cryptography

MDPI

Article

Lightweight Scheme for Secure Signaling and Data Exchanges in Intelligent Precision Agriculture

Thekaa Ali Kadhim [1], Zaid Ameen Abduljabbar [1,2,3,*], Hamid Ali Abed AL-Asadi [1], Vincent Omollo Nyangaresi [4,5], Zahraa Abdullah Ali [6] and Iman Qays Abduljaleel [1]

[1] Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
[2] Department of Business Management, Al-Imam University College, Balad 34011, Iraq
[3] Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 518000, China
[4] Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo 40601, Kenya
[5] Department of Applied Electronics, Saveetha School of Engineering, SIMATS, Chennai 602105, India
[6] Department of Cyber Security, Al-Kunooz University College, Basrah 61004, Iraq
* Correspondence: zaid.ameen@uobasrah.edu.iq

Abstract: Intelligent precision agriculture incorporates a number of Internet of Things (IoT) devices and drones to supervise agricultural activities and surroundings. The collected data are then forwarded to processing centers to facilitate crucial decisions. This can potentially help optimize the usage of agricultural resources and thwart disasters, enhancing productivity and profitability. To facilitate monitoring and decision, the smart devices in precision agriculture must exchange massive amounts of data across the open wireless communication channels. This inadvertently introduces a number of vulnerabilities, exposing the collected data to numerous security and privacy threats. To address these issues, massive security solutions have been introduced to secure the communication process in precision agriculture. However, most of the current security solutions either fail to offer perfect protection or are inefficient. In this paper, a scheme deploying efficient cryptographic primitives such as hashing, exclusive OR and random number generators is presented. We utilize the Burrows–Abadi–Needham (BAN) logic to demonstrate the verifiable security of the negotiated session keys. In addition, we execute an extensive semantic analysis which reveals the robustness of our scheme against a myriad of threats. Moreover, comparative performance evaluations demonstrate its computation overheads and energy consumption efficiency.

Keywords: precision agriculture; smart farming; security; privacy; attacks; efficiency

Academic Editor: Josef Pieprzyk

Received: 10 December 2024
Revised: 9 January 2025
Accepted: 15 January 2025
Published: 17 January 2025

Citation: Kadhim, T.A.; Abduljabbar, Z.A.; AL-Asadi, H.A.A.; Nyangaresi, V.O.; Ali, Z.A.; Abduljaleel, I.Q. Lightweight Scheme for Secure Signaling and Data Exchanges in Intelligent Precision Agriculture. *Cryptography* **2025**, *9*, 7. https://doi.org/10.3390/cryptography9010007

1. Introduction

Precision agriculture (PA) involves the integration of technologies such as drones, Wireless Sensor Networks (WSNs), IoT, artificial intelligence, geographic information systems (GISs), geospatial technologies (GTs) and machine learning to improve agricultural outputs. In PA, a large number of different sensors are used to remotely monitor farmlands. They also collect data regarding livestock health, environmental and crop growth. For instance, real-time data regarding soil conditions, quantity and quality of yield, crops, effects of deployed chemicals on crops, weather conditions and time of yield can be remotely collected and forwarded to servers for analysis and decision making [1]. This can help reduce threats to the production process and hence improve agricultural productivity. In addition, drones have been utilized for the application of pesticide spray and management of crops (such

as disease discovery and management of weeds). In IoT-based smart agriculture, several technologies such as end-user applications, radio frequency identification (RFID), cloud computing and WSNs are utilized [2]. These technologies have been shown to positively impact the agricultural sector in terms of enhanced productivity, water conservation and increased income [3]. As explained in [4], crop water usage optimization is driven by the need for sustainable agricultural practices in the face of growing global food demands and climatic changes. According to [5], PA greatly helps in resource management in terms of agronomic (improving farm inputs effectiveness and hence yields), economic (enhancing competitiveness and productivity via efficient farming practices) and environment (reduction in ecological effects of agriculture via optimization of farm inputs usage). As explained in [6], smart agronomy has been implemented to increase productivity and offer disaster protection using limited resources. Similarly, faster decision making due to the incorporation of IoT and UAVs has been noted to result in costs savings and increased yields [7,8].

It is evident that the agricultural sector has benefited from rapid and intense transformations in access methods, data acquisition and sharing technologies. This has been reflected in production quality improvements, effective usage of natural resources and environmental sustainability. However, the usage of these integrated technologies render smart farming environments vulnerable to numerous cyber-security threats. For instance, the sensed data are exchanged over the open public internet and are hence susceptible to myriad threats such as eavesdropping and unauthorized access. These threats can result in the compromise of data availability, integrity and confidentiality. It is also possible for data belonging to an agricultural partner to be tampered with or altered, reducing its trustworthiness [9]. There is therefore a need for robust data verification schemes [10,11]. Due to extensive collaboration among different PA entities from diverse domains, there is a need for device authentication so as to foster secure communication [12,13]. This ensures that sensitive agricultural data are only shared with legitimate network entities. In addition, there is need to uphold trust by protecting both data in transit and in storage against any form of misuse. In this regard, we make the following contributions:

- We timestamp all the exchanged messages and validate them at the receiver end so as to thwart any packet replays.
- Frequently refreshed random numbers are incorporated in intermediary parameters to prevent against forgery and spoofing attacks.
- Elaborate formal security substantiation is executed to demonstrate the security verifiability of the negotiated session keys.
- Extensive semantic security analyses are executed, with results showing the resilience of our protocol against myriad precision agriculture threats.
- We carry out comparative performance evaluations to demonstrate the efficiency of our scheme. Specifically, this protocol is shown to consume the lowest energy and computation overheads at relatively low communication costs.

The remainder of this paper is organized as follows: Section 2 presents the related past research works, while Section 3 presents the detailed description of our protocol. Conversely, Sections 4 and 5 discuss the security analyses and performance evaluation of the proposed scheme, respectively. Finally, Section 6 concludes this paper and offers some future research scopes.

1.1. Motivation

The requirement for data protection in precision agricultural networks and devices has seen the development of numerous security solutions. Most of these schemes are based on cryptographic operations such as public key infrastructure (PKI), blockchain

technology, elliptic curve cryptography (ECC) and identity-based cryptography (IBC). However, schemes based on PKI, such as the Rivest–Shamir–Adleman (RSA) algorithm require third-party certificate authorities (CAs) for digital certificate assignment among devices. This comes with high costs for certificate management and storage. Although IBC-based crypto-systems solves this issue by using the identity of the devices as the public key and designating private key generator (PKG) for private keys distribution, these systems have key escrow challenges. For consistency, blockchain-based schemes require that each network entity maintain identical copies of the blockchains. This is detrimental to memory-limited agriculture Internet of Things (AIoT) devices. Similarly, ECC-based schemes require operations such as scalar multiplications which results in extensive overheads. In addition, most of the security solutions for PA security are based on centralized architectures, rendering them susceptible to single point of failure. Therefore, a truly efficient but robust security solution for resource-limited PA devices is required.

1.2. Research Goals

To provide perfect security in an environment characterized by frequent security threats, the following security goals should be pursued.

Message authentication: all the exchanged messages transmitted over the public channels should be verified at the receiver end.

Confidentiality: A session key need to be negotiated to encipher all the messages exchanged across the public communication channels. This presents attackers from eavesdropping on any secret parameters from the intercepted messages.

Availability: It should be cumbersome for the attackers to launch denial of service and de-synchronization attacks that can potentially lock legitimate users from accessing the required functionality and services.

Integrity: Based on any intercepted messages, attackers should be unable to modify or insert any bogus messages to the communication channel.

Perfect key secrecy: Adversaries should be unable to deploy the captured current session keys to work out session keys for the past as well as consequent communication sessions.

Anonymity: It should be cumbersome to reveal the real identities of any network entity based on the intercepted messages.

Robustness against threats: the authentication scheme needs to be resistant against typical intelligent precision agriculture threats such as denial of service (DoS), packet replay, man-in-the-middle (MitM), de-synchronization, privileged insider, impersonation and spoofing attacks.

1.3. Threat Model

The Canetti and Krawczyk model is one of the most popular threat models, and hence, we adopt it in our proposed scheme. In this model, adversary Ã is assumed to have the capabilities of eavesdropping on the wireless communication channel, intercepting all exchanged messages, modifying, or deleting these messages. In addition, the attacker can insert malicious messages into the communication channels to mislead the unsuspecting receivers. Moreover, Ã can steal smart farm devices and sensors, after which all stored security values can be extracted via power analysis. All low-entropy passwords can also be guessed by Ã in polynomial time, in addition to accessing all ephemeral keying parameters.

2. Related Works

In light of the many cyber threats in the smart agriculture ecosystem, several security schemes have been presented in the literature. For instance, blockchain technology allows the independent auditing and verification of transactions. This renders it ideal for

enhancing trust during data access and recording in distributed network architectures [14]. Therefore, blockchain-based security solutions have been developed in [3,12,15–22]. These schemes help mitigate the single point of failure problems [23] in centralized architecture-based protocols developed in [6,24–27]. In addition, they boost both integrity and messages source verification. However, the storage and computation requirements for the blockchains are extensive IoT devices due to their limited resources [28]. Conversely, the security techniques developed in [29] utilize only one-way hash chains and hence can address the challenges in blockchain-based solutions. However, it does not offer untraceability, anonymity and dynamic node addition. In addition, it is susceptible to privileged insider, offline password guessing, impersonation and smart card loss attacks [3]. Similarly, the three factor user authentication protocol in [30] is not resilient session hijacking and eavesdropping threats.

