT environments. The proposed approach ensures mutual authentication between devices, provides efficient key management, and defends against various cyber threats while maintaining minimal computational and communication overhead. Beyond comparative analysis with existing methods, the framework undergoes comprehensive security validation through both informal analysis and formal verification using the Tamarin Prover, confirming its robustness against standard adversarial models. The results demonstrate that the solution achieves an optimal balance of security and efficiency, which makes it particularly suitable for deployment in smart farming applications.**

***Index Terms*—Authentication, Key Agreement Protocol, Smart Farming, Tamarin Prover, Security and Privacy.**

I. Introduction

The use of Internet of Things (IoT) in agriculture has accelerated the emergence of smart farming systems that enhance productivity, sustainability, and food security. IoT-enabled sensors, unmanned aerial vehicles, and wireless sensor networks facilitate precision agriculture by enabling real-time monitoring of soil, crops, livestock, and logistics. Prior studies demonstrated the applicability of IoT to agricultural monitoring, greenhouse and aquaculture systems, drone-assisted crop management, and precision environmental sensing [1], [2].

As smart farming systems expand, the volume and sensitivity of agricultural data grow, demanding solutions that are not only reliable but also scalable and resource-efficient. The dynamic nature of agricultural environments requires protocols that support interoperability among heterogeneous devices while ensuring real-time responsiveness. At the same time, the deployment of IoT in rural and resource-constrained settings necessitates lightweight mechanisms that conserve energy and bandwidth without compromising trust and reliability [3], [4].

Cybersecurity aspects of smart farming are yet to recieve necessary attention. For example, in our discussion with a subject matter expert at a dairy farm, one comment was particularly telling "*Who is going to hack my cow?*." In this context, this paper proposes a secure lightweight authentication scheme for IoT-enabled smart farming. The scheme ensures secure mutual authentication, efficient key management, and sustainable performance across diverse deployment scenarios. Its correctness and robustness are validated through both informal analysis and formal verification using the Tamarin Prover. By combining efficiency with strong assurance guarantees, this scheme provides a scalable and resilient foundation for next-generation agricultural systems.

"This paper received support from the Commonwealth Cyber Initiative (CCI), an investment in the advancement of cyber R&D, innovation, and workforce development. For more information about CCI, visit: www.cyberinitiative.org"

II. Related Work

The related work section reviews prior studies aimed at enhancing the security and efficiency of IoT-enabled smart farming systems. Ali et al. [5] proposed a remote user authentication scheme for agricultural monitoring, validated with BAN logic and AVISPA. Chen et al. [6] strengthened this design with anonymity, untraceability, forward secrecy, and resistance to insider, DDoS, and impersonation attacks. Chae and Cho [7] presented a lightweight scheme for greenhouse peer-to-peer farms, though limited to resisting dictionary and brute-force attacks. Rangwani et al. [8] developed an elliptic curve–based scheme but without anonymity, untraceability, or dynamic node addition. Wu and Tsai [9] introduced a private blockchain-based scheme, but its bilinear pairings incurred high computation and left it vulnerable to offline guessing and ephemeral secret leakage. Vangala et al. [10] applied blockchain and IoT to food supply chain security and later proposed smart contract–based key agreement for smart farming, followed by a blockchain-enabled scheme for vehicle-assisted precision agriculture with optimized elliptic curve operations [11]. While these approaches highlight the growing role of IoT, blockchain, UAVs, and deep learning in agriculture, most incur high overheads or remain attack-prone, limiting practicality. Our scheme addresses these gaps by providing secure lightweight authentication with low resource consumption and robust mutual authentication.

III. Smart Farming Scheme Models

The proposed scheme uses network and threat models.

A. Network Model

The proposed smart farming scheme uses a network model that consists of five core entities that collectively enable secure and reliable data collection, transmission, and processing (Figure 1). IoT-enabled smart farms serve as the primary data sources, comprising agricultural fields and dairy farms equipped with IoT sensors and devices. Agricultural sensors capture parameters such as soil moisture, temperature, light, and nutrient levels, whereas dairy farms utilize monitoring devices to track livestock health, milk yield, and feeding patterns. Each farm is connected to a gateway, which authenticates IoT devices, validates and aggregates sensed data, and securely forwards the information to the cloud server. The cloud server acts as the central authority responsible for secure storage, large-scale analytics, and system management, generating actionable insights and issuing configuration updates to gateways as needed. Communication within the model uses two channels: a wireless link between IoT devices and gateways, which is more susceptible to eavesdropping and jamming, and a wired connection between gateways and the cloud server to ensure reliable and secure backhaul transmission [12].