Hinged on IBC, authentication protocols are presented in [31–37]. However, IBC-based schemes are susceptible to key escrow threats [38]. Similarly, the security solutions in [3,13,20,21] face key escrow threats. Although the scheme in [32] offers data confidentiality and mitigates several attacks, the leakage of the master key can result in the compromise of all the session keys [31]. Similarly, the protocol in [33] provides identity authentication, session key security and location privacy but incurs relatively high costs [31]. On the other hand, the scheme in [34] cannot withstand MitM, impersonation, DoS, privileged insider and offline password guessing attacks [3]. Similarly, the security technique presented in [35] cannot withstand privileged insider and DoS attacks [3], while the approach in [36] does not support secure communication and cannot withstand ephemeral secret leakage (ESL) attacks [39].

Some researchers have also utilized ECC to develop security schemes to secure systems and networks; for example, ECC-based authentication protocols have been developed in [40–44]. Although the technique in [40] protects against attacks such as impersonation and replay, it incurs high communication costs [31]. The schemes in [41,42] have relatively lower computation and communication overheads but fail to support anonymity and untraceability. In addition, the technique in [41] is exposed to ESL, smart card loss, DoS and privileged insider attacks [3]. For their part, the protocols in [42–44] cannot protect against ESL, offline guessing attacks, privileged insider and DoS attacks [3]. The Rabin cryptosystem-based scheme in [45] supports both untraceability and anonymity and hence can address the issues in [41,42]. However, it is not robust against privileged insider, smart card loss, ESL and offline guessing attacks [3]. Additionally, it has extensive communication overheads. In the same breath, the protocol in [46] exhibits extensive overheads due to bilinear pairing operations [47]. On the other hand, a scheme based on fuzzy extraction is presented in [48]. However, this scheme is not evaluated against attacks such as side-channeling, de-synchronization and session hijacking. A multi-factor user authentication protocol is developed in [49], while a multi-server scheme is developed in [50]. On the other hand, a homomorphic signcryption system is presented in [51], while an ECC-based scheme for authentication in smart agriculture monitoring systems is introduced in [52]. However, the scalar point multiplications in [51,52] and the fuzzy extraction operations in [49] render these schemes computationally extensive. Similarly, the required encryptions and decryptions in [50] increase its execution time.

From the discussions above, it is evident that the achievement of low-latency authentication with minimal communication overheads presents some challenges. In addition, the majority of the works in the literature are still vulnerable and hence expose the communication process to attacks. Our scheme is demonstrated to be not only efficient but also robust against conventional attacks inherent in precision agriculture environment.

3. The Proposed Scheme

The network model of our protocol comprises the user (U_i), controller node (CN_j) and smart farm sensor node (SN_k). The user in this case is the farmer performing some remote monitoring of his/her farm, while the sensor nodes are the actual devices that collect data from the field.

Conversely, the controller node performs the registration of all the users and sensors before the actual data collection process, as shown in Figure 1. The symbols utilized throughout this work are detailed in Table 1.

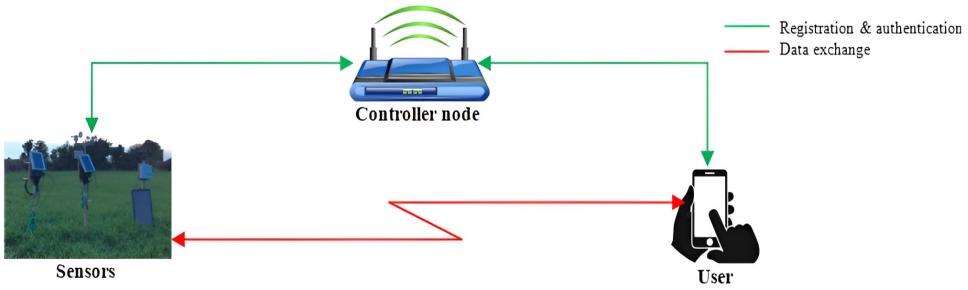

Figure 1. Network model.

Table 1. Symbols.

Symbol	Description
U_i	ith smart farm user
SN_k	Smart farm node k
CN_j	Controller node j
ϕ_j	Secret key for CN_j
SID_k	Unique identity belonging to SN_k
S_k, S_j	Sequence numbers at the SN_k and CN_j, respectively
PW_i, S_U	User password and secret key, respectively
UID_i	U_i's unique identity
GID_i	Pseudonym shared among users of the group
GK_i	Group key corresponding with GID_i
r_c, r_a	Random numbers
T_u, T_c	Timestamps
V_T	Verification table
ΔT	Maximum transmission latency
SK_C	Session key
P	A pool of generated identities
||	Concatenation operation
$\oplus$	XOR operation

In terms of execution, our protocol executes in three major phases. The specific details of registration, authentication and parameter update phases are detailed below.

3.1. Registration

In this phase, all users and smart farm sensors are registered at the controller node prior to engaging in mutual verification, key setup and data exchange. The sub-sections that follow describe these procedures in finer detail. In addition, Algorithm 1 gives a summary of the registration process.

Algorithm 1: Sensor node and user registration

Begin

** *Sensor registration* **

(1) *Choose* SID_k

(2) $SN_k \xrightarrow{SID_k} CN_j$

(3) ***if*** $SID_k \in V_T$ ***then***:

(4) Prompt SN_k to submit $SID_k \notin V_T$

(5) ***else:***

(6) Generate r_a & set $S_k = S_j = 0$

(7) Append $\{r_a, SID_k, S_j\}$ to V_T

(8) $CN_j \xrightarrow{\{S_k, r_a\}} SN_k$

(9) Store $\{S_k, r_a\}$ in SN_k's memory

** *User registration* **

(10) Generate PW_i & S_U

(11) Compute U_a

(12) $U_i \xrightarrow{U_a} CN_j$

(13) Select $UID_i \in P$ & assign it to U_i

(14) Using UID_i, retrieve $\{GID_i, GK_i\}$

(15) Extract ϕ_j & generate r_b

(16) Compute C_a, C_b & C_d

(17) Append $\{GID_i, UID_i, r_b\}$ to V_T

(18) $CN_j \xrightarrow{\{GID_i, GK_i, C_b, C_d\}} U_i$

(19) Store $\{GID_i, GK_i, C_b, C_d\}$ in SD_i

End if

End

3.1.1. Sensor Registration

Whenever a new sensor is introduced in the agricultural field, it must undergo registration prior to interacting with the rest of the network entities. To achieve this, the following three steps are carried out over the secure communication channels.

Step 1: The sensor node SN_k selects SID_k as its unique identity, which is then forwarded to the controller node CN_j, as depicted in Figure 2.

Step 2: After obtaining SID_k, CN_j checks whether it exists in its verification table (V_T). Essentially, SN_k is prompted to select a different identity if SID_k already exists in the CN_j verification table. Otherwise, CN_j generates random number ra and initializes the sequence numbers as $S_k = S_j = 0$. These sequence numbers ensure that SN_k and CN_j are always synchronized. Next, CN_j appends value set $\{r_a, SID_k, S_j\}$ to its verification table before forwarding $\{S_k, r_a\}$ to the SN_k.

Step 3: Upon obtaining $\{S_k, r_a\}$ from CN_j, the SN_k safely stores these parameters in its memory. These values will be used in the subsequent mutual authentication phase that is described in Section 3.2 below.

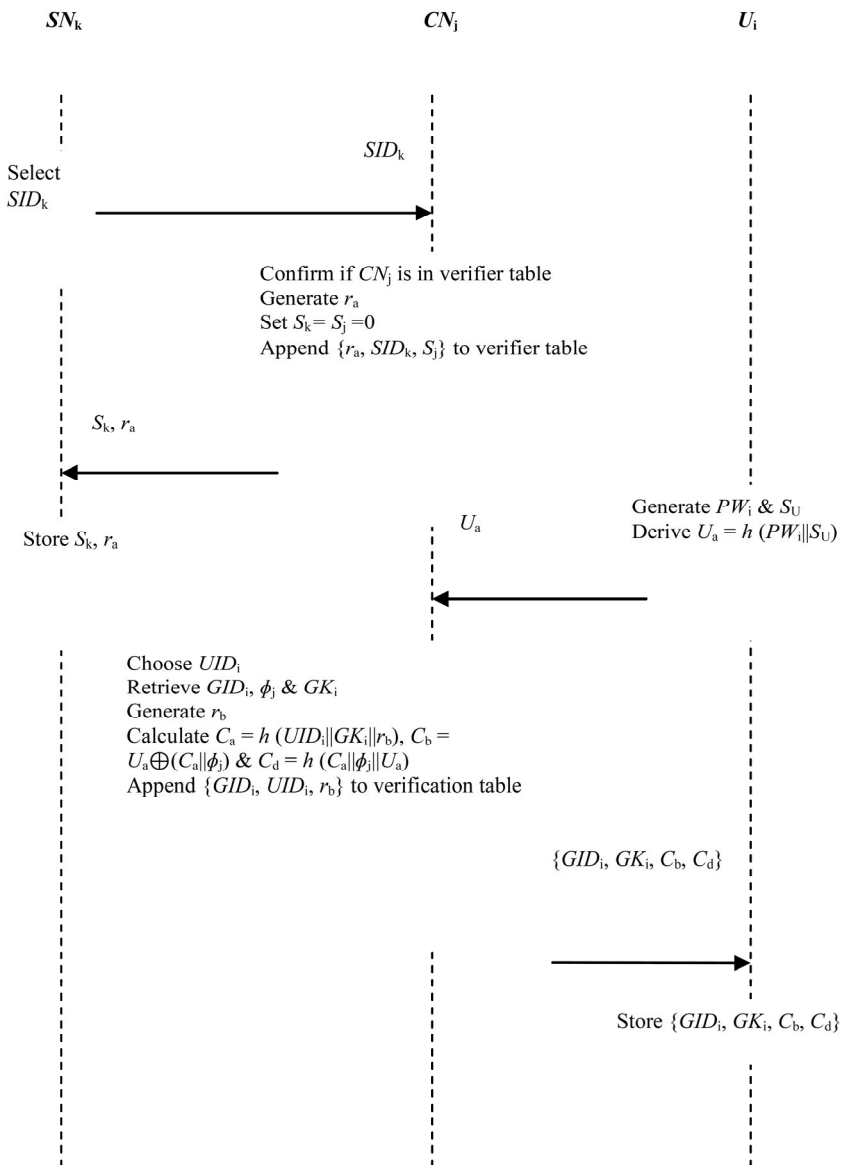

Figure 2. Registration phase.

3.1.2. User Registration

The ultimate objective of this phase is to register all users at the controller node CN_j before being permitted to access information in specific sensor node SN_k. This is accomplished by the execution of the four steps below.

Step 1: User U_i generates password PW_i and private key S_U. Next, parameter $U_a = h(PW_i || S_U)$ is derived before forwarding $\{U_a\}$ to the controller node, as shown in Figure 2.

Step 2: Upon obtaining $\{U_a\}$, CN_j randomly chooses unused $UID_i \in P$ and assigns it to the current U_i, where P is a pool of generated identities (UID_1, UID_2, ... UID_N). Essentially, CN_j maintains a pool of identities from which one is picked and assigned to a new user U_i. Next, CN_j utilizes this UID_i to obtain the equivalent pseudonym GID_i shared among users of the group, ϕ_j and GK_i. Basically, each user's unique identity UID_i is associated with a certain group whose pseudonym is GID_i and group key is GK_i. After the effective retrieval of GID_i and GK_i, CN_j retrieves its secret key ϕ_j used to compute the parameters in the subsequent steps.

Step 3: CN_j chooses random number r_b and proceeds to derive $C_a = h\ (UID_i \mid\mid GK_i \mid\mid r_b)$, $C_b = U_a \oplus (C_a \mid\mid \phi_j)$ and $C_d = h\ (C_a \mid\mid \phi_j \mid\mid U_a)$. Next, CN_j appends $\{GID_i, UID_i, r_b\}$ to its verification table and sends $\{GID_i, GK_i, C_b, C_d\}$ over to user U_i for subsequent login and mutual authentication. This verification table is searched every time new users and sensor nodes are registered so that a given identity is not assigned to more than one network entity.

Step 4: Upon obtaining $\{GID_i, GK_i, C_b, C_d\}$, the user's smart device SD_i stores these parameters in its memory. The user will deploy these security tokens to authenticate with the controller node CN_j as well as the smart farm sensor node SN_k.

3.2. Login and Authentication

When the user wishes to access some particular smart farm sensor SN_k, the two must mutually authenticate each other. In addition, they must negotiate a session key upon successful mutual verification. This key is then utilized to encipher all the exchanged information over the wireless public channels. This is accomplished using the seven steps described below, and summarized in Algorithm 2.

Step 1: User U_i enters password PW_i into his/her smart device SD_i, which then computes $U_a^* = h\ (PW_i \mid\mid S_U)$, $(C_a \mid\mid \phi_j) = C_b \oplus U_a^*$ and $C_d^* = h\ (C_a \mid\mid \phi_j \mid\mid U_a^*)$. Next, it checks if $C_d^* = C_d$, where C_d is the value stored in its memory. Essentially, the login request is rejected if this verification fails. Otherwise, U_i has successfully logged into his/her smart device SD_i.

Step 2: The smart device SD_i generates random number r_c and derives $U_b = \phi_j \oplus h\ (GID_i \mid\mid GK_i \mid\mid T_u)$, $U_c = (r_c \mid\mid SID_k \mid\mid) \oplus h(GID_i \mid\mid \phi_j \mid\mid C_a \mid\mid T_u)$ and $U_d = h\ (r_c \mid\mid C_a \mid\mid GID_i \mid\mid T_u)$. At the end, the smart device SD_i composes authentication message $Auth_u = \{GID_i, U_b, U_c, U_d, T_u\}$, which is forwarded to CN_j, as shown in Figure 3.

Step 3: Upon obtaining $Auth_u$, CN_j determines current timestamp T_c that it deploys to validate the received T_u against ΔT by checking if $\mid T_c - T_u \mid \leq \Delta T$. On the condition that T_u is not fresh, the session is aborted. Otherwise, CN_j retrieves GK_i based on the received GID_i. It then computes $\phi_j = U_b \oplus h\ (GID_i \mid\mid GK_i \mid\mid T_u)$ and retrieves user identity UID_i from its verification table. Next, values $C_a = h\ (UID_i \mid\mid GK_i \mid\mid r_b)$, $(r_c \mid\mid SID_k \mid\mid) = U_c \oplus h(GID_i \mid\mid \phi_j \mid\mid C_a \mid\mid T_u)$ and $U_d^* = h\ (r_c \mid\mid C_a \mid\mid GID_i \mid\mid T_u)$ are derived before checking whether the derived U_d^* is equivalent to U_d received in message $Auth_u$. Provided that these two values are dissimilar, the session is aborted immediately. If not, CN_j randomly selects SK_C as the session key, which it utilizes to calculate $C_e = (SK_C \mid\mid UID_i) \oplus h\ (S_j \mid\mid SID_k \mid\mid r_a)$ and $C_f = h\ (SK_C \mid\mid UID_i \mid\mid SID_k \mid\mid S_j \mid\mid r_a)$.

Step 4: CN_j updates random number r_a as $r_a^{new} = h\ (r_a \mid\mid SID_k)$. In addition, it increments the sequence number S_j by 1; that is, $S_j = S_j + 1$. At the end, it constructs authentication message $Auth_c = \{S_j, C_e, C_f\}$ and sends it over to SN_k.

Step 5: Having received $Auth_c$, the SN_k checks if $1 \leq \mid S_j - S_k \mid \leq P$. In a nutshell, the authentication session is aborted when this validation fails. However, if the verification is successful, the SN_k proceeds to derive $(SK_C \mid\mid UID_i) = C_e \oplus h\ (S_k \mid\mid SID_k \mid\mid r_a)$ and $C_f^* = h\ (SK_C \mid\mid UID_i \mid\mid SID_k \mid\mid S_k \mid\mid r_a)$. Next, it confirms whether $C_f^* = C_f$ and terminates the session if this condition is false. Otherwise, $S_a = h\ (UID_i \mid\mid SID_k \mid\mid SK_C \mid\mid S_j)$ is computed. This is followed by updating r_a as $r_a^{new} = h\ (r_a \mid\mid SID_k)$ and setting $S_k = S_j$. At the end, the SN_k composes authentication message $Auth_s = \{SID_k, S_a\}$ which is transmitted towards CN_j.

Step 6: Upon obtaining message $Auth_s$, CN_j derives $S_a^* = h\ (UID_i \mid\mid SID_k \mid\mid SK_C \mid\mid S_j)$ and checks if S_a^* is equivalent to S_a received $Auth_s$. Provided that these two parameters are dissimilar, the authentication session is aborted. However, if the verification succeeds, CN_j calculates $C_g = (SK_C \mid\mid UID_i) \oplus h\ (SID_k \mid\mid r_c \mid\mid C_a \mid\mid \phi_j)$ and $C_h = h\ (SK_C \mid\mid r_c \mid\mid UID_i \mid\mid SID_k)$. Finally, CN_j constructs message $Auth_n = \{C_g, C_h\}$, which is forwarded towards U_i.