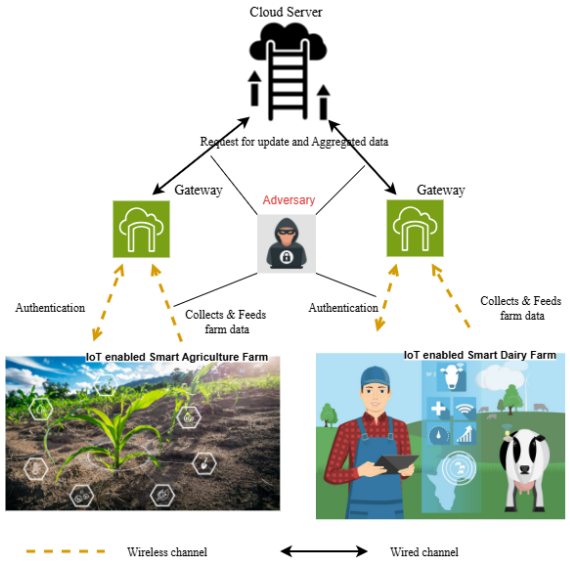

Fig. 1. Proposed Communication Scheme for Smart Farming.

B. Threat Model

A threat model is essential for systematically identifying adversarial behaviors and vulnerabilities in order to design a secure authentication and key management scheme. We consider a combination of the Dolev–Yao (DY) model [13], the Canetti–Krawczyk (CK) adversary model [14], and physical capture scenarios. The DY model, widely used in communication security, assumes that entities communicate over an insecure medium where adversaries can eavesdrop, intercept, modify, delete, or replay messages, treating IoT devices as potentially untrusted endpoints. The CK model extends these assumptions by allowing the adversary to access internal session states, such as ephemeral credentials and session keys, thereby capturing more realistic threats in resource-constrained IoT environments. Additionally, in smart farming systems, physical capture of devices is a significant risk since they cannot be continuously monitored [15]. Through side-channel techniques such as power analysis, an attacker may extract private data and use it to launch replay, impersonation, privileged-insider, or man-in-the-middle attacks [16].

IV. Proposed Scheme

The proposed scheme is structured into the following phases: *(A)* Initialization, *(B)* Gateway Registration, *(C)* IoT Device Registration, *(D)* Mutual Authentication and Key Agreement, *(E)* Dynamic Gateway Addition , and *(F)* Dynamic IoT Device Addition Phases. The notations employed throughout the text are summarized in Table I.

TABLE I
The notations employed throughout the text.

Symbol	Description
GW, d	Gateway, IoT device
ID_{GW_i}, ID_{d_i}	Identity of i^{th} Gateway and device
SK	Session key
SK_{GW_i}	i^{th} Gateway long-term secret key
PU_{GW_i}, PR_{GW_i}	i^{th} Gateway public/private key pair
R_{d_i}, n_1, n_2, n_3	Random numbers
$H(.)$	One-way hash function
$\|, \oplus$	Concatenation, XOR operations
$Æ$	Adversary

A. Initialization Phase

The Cloud Server (*CS*) generates the global system parameters and secret values required for secure communication:

1) The *CS* selects a collision-resistant one-way cryptographic hash function $H(.)$.
2) The *CS* chooses a long-term private key SK_{GW_i}, which is securely stored and never disclosed.

B. Gateway Registration Phase

Each gateway GW_i, which serves as a critical intermediary between IoT devices and the control server, must undergo registration with the *CS* over a secure channel:

1) The gateway GW_i selects its unique identifier ID_{GW_i} and transmits a registration request containing ID_{GW_i} to the *CS*.
2) Upon receiving the request, the *CS* computes $K_{GW_i} = H(ID_{GW_i} \| SK_{GW_i})$ and records the pair (ID_{GW_i}, K_{GW_i}) in the gateway registration table T_{GW}.
3) The *CS* then generates a public/private key pair (PU_{GW_i}, PR_{GW_i}) for the gateway, stores them securely, and delivers $< K_{GW_i}, H(SK_{GW_i}), PU_{GW_i}, PR_{GW_i} >$ to the gateway.

C. IoT Device Registration Phase

This phase enables IoT devices deployed in smart farming environments to register securely with the *CS* through a trusted channel:

1) Each IoT device d_i selects an identifier ID_{d_i} and a random number $R_{d_i} \in Z_p^*$. It then computes $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$ and $Y_{d_i} = R_{d_i} \oplus H(ID_{d_i})$, before sending $< PID_{d_i} >$ to the *CS*.
2) The *CS* verifies the uniqueness of PID_{d_i}.It then computes: $Q_S = H(PID_{d_i} \parallel K_{GW_i})$, $R_S = Q_S \oplus PID_{d_i}$, $W_S = H(PID_{d_i} \parallel SK_{GW_i})$, $C_{d_i} = H(PID_{d_i} \parallel W_S)$. The server stores $< ID_{GW_i}, R_S >$ in the IoT registration table T_d and maintains $< PID_{d_i}, C_{d_i}, T_{d_i} >$ in the verification table T_C. Finally, the *CS* delivers $< W_S, T_{d_i}, H(SK_{GW_i}) >$ to the device.
3) Upon receiving the response, the IoT device d_i computes $X_S = W_S \oplus PID_{d_i}$ and $C_{d_i} = H(ID_{d_i} \parallel W_S)$. It replaces W_S with X_S and stores $< X_S, Y_{d_i}, T_{d_i}, C_{d_i}, H(SK_{GW_i}) >$ in its tamper-proof memory for subsequent authentication.

D. Mutual Authentication and Key Agreement Phase

When an IoT device d_i intends to transmit sensed data or request services in the smart farming environment, it must first establish trust with the registered gateway GW_i through a secure mutual authentication and key agreement procedure:

1) The device d_i computes the following values: $R_{d_i} = Y_{d_i} \oplus H(ID_{d_i})$, $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$, $W_S = X_S \oplus PID_{d_i}$. It then verifies the consistency of $C_{d_i} = H(PID_{d_i} \parallel W_S)$. If the verification holds, the device continues; otherwise, the execution is aborted.
2) The device derives $Q_S = R_S \oplus PID_{d_i}$, selects a fresh random nonce $n_1 \in Z_p^*$, and computes: $B_{dg} = PID_{d_i} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$, $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$.
3) The device then transmits the login request $M_1 = \langle B_{dg}, D_{dg}, n_1 \rangle$ to the gateway GW_i.
4) Upon receiving M_1, the gateway GW_i recovers $PID_{d_i} = B_{dg} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$ and validates whether $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$. If the verification succeeds, GW_i authenticates d_i; otherwise, the process is terminated.
5) The gateway GW_i selects a fresh random nonce $n_2 \in Z_p^*$ and derives the session key: $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$. It further computes the authentication token $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$ and responds with $M_2 = \langle E_{dg}, n_2 \rangle$ to d_i.
6) On receipt of M_2, the device recomputes $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$ and verifies $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$. If the verification holds, d_i authenticates the gateway; otherwise, the process terminates. The device then computes $F_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$ and transmits $M_3 = \langle F_{dg} \rangle$ to GW_i.
7) Finally, the gateway verifies the correctness of $F_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$. Successful verification reconfirms the authenticity of the device. At this point, both d_i and GW_i securely establish and share the session key SK, which will be employed for subsequent encrypted communications.