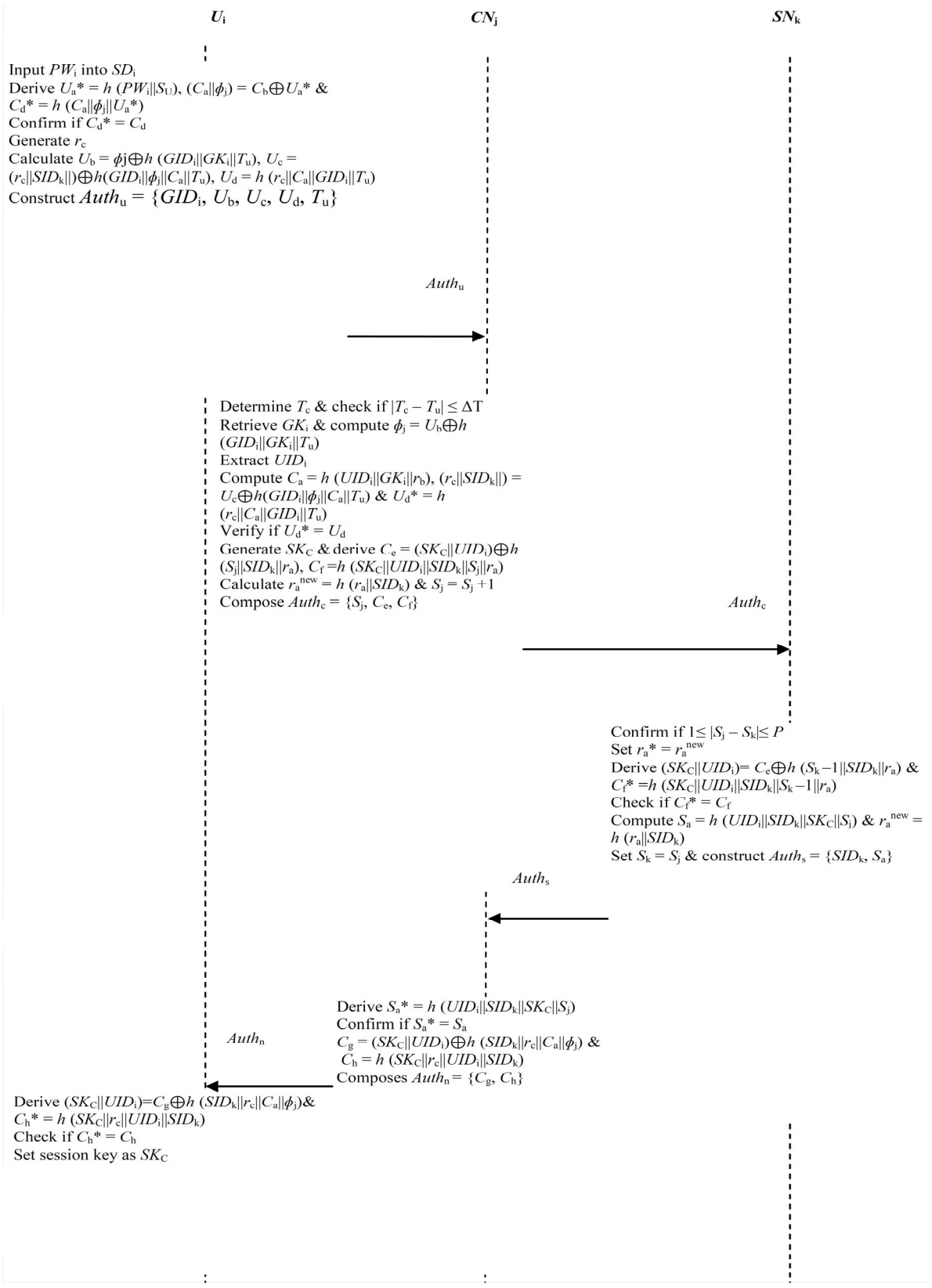

Figure 3. Login and authentication.

Step 7: After receiving message $Auth_n$, the user's smart device SD_i computes $(SK_C||UID_i) = C_g \oplus h\ (SID_k||r_c||C_a||\phi_j)$ and $C_h^* = h\ (SK_C||r_c||UID_i||SID_k)$. Next, the smart device SD_i confirms whether the derived C_h^* is equivalent to C_h received in message

$Auth_n$. Provided that this verification is successful, the authentication among U_i, CN_j and SN_k is considered complete, and SK_C is set as the session key.

Algorithm 2: Login and authentication

Begin

(1) Input PW_i into SD_i
(2) Calculate U_a*, (C_a | | ϕ_j) & C_d*
(3) if C_d* != C_d ***then***:
(4) Reject login request
(5) ***else***
(6) Generate r_c
(7) Compute U_b, U_c & U_d
(8) Construct $Auth_u$
(9) $U_i \xrightarrow{} Auth_u$ CN_j
(10) Determine T_c
(11) ***end if***
(12) ***if*** $|T_c - T_u| \geq \Delta T$ ***then***:
(13) flag $Auth_u$ as replay
(14) ***else***:
(15) Based on GID_i, retrieve GK_i
(16) Derive ϕ_j & retrieve UID_i from V_T
(17) Compute C_a,), (r_c | | SID_k | |) & U_d*
(18) ***end if***
(19) ***if*** U_d* != U_d ***then***:
(20) Halt session
(21) Choose SK_C
(22) Calculate C_e & C_f
(23) Update r_a to $r_a{}^{new}$ & set $S_j = S_j + 1$
(24) Compose $Auth_c$
(25) $CN_j \xrightarrow{} Auth_c$ SN_k
(26) ***end if***
(27) ***if*** $1 \geq |S_j - S_k| \geq P$ ***then***:
(28) Terminate session
(29) ***else***:
(30) Derive (SK_C | | UID_i) & C_f*
(31) ***end if***
(32) ***if*** C_f* != C_f ***then***:
(33) Abort session
(34) ***else***:
(35) Calculate S_a & update r_a to $r_a{}^{new}$
(36) Set $S_k = S_j$
(37) Construct $Auth_s$
(38) $SN_k \xrightarrow{} Auth_s$ CN_j
(39) Derive S_a*
(40) ***end if***
(41) ***if*** S_a* != S_a ***then***:
(42) Halt the session

Algorithm 2: *Cont.*

(43) ***else***:
(44) Compute C_g & C_h
(45) Compose $Auth_n$
(46) $CN_j \xrightarrow{} Auth_n\ U_i$
(47) Calculate (SK_C | | UID_i) & C_h*
(48) ***end if***
(49) ***if*** C_h* != C_h ***then***:
(50) Abort session
(51) ***else***:
(52) Set SK_C as the session key
(53) ***end if***

End

3.3. Parameter Update Phase

The two procedures below are activated upon the compromising of user password PW_i.

Step 1: U_i enters his/her password into SD_i. This is followed by the calculation of $U_a = h\ (PW_i \mid\mid S_U)$, $(C_a \mid\mid \phi_j) = U_a \oplus C_b$ and $C_d^* = h\ (C_a \mid\mid \phi_j \mid\mid U_a)$. Next, it confirms whether the derived C_d^* is equivalent to C_d, stored in its memory. If this verification is unsuccessful, the password change request is denied. However, if the verification is successful, the SD_i inputs new password PW_i^{new}.

Step 2: SD_i calculates $U_a^{new} = h\ (PW_i^{new} \mid\mid S_U)$, $C_b^{new} = U_a^{new} \oplus (C_a \mid\mid \phi_j)$ and $C_d^{new} = h\ (C_a \mid\mid \phi_j \mid\mid U_a^{new})$. At the end, the SD_i substitutes parameter set $\{C_b, C_d\}$ with its updated version $\{C_b^{new}, C_d^{new}\}$ in its memory.

4. Security Analysis

In this part, we carry out both formal and informal security analyses of our scheme. The aim is to demonstrate its semantic robustness and resilience to typical attacks in the precision agriculture environment.

4.1. Formal Security Evaluation

In this sub-section, we utilize the Burrows–Abadi–Needham (BAN) logic to reveal the robustness of the authentication procedures as well as the session key setup between U_i and SN_k. In this proof, we let A and B represent statements, while S and T denote the subjects. The symbols used during the BAN logic proofs are detailed below.

(B): Message B is fresh;
$S \mid\equiv A$: Subject S considers statement A to be true;
$S \mid\sim A$: At some point, subject S sent message A;
S $\mid\triangleleft$ A: S has seen message A;
S $\mid\Rightarrow$ A: S has control over A;
$S \overset{k}{\longleftrightarrow} T$: Subject S and T are sharing key k;
$S \overset{k}{\rightleftharpoons} T$: k is the shared secret between principals S and T;
$\{A\}_k$: Message A is encrypted using key k;
$\langle A_k \rangle$: Secret k is combined with A.

In addition to the above logic symbols, we deployed BAN logic rules below to demonstrate that the authentication among U_i, CN_j and SN_k is carried out in a secure manner.

Message-meaning rule (MMR): $\dfrac{S \mid\equiv S \overset{k}{\leftrightarrow} T, S \triangleleft \{A\}_k}{S \mid\equiv T \mid\sim A}$;

Nonce verification rule (NVR): $\frac{S|\equiv\#(A),S|\equiv T|\sim A}{S|\equiv T|\equiv A}$;

Jurisdiction rule (JR): $\frac{S|\equiv T\Rightarrow A,S|\equiv T|\equiv A}{S|\equiv A}$;

Believe rule (BR): $\frac{S|\equiv(A,B)}{S|\equiv A}$;

Freshness rule (FR): $\frac{S|\equiv\#(A)}{S|\equiv\#(A,B)}$.

To demonstrate that our scheme achieves perfect and protected joint validation between U_i and SN_k, the goals below must be attained.

Goal 1: $U_i|\equiv U_i \xleftrightarrow{SK_C} SN_k$;

Goal 2: $U_i|\equiv SN_k|\equiv U_i \xleftrightarrow{SK_C} SN_k$;

Goal 3: $SN_k|\equiv U_i \xleftrightarrow{SK_C} SN_k$;

Goal 4: $SN_k|\equiv U_i|\equiv U_i \xleftrightarrow{SK_C} SN_k$.

During the login and authentication process, four messages are exchanged among U_i, CN_j and SN_k. For effective proofs, these messages are converted into idealized format as detailed below.

$\boldsymbol{Auth_u}$ ($U_i \rightarrow CN_j$): $\{GID_i, U_b, U_c, U_d, T_u\}$;

Idealized form: $(U_i \xleftrightarrow{r_c} CN_j, SID_k)_{U_i \xleftrightarrow{C_a||GK_i} CN_j}, < GID_i, SID_k, r_c, T_u >_{U_i \xleftrightarrow{C_a||GK_i} CN_j}$;

$\boldsymbol{Auth_c}$ ($CN_j \rightarrow SN_k$): $\{S_j, C_e, C_f\}$;

Idealized form: $(CN_j \xleftrightarrow{SK_C} SN_k,\ UID_i)_{CN_j \xleftrightarrow{r_a} SN_k}, < UID_i,\ SID_k,\ CN_j \xleftrightarrow{SK_C} SN_k, S_j >_{CN_j \xleftrightarrow{r_a} SN_k}$;

$\boldsymbol{Auth_s}$ ($SN_k \rightarrow CN_j$): $\{SID_k, S_a\}$;

Idealized form: $< UID_i, SID_k, SN_k \xleftrightarrow{SK_C} CN_j >_{SN_k \xleftrightarrow{r_a} CN_j}$;

$\boldsymbol{Auth_n}$ ($CN_j \rightarrow U_i$): $\{C_g, C_h\}$;

Idealized form: $(CN_j \xleftrightarrow{SK_C} U_i)_{U_i \xleftrightarrow{C_a} CN_j}, < UID_i, SID_k, CN_j \xleftrightarrow{SK_C} U_i >_{U_i \overset{r_c}{\rightleftharpoons} CN_j}$.