E. Dynamic Gateway Addition Phase

In this phase, a new gateway GW^{new} can join the smart farming system by sending a registration request to *CS*. Once registered, the information of GW^{new} is securely shared with all previously registered IoT devices and gateways to maintain synchronization:

1) The new gateway GW^{new} initiates the process by sending a registration request containing its identity $ID_{GW_i}^{new}$ to the *CS*.
2) Upon receiving the request, the *CS* computes $K_{GW_i}^{new} = H(ID_{GW_i}^{new} \parallel SK_{GW_i}^{new})$ and stores the entry $\langle ID_{GW_i}^{new}, K_{GW_i}^{new} \rangle$ in the gateway registration table T_{GW}.
3) The *CS* then generates a public/private key pair $(PU_{GW_i}^{new}, PR_{GW_i}^{new})$ for the newly added gateway, stores them securely, and delivers $\langle K_{GW_i}^{new}, PU_{GW_i}^{new}, PR_{GW_i}^{new} \rangle$ to GW^{new}.
4) To synchronize with existing IoT devices, the *CS* retrieves the stored values $\langle PID_{d_i}, C_{d_i} \rangle$ of all registered IoT devices, and computes $Q_S^{new} = H(PID_{d_i} \parallel K_{GW_i}^{new})$. The *CS* delivers Q_S^{new} securely to each IoT device and updates $\langle PID_{d_i}, C_{d_i} \rangle$ in the verification table T_C. Each IoT device then computes $R_S^{new} = Q_S^{new} \oplus PID_{d_i}$ and stores it in its local table T_d.

F. Dynamic IoT Device Addition Phase

This phase enables the secure registration of a new IoT device d_i^{new} within the smart farming system through interaction with the *CS*:

1) The IoT device d_i^{new} selects an identifier $ID_{d_i}^{new}$ and chooses a random number $R_{d_i}^{new} \in Z_p^*$. It then computes: $PID_{d_i}^{new} = H(ID_{d_i}^{new} \parallel R_{d_i}^{new})$, $Y_{d_i}^{new} = R_{d_i}^{new} \oplus H(ID_{d_i}^{new})$. The device transmits $\langle PID_{d_i}^{new}, Y_{d_i}^{new} \rangle$ to the *CS* for registration.
2) The *CS* verifies the uniqueness of $PID_{d_i}^{new}$ and then computes: $Q_S^{new} = H(PID_{d_i}^{new} \parallel K_{GW_i})$, $R_S = Q_S^{new} \oplus PID_{d_i}^{new}$, $W_S = H(PID_{d_i}^{new} \parallel SK_{GW_i})$. The *CS* stores $\langle ID_{GW_i}, R_S \rangle$ in the IoT device registration table T_d, and also records $\langle PID_{d_i}^{new}, C_{d_i}^{new}, T_{R=1} \rangle$ in the verification table T_C. Finally, it delivers $\langle W_S, T_d, H(SK_{GW_i}) \rangle$ securely to d_i^{new}.
3) Upon receiving the response, the IoT device d_i^{new} computes $X_S = W_S \oplus PID_{d_i}^{new}$ and $C_{d_i}^{new} = H(ID_{d_i}^{new} \parallel W_S)$. The device replaces W_S with X_S and securely stores $\langle X_S, Y_{d_i}^{new}, T_d, C_{d_i}^{new}, H(SK_{GW_i}) \rangle$ in its tamper-proof onboard memory for future authentication and communication.

V. Security Analysis

A. Informal Security Analysis

We provide an informal security analysis of the proposed scheme (Propositions 1–8), demonstrating its robustness

TABLE II
MUTUAL AUTHENTICATION AND KEY AGREEMENT PHASE

IoT Device d_i		***Gateway*** GW_i
Computes $R_{d_i} = Y_{d_i} \oplus H(ID_{d_i})$, $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$, $W_S = X_S \oplus PID_{d_i}$ and verifies $C_{d_i} = H(PID_{d_i} \parallel W_S)$.		
	$d_i \rightarrow GW_i$: $M_1 = \langle B_{dg}, D_{dg}, n_1 \rangle$	
Generates $B_{dg} = PID_{d_i} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$, $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$.		Computes $PID_{d_i} = B_{dg} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$, verifies $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$. Selects $n_2 \in Z_p^*$, computes $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$, $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$.
	$GW_i \rightarrow d_i$: $M_2 = \langle E_{dg}, n_2 \rangle$	
Computes $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$, verifies $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$. Generates $F_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$.		
	$d_i \rightarrow GW_i$: $M_3 = \langle F_{dg} \rangle$	
		Verifies $F_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$, confirms authenticity of d_i.
Result: Both d_i and GW_i establish shared session key SK.		

against a range of critical security threats in smart farming environments. The analysis outlines how the proposed scheme preserves confidentiality, integrity, and privacy while maintaining efficiency.

***Proposition 1*: Prevents IoT device impersonation attacks.**

Proof: Case 1: To impersonate a legitimate device d_i, an adversary Æ must construct a valid login request $M_1 = \langle B_{dg}, D_{dg}, n_1 \rangle$. However, the login procedure requires the device to compute $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$, $Y_{d_i} = R_{d_i} \oplus H(ID_{d_i})$, $W_S = X_S \oplus PID_{d_i}$, and validate $C_{d_i} = H(PID_{d_i} \parallel W_S)$. Without the correct credentials (ID_{d_i}, R_{d_i}), Æ cannot proceed. The probability of successful guessing is negligible.

Case 2: The device d_i never discloses its true identity to any GW_i. The pseudo-identity PID_{d_i} is only obtained by the gateway during authentication as $PID_{d_i} = B_{dg} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$. Even if a compromised gateway attempts impersonation, it must still compute $Q_S = H(PID_{d_i} \parallel K_{GW_i})$, where $K_{GW_i} = H(ID_{GW_i} \parallel SK_{GW_i})$. Since Q_S is unique for each gateway, impersonation remains infeasible.

***Proposition 2*: Prevents gateway impersonation attacks.**

Proof: Suppose an adversary Æ intercepts M_1 and tries to impersonate a legitimate gateway by forging a response $M_2^{Æ}$. Without access to PID_{d_i} and the correct Q_S, Æ cannot compute the session key $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$ or the authentication token $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$.

If Æ transmits a forged $M_2^{Æ} = \langle E_{dg}^{Æ}, n_2 \rangle$, the device computes its own E_{dg} and detects the mismatch ($E_{dg}^{Æ} \neq E_{dg}$). Thus, impersonation attempts by adversaries posing as gateways are unsuccessful.

***Proposition 3*: Resistant to man-in-the-middle attacks.**

Proof: During authentication, d_i and GW_i exchange M_1, M_2, and M_3 over an open channel. Although Æ may capture these messages, each value is derived using secure one-way hash operations, e.g., $B_{dg} = PID_{d_i} \oplus H(ID_{GW_i} \parallel n_1 \parallel H(SK_{GW_i}))$, $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$, $E_{dg} = H(SK \parallel ID_{GW_i} \parallel n_1 \parallel n_2)$. Because Q_S and SK are session-dependent and never transmitted, an adversary cannot extract sensitive data or forge valid responses. Hence, these attacks thus succeed only with negligible probability.

***Proposition 4*: Prevents replay attacks.**

Proof: An adversary may attempt to reuse an earlier login message M_1. However, each gateway records (PID_{d_i}, n_1) for every session. If the same message is replayed, the mismatch in nonces causes the gateway to reject the request. Similar verification prevents replay of M_2 and M_3. Since fresh nonces (n_1, n_2, n_3) are employed in every session, replay attacks are effectively mitigated.

***Proposition 5*: Withstands privileged insider attacks.**

Proof: During registration, the device d_i never transmits raw identifiers or passwords. Instead, it submits pseudo-identifiers such as $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$ and $Y_{d_i} = R_{d_i} \oplus H(ID_{d_i})$, where R_{d_i} is a random nonce. Since the server does not store sensitive tables (e.g., plaintext password databases), insiders cannot reconstruct original credentials. Authentication relies solely on verification of dynamic values such as $D_{dg} = H(PID_{d_i} \parallel Q_S \parallel n_1)$. Hence, insider attacks are prevented.

***Proposition 6*: Ensures device anonymity & untraceability.**

Proof: The true identity ID_{d_i} is never revealed during communication. Instead, $PID_{d_i} = H(ID_{d_i} \parallel R_{d_i})$ is used as a pseudo-identity. Additionally, session values (M_1, M_2, M_3) are randomized using fresh nonces, making each session unlinkable. Due to the one-way property of hash functions, adversaries cannot trace or correlate d_i across multiple sessions. Thus, anonymity and untraceability are preserved.