Next, the following preliminary state assumptions (PSA_i) are made regarding our proposed scheme. The freshness and legitimacy of the values in these assumptions are deployed to demonstrate the robustness of the authentication procedures and the session key setup between Ui and SNk (as seen in proofs P1 to P24 that follow).

PSA_1: $CN_j\ |\equiv \#(T_u)$;

PSA_2: $CN_j\ |\equiv \#(r_c)$;

PSA_3: $SN_k\ |\equiv \#(SK_C)$;

PSA_4: $SN_k\ |\equiv \#(SK_C)$;

PSA_5: $U_i|\equiv U_i \xleftrightarrow{C_a||GK_i} CN_j$;

PSA_6: $CN_j|\equiv U_i \xleftrightarrow{C_a||GK_i} CN_j$;

PSA_7: $SN_k|\equiv SN_{k\,i} \xleftrightarrow{r_a} CN_j$;

PSA_8: $CN_j|\equiv SN_{k\,i} \xleftrightarrow{r_a} CN_j$;

PSA_9: $U_i|\equiv CN_j\ |\Rightarrow U_i \xleftrightarrow{SK_C} SN_k$;

PSA_{10}: $SN_k|\equiv CN_j\ |\Rightarrow U_i \xleftrightarrow{SK_C} SN_k$.

Using these notation, rules, initial state assumptions and idealized messages, we execute the rigorous BAN logic proofs to demonstrate the existence of robust and secured authentication among U_i, CN_j and SN_k.

Based on message $Auth_u$, we obtain the following proof.

P1: $CN_j \triangleleft (U_i \xleftrightarrow{r_c} CN_j, SID_k)_{U_i \xleftrightarrow{C_a||GK_i} CN_j}$.

According to PSA_6, the MMR is applied to yield P2.

P2: $CN_j\ |\equiv U_i \xleftrightarrow{r_c} CN_j, SID_k$.

On the other hand, the freshness rule is used in PSA_6 to obtain P3.
P3: $CN_j \mid \equiv \#\ (UID_i, SID_k, GID_i, T_u, U_i \xleftrightarrow{r_c} CN_j)$.
Based on P2 and P3, we apply the NVR to obtain P4.
P4: $CN_j \mid \equiv U_i \mid \equiv\ (UID_i, SID_k, GID_i, T_u, U_i \xleftrightarrow{r_c} CN_j)$.
According to $Auth_c$, we obtain P5 as follows.
P5: $SN_k \lhd (CN_j \xleftrightarrow{SK_C} SN_k, UID_i)_{CN_j \xleftrightarrow{r_a} SN_k}$.
Based on PSA_7 and P5, the message-meaning rule is applied to yield P6.
P6: $SN_k \mid \equiv CN_j \mid \sim (CN_j \xleftrightarrow{SK_C} SN_k, UID_i)$.
In accordance with PSA_3, the freshness rule is effected to obtain P7.
P7: $SN_k \mid \equiv \#\ (UID_i, SID_k, CN_j \xleftrightarrow{SK_C} SN_k, S_j)$.
From P6 and P7, we utilize the nonce-verification rule to obtain P8.
P8: $SN_k \mid \equiv CN_j \mid \equiv (UID_i, SID_k, CN_j \xleftrightarrow{SK_C} SN_k, S_j)$.
Based on $Auth_s$, we obtain P9 as follows.
P9: $CN_j \lhd < UID_i, SID_k, SN_k \xleftrightarrow{SK_C} CN_j >_{SN_k \xleftrightarrow{r_a} CN_j}$.
In accordance with PSA_8 and P9, the message-meaning rule is applied to yield P10.
P10: $CN_j \mid \equiv SN_k \mid \sim (UID_i, SID_k, SN_k \xleftrightarrow{SK_C} CN_j)$.
The application of the freshness rule to P10 results in P11 as follows.
P11: $CN_j \mid \equiv SN_k \mid \equiv (UID_i, SID_k, SN_k \xleftrightarrow{SK_C} CN_j)$.
Based on message $Auth_n$, we obtain P12.
P12: $U_i \lhd (CN_j \xleftrightarrow{SK_C} U_i)_{U_i \xleftrightarrow{C_a} CN_j}$.
In accordance with PSA_5 and P12, we utilize the message-meaning rule to obtain P13.
P13: $U_i \mid \equiv CN_j \mid \sim (CN_j \xleftrightarrow{SK_C} U_i)$.
The application of the freshness rule on PSA_4 results in P14.
P14: $U_i \mid \equiv \#\ (UID_i, SID_k, CN_j \xleftrightarrow{SK_C} U_i)$.
On the other hand, NVR is applied to P13 and P14 to obtain the following.
P15: $U_i \mid \equiv CN_j \mid \equiv (UID_i, SID_k, CN_j \xleftrightarrow{SK_C} U_i)$.
Similarly, the belief rule is applied to P6 and P7 to obtain P16.
P16: $SN_k \mid \equiv (CN_j \xleftrightarrow{SK_C} SN_k)$.
From P8, we utilize the belief rule to obtain P17.
P17: $SN_k \mid \equiv CN_j \mid \equiv (CN_j \xleftrightarrow{SK_C} U_i)$.
On the other hand, the belief rule is used in P11 to obtain P18.
P18: $CN_j \mid \equiv SN_k \mid \equiv (SN_k \xleftrightarrow{SK_C} CN_j)$.
Similarly, the belief rule is utilized in P13 and P14 to obtain P19.
P19: $U_i \mid \equiv (CN_j \xleftrightarrow{SK_C} U_i)$.
However, the application of the belief rule in P15 yields P20.
P20: $U_i \mid \equiv CN_j \mid \equiv (CN_j \xleftrightarrow{SK_C} U_i)$.
Based on PSA_{10} and P16, we obtain P21.
P21: $SN_k \mid \equiv (U_i \xleftrightarrow{SK_C} SN_k)$; hence, Goal 3 is realized.
On the other hand, we obtain P22 from PSA_{10} and P17 as follows.
P22: $SN_k \mid \equiv U_i \mid \equiv (U_i \xleftrightarrow{SK_C} SN_k)$ and hence Goal 4 is attained.
From PSA_9, P18 and P19, we obtain the following.
P23: $U_i \mid \equiv (SN_k \xleftrightarrow{SK_C} U_i)$, achieving Goal 1.
Finally, from PSA_9, P18 and P20, we obtain P24 as follows.
P24: $U_i \mid \equiv SN_k \mid \equiv (SN_k \xleftrightarrow{SK_C} U_i)$, attaining Goal 2.

4.2. Informal Security Analysis

In this sub-section, we formulate and prove a number of lemmas to show that our scheme is secure under the adversarial capabilities in the Canetti–Krawczyk attack model. These capabilities are described in [53].

Lemma 1. *MitM and forgery threats are prevented.*

Proof. Let us assume that adversary Ã is wants to forge validation messages so as to deceive any unsuspecting receivers. To achieve this goal, messages $Auth_u = \{GID_i, U_b, U_c, U_d, T_u\}$, $Auth_c = \{S_j, C_e, C_f\}$, $Auth_s = \{SID_k, S_a\}$ and $Auth_n = \{C_g, C_h\}$ are captured. Next, attempts are made to modify these messages. Here, $U_b = \phi_j \oplus h\ (GID_i || GK_i || T_u)$, $U_c = (r_c || SID_k ||) \oplus h(GID_i || \phi_j || C_a || T_u)$, $U_d = h\ (r_c || C_a || GID_i || T_u)$, $C_e = (SK_C || UID_i) \oplus h\ (S_j || SID_k || r_a)$, $C_f = h\ (SK_C || UID_i || SID_k || S_j || r_a)$, $S_a = h\ (UID_i || SID_k || SK_C || S_j)$, $C_g = (SK_C || UID_i) \oplus h\ (SID_k || r_c || C_a || \phi_j)$ and $C_h = h\ (SK_C || r_c || UID_i || SID_k)$. Evidently, adversary $\tilde{A}$ does not have access to keys such as GK_i and SK_C, sequence number S_j, unique identities SID_k and UID_i, and random numbers r_c and r_a. Since these two threats are easily mitigated in our protocol, both integrity and confidentiality are upheld. □

Lemma 2. *Robust mutual verification is executed.*

Proof. During the login and validation phase, entities U_i, SD_i, CN_j and SN_k mutually verify each other. To gain access to his/her smart device SD_i, it checks whether $C_d^* = C_d$, where C_d is the value stored in its memory. Basically, the login request is rejected if this verification unsuccessful. On the other hand, CN_j validates SD_i by deriving $U_d^* = h\ (r_c || C_a || GID_i || T_u)$ and confirming if $U_d^* = U_d$. For its part, the SN_k verifies CN_j by deriving $C_f^* = h\ (SK_C || UID_i || SID_k || S_k - 1 || r_a)$ and confirming whether $C_f^* = C_f$. Similarly, CN_j validates SN_k through the computation of $S_a^* = h\ (UID_i || SID_k || SK_C || S_j)$ and confirmation of whether $S_a^* = S_a$. Finally, SD_i authenticates CN_j by calculating $C_h^* = h\ (SK_C || r_c || UID_i || SID_k)$ and checking if $C_h^* = C_h$. For all these scenarios, the authentication session is halted upon verification failure. □