***Proposition 7*: Achieves forward secrecy.**

Proof: Forward secrecy requires that compromise of long-term keys does not affect past session keys. Here, the session key is derived as $SK = H(PID_{d_i} \parallel n_1 \parallel n_2 \parallel Q_S)$. Even if n_1 and n_2 are exposed, Æ cannot reconstruct SK without knowledge of PID_{d_i} and Q_S. Since these values are derived with randomness and not stored, past sessions remain secure.

***Proposition 8*: Mitigates denial-of-service attacks.**

Proof: An adversary may attempt to overload a gateway with invalid messages. The proposed scheme employs lightweight operations such as hashing, XOR, and concatenation, enabling GW_i to quickly discard illegitimate messages. This reduces the computational burden and prevents service degradation. While physical jamming remains outside scope, the scheme effectively mitigates computational DoS attempts.

```
==============================================================
summary of summaries:

analyzed: farming.spthy

executable (all-traces): verified (7 steps)
Mutual_authentication_Device (all-traces): verified (5 steps)
Mutual_authentication_GW (all-traces): verified (5 steps)
Session_key_secrecy (all-traces): verified (4 steps)
Forward_secrecy (exists-trace): verified (3 steps)
Anonymity_device (exists-trace): verified (3 steps)
Unlinkability_device (all-traces): verified (1 step)
Session_key_freshness (all-traces): verified (2 steps)
==============================================================
```

Fig. 2. Tamarin verification result.

B. Formal Security Analysis with Tamarin Prover

Tamarin is a state-of-the-art formal verification tool designed for analyzing the security properties of cryptographic protocols. It is highly versatile and supports the modeling of complex cryptographic operations through built-in equational theories. For example, elliptic-curve cryptographic operations such as point multiplication are natively supported by the function `pmult` [17], highlightsing the expressiveness of the tool. Another distinguishing feature of Tamarin is its ability to test protocols against user-defined adversarial capabilities, setting it apart from many other formal verification frameworks. Furthermore, the tool allows the modeling of global mutable states, making it suitable for the analysis of a wide spectrum of protocols ranging from simple authentication schemes to complex group-key protocols.

Security properties to be verified are expressed in first-order logic. When a property holds, Tamarin outputs a confirmation of verification; otherwise, it reports a falsification and generates an attack graph to illustrate the adversarial strategy leading to the violation. This graphical output enhances the interpretability of failed verifications. A comprehensive description of the tool and its functionalities is available in the official manual [18].

We use the Tamarin prover to verify that the proposed scheme meets essential security goals, including mutual authentication, known-key secrecy, forward secrecy, session key freshness, anonymity, unlinkability, resistance against masquerading, replay, and man-in-the-middle attacks Figure 2.

VI. COMPARATIVE STUDY

We compare the proposed scheme with several widely cited approaches, including those of Chen et al. [6], Ali et al. [5], Chae and Cho [7], and Rangwani et al. [8]. The comparison considers both communication and computation costs, as well as core functional and security features, to provide a balanced assessment of the proposed scheme's effectiveness.

A. Communication Cost

To evaluate the communication costs of the proposed scheme in comparison with existing schemes, the following assumptions are adopted: the identity size is 160 bits, the random number is 160 bits, the hash output (using SHA-256) is 256 bits, and the timestamp is 32 bits. For a point on an elliptic curve of the form $P = (P_x, P_y)$, where P_x and P_y represent the x and y coordinates, respectively, the size is calculated as $(160+160) = 320$ bits. This assumption is based on the observation that a 160-bit elliptic curve cryptosystem (ECC) offers a security level approximately equivalent to that of a 1024-bit RSA-based public key cryptosystem [19].

TABLE III
COMPARISON OF COMMUNICATION COSTS.

Scheme	No. of messages	Total cost (in bits)
Rangwani et al. [8]	5	4128
Chen et al. [6]	4	4960
Chae and Cho [7]	4	12896
Ali et al. [5]	5	5504
Vangala et al. [11]	2	2240
Proposed work	3	1344

Table III compares the communication costs of existing schemes in terms of message exchanges and total bits. The proposed scheme requires only three messages and 1344 bits, which is substantially lower than all other methods, including Vangala et al. [11] (2240 bits) and Chae and Cho [7] (12896 bits). This efficiency underscores its suitability for resource-constrained smart farming environments.

B. Computational Cost

The computation costs of the proposed scheme are systematically compared with those of Ali et al. [5], Chen et al. [6], Chae and Cho [7], Rangwani et al. [8], and Vangala et al. [11]. To ensure clarity and consistency, the following notations are adopted: T_h, T_{em}, T_{ea}, T_{ep}, T_{fe}, and $T_{enc/dec}$, denoting the execution times of the one-way hash function (SHA-256 producing a 256-bit digest, elliptic curve multiplication, elliptic curve addition, modular exponentiation, fuzzy extractor operation for biometric authentication schemes, and symmetric encryption/decryption using AES-128, respectively.

The computation time estimates are based on experimental benchmarks reported in [20], measured in milliseconds. The corresponding values are: $T_{em} \approx 13.405$ ms, $T_{ea} \approx 0.081$ ms, $T_h \approx 0.056$ ms, and $T_{ep} \approx 2.249$ ms. Additionally, $T_{fe} \approx T_{em}$, while $T_{enc/dec} \approx T_h$. Table IV compares computation costs across schemes. The proposed scheme achieves the lowest overhead, requiring only $16T_h$ (0.896 ms), far outperforming others such as Chae and Cho [7] (107.85 ms) and Vangala et al. [11] (134.99 ms), confirming its efficiency for smart farming environments.

C. Security Features

Table V compares the security and functionality properties of several representative schemes. The results clearly show that the proposed scheme offers the most comprehensive protection, surpassing all baseline approaches. Rangwani et al. [8]

TABLE IV
COMPARISON OF COMPUTATION COSTS.

Scheme	Total cost	In ms
Rangwani et al. [8]	$15T_h + 6T_{em}$	81.27
Chen et al. [6]	$3T_h$	2.07
Chae and Cho [7]	$8T_h + 8T_{em} + 2T_{ea}$	107.85
Ali et al. [5]	$19T_h + T_{fe} + T_{em} + T_{enc/dec}$	14.92
Vangala et al. [11]	$14T_h + T_{em} + 2T_{ea}$	134.99
Proposed Work	$16T_h$	0.896

TABLE V
COMPARISON OF SECURITY AND FUNCTIONALITY ATTRIBUTES.

Scheme	SF_1	SF_2	SF_3	SF_4	SF_5	SF_6	SF_7	SF_8	SF_9	SF_{10}
Rangwani et al. [8]	×	✓	✓	✓	✓	✓	×	×	✓	×
Chen et al. [6]	✓	✓	✓	✓	✓	×	×	✓	✓	✓
Chae et al. [7]	×	×	✓	×	×	×	×	×	✓	×
Ali et al. [5]	×	×	✓	✓	✓	×	×	×	✓	×
Vangala et al. [11]	✓	✓	×	✓	✓	✓	✓	✓	✓	×
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Ten security and functional properties — SF_1: Anonymity; SF_2: Device impersonation attack; SF_3: Gateway impersonation attack; SF_4: Replay attack; SF_5: Man-in-the-middle attack; SF_6: Privileged insider attack; SF_7: Forward secrecy; SF_8: Dynamic device addition; SF_9: Mutual authentication; and SF_{10}: Denial-of-service (DoS) attack. (✓*: scheme is secure or supports the feature;* ×*: scheme is insecure or does not support the feature.*)

fails to achieve SF_1 (anonymity) and SF_7–SF_8, while Chen et al. [6] omits SF_6 (insider attack resistance) and SF_7 (session key secrecy). Chae et al. [7] is significantly weaker, lacking SF_1, SF_2, SF_4–SF_8, and SF_{10}. Similarly, Ali et al. [5] does not support SF_1, SF_2, SF_6–SF_8, and SF_{10}. Although Vangala et al. [11] achieves broader coverage, it still misses SF_3 and SF_{10}. In contrast, the proposed scheme satisfies all properties (SF_1–SF_{10}), establishing its superiority as a secure and practical solution for IoT and CPS environments.