Lemma 3. *Our security technique prevents replay threats.*

Proof. In our scheme, we make use of timestamps, sequence numbers and random numbers to curb these threats. During the authentication procedures, messages $Auth_u = \{GID_i, U_b, U_c, U_d, T_u\}$, $Auth_c = \{S_j, C_e, C_f\}$, $Auth_s = \{SID_k, S_a\}$ and $Auth_n = \{C_g, C_h\}$ are exchanged. Here, $U_b = \phi_j \oplus h\ (GID_i || GK_i || T_u)$, $U_c = (r_c || SID_k ||) \oplus h(GID_i || \phi_j || C_a || T_u)$, $C_a = h\ (UID_i || GK_i || r_b)$, $U_d = h\ (r_c || C_a || GID_i || T_u)$, $C_e = (SK_C || UID_i) \oplus h\ (S_j || SID_k || r_a)$, $C_f = h\ (SK_C || UID_i || SID_k || S_j || r_a)$, $S_a = h\ (UID_i || SID_k || SK_C || S_j)$, $C_g = (SK_C || UID_i) \oplus h\ (SID_k || r_c || C_a || \phi_j)$ and $C_h = h\ (SK_C || r_c || UID_i || SID_k)$. Evidently, these messages incorporate timestamp T_u, random numbers r_c, r_b and r_a, and sequence number S_j. Therefore, any replayed message can be effortlessly decerned at the receiver end since these parameters will fail the freshness tests. □

Lemma 4. *Anonymity and untraceability are preserved.*

Proof. The objective of anonymity is to prevent adversaries from discerning the actual identity of the users hinged on the messages transferred across the public channels. On the other hand, untraceability ensures that users are not tracked based on the publicly exchanged messages. In our scheme, messages $Auth_u = \{GID_i, U_b, U_c, U_d, T_u\}$, $Auth_c = \{S_j, C_e, C_f\}$, $Auth_s = \{SID_k, S_a\}$ and $Auth_n = \{C_g, C_h\}$ are exchanged over the public internet. It is

clear that the user's real identity UID_i is never sent in plaintext in all these messages. In fact, it is only the GID_i (pseudonym shared among users of the group) that can be extracted from these messages. Any effort to retrieve UID_i from $C_e = (SK_C || UID_i) \oplus h(S_j || SID_k || r_a)$, $C_f = h(SK_C || UID_i || SID_k || S_j || r_a)$, $S_a = h(UID_i || SID_k || SK_C || S_j)$, $C_g = (SK_C || UID_i) \oplus h(SID_k || r_c || C_a || \phi_j)$ and $C_h = h(SK_C || r_c || UID_i || SID_k)$ will fail due to its encapsulation in other parameters. In addition, there is need to reverse $h(.)$, which is computationally cumbersome. □

Lemma 5. *Our protocol mitigates eavesdropping and sensor node spoofing threats.*

Proof. The ultimate goal of this attack is to derive verification parameter $S_a = h(UID_i || SID_k || SK_C || S_j)$ used to authenticate SN_k at CN_j. This requires that adversary $\tilde{A}$ access identities UID_i and SID_k, session key SK_C and sequence number S_j. Based on Lemma 4, adversary $\tilde{A}$ does not have access to UID_i. Similarly, session key SK_C and sequence number S_j are generated at the CN_j and hence not available to $\tilde{A}$. Therefore, although SID_k can be obtained by eavesdropping on message $Auth_s$, an adversary cannot execute sensor node spoofing. □

Lemma 6. *KSSTI attacks are thwarted and forward secrecy is preserved.*

Proof. To uphold forward key secrecy, it is required that the leakage of long-term secret keys cannot result in the recovery of the previous and subsequent session keys. Let us assume that $\tilde{A}$ has compromised long-term secrets such as S_U and GK_i. The session key in our protocol is generated at CN_j and encapsulated in parameters such as $C_e = (SK_C || UID_i) \oplus h(S_j || SID_k || r_a)$, $C_f = h(SK_C || UID_i || SID_k || S_j || r_a)$, $S_a = h(UID_i || SID_k || SK_C || S_j)$, $C_g = (SK_C || UID_i) \oplus h(SID_k || r_c || C_a || \phi_j)$ and $C_h = h(SK_C || r_c || UID_i || SID_k)$. Evidently, these long-term keys cannot help adversary $\tilde{A}$ in deriving session key SK_C. □

Lemma 7. *Our approach can withstand user impersonation attacks.*

Proof. In the proposed protocol, user U_i constructs message $Auth_u = \{GID_i, U_b, U_c, U_d, T_u\}$, which is transmitted towards CN_j over the public internet. Let us assume that adversary $\tilde{A}$ wants to construct this message so as to impersonate U_i. Here, $U_b = \phi_j \oplus h(GID_i || GK_i || T_u)$, $\phi_j = U_b \oplus h(GID_i || GK_i || T_u)$, $U_c = (r_c || SID_k ||) \oplus h(GID_i || \phi_j || C_a || T_u)$, $C_a = h(UID_i || GK_i || r_b)$ and $U_d = h(r_c || C_a || GID_i || T_u)$. According to Lemma 4, adversary $\tilde{A}$ cannot access UID_i. In addition, group key GK_i cannot be eavesdropped on from the public Internet since it is never sent in messages $Auth_u$, $Auth_c$, $Auth_s$ and $Auth_n$. For the same reason, adversary $\tilde{A}$ does not have access to random numbers r_b and r_c. □

Lemma 8. *Our scheme can withstand physical, stolen device and side-channeling threats.*

Proof. Supposing that an adversary has stolen or captured sensor node SN_k and extracted value set $\{S_k, r_a\}$ stored in it through power analysis. Using the extracted parameters, attempts are made to derive sensor node verification parameter $S_a = h(UID_i || SID_k || SK_C || S_j)$. It is clear that the extracted values cannot help the adversary to derive S_a. Similarly, the physical capture and extraction of value set $\{GID_i, GK_i, C_b, C_d\}$ cannot facilitate the computation of user verification parameter $U_d = h(r_c || C_a || GID_i || T_u)$. □

Lemma 9. *Privileged insider attacks are thwarted.*

Proof. Suppose that some entities in the control node CN_j turns out to be malicious. As such, attempts may be made to capture user password PW_i and use it for malicious

activities. During user registration, U_i sends parameter $U_a = h\ (PW_i \mid\mid S_U)$ to CN_j over secured channels. Although U_a contains PW_i, it is masked in secret key S_U prior to being exposed to one-way hashing operation. Therefore, it is extremely cumbersome for $\tilde{A}$ to extract PW_i from parameter U_a. □

Lemma 10. *Our scheme can withstand password guessing and session hijacking threats.*

Proof. Let us assume that adversary $\tilde{A}$ wants to gain access to user smart device SD_i and thereafter hijack the session and exchange data with other network entities. In our scheme, after entering $PW_i^{\tilde{A}}$ into SD_i, it computes parameters $U_a^* = h\ (PW_i^{\tilde{A}} \mid\mid S_U)$, $(C_a \mid\mid \phi_j) = C_b \oplus U_a^*$ and $C_d^* = h\ (C_a \mid\mid \phi_j \mid\mid U_a^*)$. Next, it checks if $C_d^* = C_d$, where C_d is the value stored in SD_i's memory. Since $PW_i^{\tilde{A}} \neq PW_i$, the adversary $\tilde{A}$ will fail the $C_d^* \triangleleft C_d$ check. □

Lemma 11. *Denial of service and de-synchronization attacks are mitigated.*

Proof. Supposing that adversary $\tilde{A}$ is determined to de-sync the controller node CN_j so that the sensor nodes are denied access to CN_j. To avert this, our scheme uses sequence numbers S_k and S_j to ensure that SN_k and CN_j are always synchronized. During the authentication procedures, message $Auth_c = \{S_j, C_e, C_f\}$ is sent by CN_j towards the SN_k. Here, $C_e = (SK_C \mid\mid UID_i) \oplus h\ (S_j \mid\mid SID_k \mid\mid r_a)$ and $C_f = h\ (SK_C \mid\mid UID_i \mid\mid SID_k \mid\mid S_j \mid\mid r_a)$. Similarly, message $Auth_s = \{SID_k, S_a\}$ is forwarded from the SN_k towards CN_j, where $S_a = h\ (UID_i \mid\mid SID_k \mid\mid SK_C \mid\mid S_j)$. It is clear that these messages contain sequence number S_j. □

Lemma 12. *Scalability is enhanced in the proposed scheme.*

Proof. In our scheme, the network entities do not rely solely on the controller node to generate and distribute all the identities and keying parameters. For instance, during the registration phase, the sensor node SN_k selects SID_k as its unique identity, which is then forwarded to the controller node CN_j. Similarly, user U_i generates password PW_i and private key S_U, and computes value $U_a = h\ (PW_i \mid\mid S_U)$ before forwarding $\{U_a\}$ to the controller node. This lack of total dependency on CN_j for derivation and distribution of identities and keying parameters imply that more users and sensors can be added without overwhelming the controller node. □

5. Comparative Performance Evaluations

In the majority of the security techniques, communication costs, offered functionalities, energy consumption levels and computation overheads are frequently utilized to evaluate their performance. Therefore, we utilize these four performance measures to evaluate our scheme as described below.

5.1. Computation Overheads

The implementation details of our scheme involved a laptop with 4 GB of RAM, running on an Intel processor with 2.4 Ghz of clock frequency. On the other hand, the operating system installed in this machine is Ubuntu 22.04 LTS. Under these operational conditions, the Pairing-Based Cryptography (PBC) library is used to yield the execution time for various cryptographic primitives as follows: elliptic curve scalar multiplications ($T_{pm} \approx 3.53$ ms), one-way hashing ($T_h \approx 0.128$ ms), elliptic curve point addition ($T_{pa} \approx 0.026$ ms), fuzzy extraction ($T_{fe} \approx 3.53$ ms), t-degree univariate polynomial evaluation ($T_{pe} \approx 16.28$ ms), bilinear pairing ($T_b \approx 27.52$ ms), modular exponentiation

($T_e \approx 0.275$ ms) and symmetric encryption/decryption ($T_{en} \approx 0.028$ ms). During the login and verification procedures, SD_i executes $8T_h$, while SN_k executes $5T_h$.

On the other hand, the controller node CN_j executes $10T_h$ operations, and hence, the cumulative computation overhead in our scheme is $23T_h$. Table 2 presents the comparison of this computation overhead with other related schemes.

Table 2. Computation overheads.

Scheme	Computations			Total (ms)
	User/Smart Device	Sensor	Controller/Gateway	
[6]	$4T_h + 2T_{pm}$	$4T_h + T_{pm}$	$7T_h + T_{pm}$	16.04
[12]	$4T_h + 5T_{pm} + T_{pa}$	$3T_h + 4T_{pm} + 2T_{pa}$	$2T_h + 3T_{pm} + T_{pa}$	43.62
[16]	-	$9T_h + 4T_{pm}$	$9T_h + 4T_{pm}$	30.54
[18]	-	$7T_h + 6T_{pm} + 2T_{PA} + T_{pe}$	$7T_h + 6T_{pm} + 2T_{PA} + T_{pe}$	76.82
[30]	$13T_h + 4T_{pm} + T_{pa} + T_{fe}$	$9T_H + 4T_{pm} + T_{pa}$	$12T_h + 6T_{pm} + 2T_{pa}$	57.41
[48]	$9T_h + T_{fe} + T_{en}$	$3T_h + T_{en}$	$5T_h + 5T_{en}$	5.91
[49]	$11T_h + T_{fe}$	$6T_h$	$13T_h$	7.37
[50]	$14T_h$	$9T_h + T_{en}$	$20T_h + T_{en}$	5.56
[51]	$3T_{pm}$	$4T_{pm}$	-	24.71
[52]	$7T_h + T_{pm} + T_{en} + T_{fe}$	$6T_h + T_{pm} + T_{en}$	$2T_h$	12.54
Proposed	$8T_h$	$5T_h$	$10T_h$	2.94

As demonstrated in Figure 4, the approach in [18] incurs the heaviest computation costs of 76.82 ms. This is followed by the security methods in [6,12,16,30,48–52] with computation overheads of 57.41 ms, 43.62 ms, 30.54 ms, 16.04 ms, 12.54 ms, 7.37 ms, 5.91 ms and 5.56 ms, respectively. The high execution durations of these schemes is attributed to the computationally extensive scalar multiplications, fuzzy extractions and encryptions.

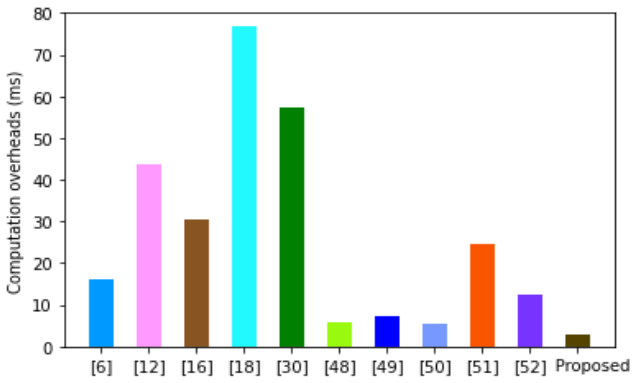

Figure 4. Computation overheads.

However, the proposed scheme involves only efficient one-way hashing and XOR operations. This explains its low computation costs. High computation overheads translate to increased operational delays, which can result in packet drops. These frequent packet drops can potentially interfere with the proper transmission and reception of messages exchanged in precision agriculture. Since the proposed scheme yields the lowest computation overheads, it exhibits the lowest latencies and hence achieves the greatest packet delivery ratio.

5.2. Communication Overheads

In this part, we utilize the sizes of the messages transferred during the login and authentication procedures to obtain the cumulative communication overheads of the developed protocol. To accomplish this, we deploy the values in [18] in which the sizes of the passwords, timestamps and sequence numbers, identities, finite group points, hashing output, elliptic curve points and random numbers are 160 bits, 32 bits, 160 bits, 512 bits, 160 bits, 320 bits and 160 bits, correspondingly. Therefore, the sizes of $Auth_{u}$, $Auth_{c}$, $Auth_{s}$ and $Auth_{n}$ are (160 + 160 + 160 + 160 + 32 = 672 bits), (32 + 160 + 160 = 352 bits), (160 + 160 = 320 bits), and (160 + 160 = 320 bits), respectively. As such, our scheme has 1664 bits as its communication costs. Table 3 shows the communication costs of other related schemes, where *HGWN* is the home gateway node, *U* is the user, *SN* is the sensor node, *DR* is the drone, *GSS* is the ground station server, *AG* is the aggregator, *DSP* is the data service provider, *CBC* is the consortium blockchain, *SM* is the service manager and *CA* is the controlling authority.

Table 3. Communication overheads.

Scheme	Message Exchange Details	Size (Bits)
[6]	$U \xleftrightarrow{1152} GWN \xleftrightarrow{1600} SN$	2752
[12]	$SN \xrightarrow{1344} SM \xrightarrow{256} CBC \xrightarrow{1312} SM \xrightarrow{256} DSP \xrightarrow{1088} SN$	4256
[16]	$SN \xrightarrow{928} GWN \xrightarrow{1088} SN \xrightarrow{288} GWN$	2304
[18]	$GSS \xrightarrow{1088} DR \xrightarrow{928} GSS$	2016
[30]	$U \xrightarrow{1088} CN \xrightarrow{1344} SN \xrightarrow{1376} CN \xrightarrow{1984} U$	5792
[48]	$U \xrightarrow{704} HGWN \xrightarrow{640} SN \xrightarrow{384} U$	1728
[49]	$U \xrightarrow{832} GWN \xrightarrow{672} SN \xrightarrow{352} GWN \xrightarrow{512} U$	2368
[50]	$U \xrightarrow{672} SN \xrightarrow{1344} CA \xrightarrow{800} SN \xrightarrow{960} U$	3776
[51]	$U \xrightarrow{100} AG \xleftrightarrow{320} SN \xleftrightarrow{320} AG \xrightarrow{320} U$	1060
[52]	$U \xrightarrow{320} GWN \xrightarrow{352} SN \xrightarrow{320} GWN \xrightarrow{320} U$	1312
Proposed	$U \xrightarrow{672} CN \xrightarrow{352} SN \xrightarrow{320} CN \xrightarrow{320} U$	1664

It is apparent from Figure 5 that the protocol in [30] incurs the highest communication costs of 5792 bits. This is followed by the schemes in [6,12,16,18,48–50], the proposed protocol, and [51,52] with communication costs of 4256 bits, 3776 bits, 2752 bits, 2368 bits, 2304 bits, 2016 bits, 1728 bits, 1664 bits, 1312 bits and 1060 bits, respectively. Although the schemes in [51,52] incur relatively lower communication costs, they have high computation overheads and are vulnerable to numerous attacks, as shown in Table 4.

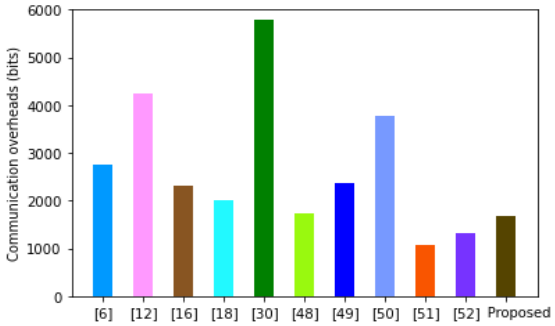

Figure 5. Communication overheads.

Table 4. Energy consumption levels.

Scheme	Energy (mJ)
[6]	0.106
[12]	0.288
[16]	0.202
[18]	0.507
[30]	0.379
[48]	0.039
[49]	0.049
[50]	0.037
[51]	0.163
[52]	0.083
Proposed	0.019

Therefore, although our protocol incurs relatively high communication overheads, it offers perfect security at the lowest computation overheads. It is therefore suitable for addressing security issues in precision agriculture.

5.3. Energy Consumption

In this sub-section, we estimate the power consumed by our scheme during the login and authentication process. The power consumption (in mJ) is given by the product of voltage in volts, current (in milliamps, mA) drawn in active mode and the time in seconds. That is,

$$\textit{Power consumption} = \textit{supply voltage (in V)} \times \textit{current (in mA)} \times \textit{execution time (in S)},$$

where execution time is equivalent to the computation overheads in Table 2 above.

According to [54], this current is 2.2 mA, while the supply voltage is 3V. Using these values, the energy consumption of our scheme is 0.019 mJ. Table 4 presents the comparisons of energy consumption of our scheme against other related protocols.