VII. CONCLUSION

This work addressed the critical need for secure and efficient authentication in IoT-enabled smart farming systems. While prior approaches have contributed to advancing data-driven agriculture, many suffer from high computational and communication costs or remain vulnerable to security threats, limiting their practicality. The proposed lightweight scheme overcomes these shortcomings by ensuring secure mutual authentication, robust key management, and resistance against diverse attacks, while maintaining minimal resource consumption. Comparative analysis confirms that the scheme achieves an effective balance between security and efficiency, making it well-suited for resource-constrained agricultural IoT environments. Furthermore, the design principles underpinning this work provide a foundation for extending lightweight, secure communication frameworks to broader cyber-physical systems, with future directions including blockchain, post-quantum security to enhance resilience.

REFERENCES

[1] M. Dutta, D. Gupta, S. Tharewal, D. Goyal, J. K. Sandhu, M. Kaur, A. A. Alzubi, and J. M. Alanazi, "Internet of things-based smart precision farming in soilless agriculture: Opportunities and challenges for global food security," *IEEE Access*, 2025.

[2] K. Haseeb, I. Ud Din, A. Almogren, and N. Islam, "An energy efficient and secure IoT-based WSN framework: An application to smart agriculture," *Sensors*, vol. 20, no. 7, p. 2081, 2020.

[3] U. Peddibhotla, R. Kumar, C. Sobin, P. Kumar, D. Javeed, and N. Islam, "Securing agricultural communications: Blockchain integration in uav networks for smart farming," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2024, pp. 1443–1449.

[4] R. B. Ponnuru, S. A. Kumar, M. Azab, B. Palaniswamy, and G. R. Alavalapati, "Robust authentication and key agreement protocol for smart microgrid environment," *Journal of Information Security and Applications*, vol. 94, p. 104202, 2025.

[5] R. Ali, A. K. Pal, S. Kumari, M. Karuppiah, and M. Conti, "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Generation Computer Systems*, vol. 84, pp. 200–215, 2018.

[6] M. Chen, T.-F. Lee, and J.-I. Pan, "An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring," *Sensors*, vol. 19, no. 5, p. 1146, 2019.

[7] C.-J. Chae and H.-J. Cho, "Enhanced secure device authentication algorithm in P2P-based smart farm system," *Peer-to-peer networking and applications*, vol. 11, no. 6, pp. 1230–1239, 2018.

[8] D. Rangwani, D. Sadhukhan, S. Ray, M. K. Khan, and M. Dasgupta, "An improved privacy preserving remote user authentication scheme for agricultural wireless sensor network," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, p. e4218, 2021.

[9] H.-T. Wu and C.-W. Tsai, "An intelligent agriculture network security system based on private blockchains," *Journal of Communications and Networks*, vol. 21, no. 5, pp. 503–508, 2019.

[10] A. Vangala, A. K. Das, N. Kumar, and M. Alazab, "Smart secure sensing for IoT-based agriculture: Blockchain perspective," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17 591–17 607, 2020.

[11] A. Vangala, S. Roy, and A. K. Das, "Blockchain-based lightweight authentication protocol for IoT-enabled smart agriculture," in *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*. IEEE, 2022, pp. 110–115.

[12] D. Gračanin, K. P. Adams, and M. Eltoweissy, "Data replication in collaborative sensor network systems," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC 2006)*, 10–12 Apr. 2006, pp. 389–396.

[13] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 2003.

[14] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2002, pp. 337–351.

[15] P. R. Babu, A. G. Reddy, B. Palaniswamy, and A. K. Das, "Ev-puf: Lightweight security protocol for dynamic charging system of electric vehicles using physical unclonable functions," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3791–3807, 2022.

[16] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE transactions on computers*, vol. 51, no. 5, pp. 541–552, 2002.

[17] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," in *International Conference on Computer Aided Verification*. Springer, 2013, pp. 696–701.

[18] "Tamarin Book: Modeling and analyzing security protocols with tamarin: A comprehensive guide," https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf/, [Online; accessed 11-Oct-2025].

[19] E. Barker and Q. Dang, "NIST special publication 800-57 part 1, revision 4," *NIST, Tech. Rep*, vol. 16, p. 51, 2016.

[20] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2018.