Based on the graphs in Figure 6, the protocol in [18] exhibits the heaviest amount of energy consumption of 0.507 mJ. This is followed by the schemes in [6,12,16,30,48–52] with energy consumption levels of 0.379 mJ, 0.288 mJ, 0.202 mJ, 0.163 mJ, 0.106 mJ, 0.083 mJ, 0.049 mJ, 0.039 mJ and 0.037 mJ.

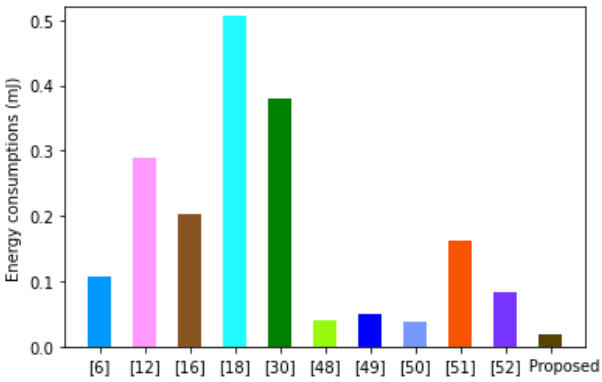

Figure 6. Energy consumption levels.

However, the proposed scheme consumes only 0.019 mJ of energy, which is the lowest. Since most of the PA sensor devices are limited in terms of energy, our scheme is the best suited for deployment in these sensors.

5.4. Supported Functionalities

The proposed protocol supports a wide range of security and privacy characteristics which are key in precision agriculture. In addition, it has been demonstrated to be robust against numerous security threats. Table 5 presents the comparative evaluation of the features of our scheme against other related protocols.

Table 5. Security characteristics.

	[12]	[49]	[50]	[51]	[52]	[30]	[6]	[18]	[16]	[48]	**Proposed**
Security features:											
F1	√	√	√	√	√	√	√	×	√	√	√
F2	×	√	√	×	√	√	√	×	√	√	√
F3	√	√	√	√	√	√	√	√	√	√	√
F4	√	√	√	√	√	√	√	√	×	√	√
Robust against the following:											
F5	×	×	×	×	×	√	√	√	√	×	√
F6	×	√	×	√	×	√	√	√	√	√	√
F7	×	×	×	√	√	×	×	√	×	×	√
F8	×	√	√	×	√	√	√	×	×	√	√
F9	×	×	×	×	×	×	×	×	×	×	√
F10	×	×	×	√	×	×	×	√	×	×	√
F11	√	√	√	√	√	√	√	√	√	√	√
F12	×	×	×	×	×	×	×	×	×	×	√
F13	√	√	√	√	√	√	√	√	√	√	√
F14	×	×	×	×	√	×	×	×	×	×	√
F15	√	×	√	√	√	√	√	√	√	√	√
F16	×	√	√	×	√	√	√	√	√	√	√
F17	×	√	√	×	×	×	√	√	√	×	√
F18	√	√	×	√	√	√	√	×	√	√	√
F19	×	√	√	√	×	√	√	√	√	√	√

F1—Anonymity; F2—Untraceability; F3—Authentication; F4—Perfect key secrecy; F5—Side-channeling; F6—Physical capture; F7—Eavesdropping; F8—Offline guessing; F9—Spoofing; F10—Forgery; F11—Replay; F12—Session hijacking; F13—Impersonation; F14—De-synchronization; F15—MitM; F16—Privileged insider; F17—KSSTI; F18—DoS; F19—Stolen smart device; √ Supported; × Not supported or not considered.

It is evident from Table 5 that the security protocol in [12] supports only seven features, rendering it the most vulnerable. Conversely, the protocols in [50,51] support 11 features each, while the schemes in [16,18,48,49,52] support 12 features each. For its part, the protocol in [30] supports 13 features, while the scheme in [6] offers support for 14 features. However, our proposed scheme supports all 19 of the security features. Overall, the proposed scheme has been demonstrated to offer robust security at relatively low communication costs and the least computation costs, as well as the lowest energy consumption. Since the majority of the devices in precision agriculture are resource-constrained, our scheme is the most suitable this environment. Our scheme adopts the conventional communication framework typical in existing farming systems, where we have central controllers,

users and sensor nodes. In addition, the proposed scheme deploys only one-way hashing functions, which are lightweight. As such, no additional hardware is required for extensive computations. Consequently, our protocol is easy to integrate into existing farming systems such as drones and other IoT devices.

6. Conclusions

The need to increase the food supply in the face of limited resources for the ever-growing global population has seen the incorporation of smart technologies in agriculture. Sensors in precision agriculture collect information such as water content, soil moisture, humidity and temperature, which are then analyzed to facilitate decision making. These technologies have been shown to result in increased agricultural yields at low labor and resources. In spite of these positive contributions of smart agronomy, the amalgamation of a wide range of sensors and technologies coupled with the deployment of open wireless internet for message exchange results in numerous security threats. Although many solutions have been presented in the literature, security vulnerabilities and high resource requirements in most of these schemes render them unsuitable for deployment in resource-limited PA devices. The developed scheme is demonstrated to mitigate the majority of these security threats, such as packet replay, MitM, DoS, de-synchronization, privileged insider, impersonation and spoofing attacks. From the performance perspective, our technique consumes the least energy and computational resources while incurring relatively lower communication overheads. As such, our protocol can reduce operational costs and improve the resilience of agricultural IoT networks. Future research in this area will involve further improvements in the obtained communication costs so as to make it bandwidth-friendly. In addition, real-world testing and various case studies may be carried out against this scheme.

Author Contributions: All the authors have contributed equally to this article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Torky, M.; Hassanein, A.E. Integrating Blockchain and the Internet of Things in Precision Agriculture: Analysis, Opportunities, and Challenges. *Comput. Electron. Agric.* **2020**, *178*, 105476. [CrossRef]
2. Fathy, C.; Ali, H.M. A Secure IoT-Based Irrigation System for Precision Agriculture Using the Expeditious Cipher. *Sensors* **2023**, *23*, 2091. [CrossRef]
3. Vangala, A.; Das, A.K.; Mitra, A.; Das, S.K.; Park, Y. Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 904–919. [CrossRef]
4. Munaganuri, R.K.; Rao, Y.N. PAMICRM: Improving Precision Agriculture through Multimodal Image Analysis for Crop Water Requirement Estimation Using Multidomain Remote Sensing Data Samples. *IEEE Access* **2024**, *12*, 52815–52836. [CrossRef]
5. Lanucara, S.; Praticò, S.; Pioggia, G.; Di Fazio, S.; Modica, G. Web-Based Spatial Decision Support System for Precision Agriculture: A Tool for Delineating Dynamic Management Unit Zones (MUZs). *Smart Agric. Technol.* **2024**, *8*, 100444. [CrossRef]
6. Rangwani, D.; Sadhukhan, D.; Ray, S.; Khan, M.K.; Dasgupta, M. An Improved Privacy Preserving Remote User Authentication Scheme for Agricultural Wireless Sensor Network. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4218. [CrossRef]
7. Abduljabbar, Z.A.; Nyangaresi, V.O.; Jasim, H.M.; Ma, J.; Hussain, M.A.; Hussien, Z.A.; Aldarwish, A.J.Y. Elliptic Curve Cryptography-Based Scheme for Secure Signaling and Data Exchanges in Precision Agriculture. *Sustainability* **2023**, *15*, 10264. [CrossRef]
8. Raj, M.; Harshini; Gupta, S.; Atiquzzaman, M.; Rawlley, O.; Goel, L. Leveraging Precision Agriculture Techniques Using UAVs and Emerging Disruptive Technologies. *Energy Nexus* **2024**, *14*, 100300. [CrossRef]

9. Sonavane, S.M.; Prashantha, G.R.; Nikam, P.D.; Mayuri, A.V.R.; Chauhan, J.; Sountharrajan, S.; Bavirisetti, D.P. Optimizing QoS and Security in Agriculture IoT Deployments: A Bioinspired Q-Learning Model with Customized Shards. *Heliyon* **2024**, *10*, e24224. [CrossRef] [PubMed]
10. You, M.; Kim, Y.; Kim, J.; Seo, M.; Son, S.; Shin, S.; Lee, S. FuzzDocs: An Automated Security Evaluation Framework for IoT. *IEEE Access* **2022**, *10*, 102406–102420. [CrossRef]
11. Bouzidi, M.; Gupta, N.; Cheikh, F.A.; Shalaginov, A.; Derawi, M. A Novel Architectural Framework on IoT Ecosystem, Security Aspects and Mechanisms: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 101362–101384. [CrossRef]
12. Luo, F.; Huang, R.; Xie, Y. Hybrid Blockchain-Based Many-to-Many Cross-Domain Authentication Scheme for Smart Agriculture IoT Networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 101946. [CrossRef]
13. Mutlaq, K.A.-A.; Nyangaresi, V.O.; Omar, M.A.; Abduljabbar, Z.A.; Abduljaleel, I.Q.; Ma, J.; Al Sibahee, M.A. Low Complexity Smart Grid Security Protocol Based on Elliptic Curve Cryptography, Biometrics and Hamming Distance. *PLoS ONE* **2024**, *19*, e0296781. [CrossRef] [PubMed]
14. Huang, H.; Zhou, S.; Lin, J.; Zhang, K.; Guo, S. Bridge the Trustworthiness Gap amongst Multiple Domains: A Practical Blockchain-Based Approach. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020.
15. Wu, H.-T.; Tsai, C.-W. An Intelligent Agriculture Network Security System Based on Private Blockchains. *J. Commun. Netw.* **2019**, *21*, 503–508. [CrossRef]
16. Vangala, A.; Sutrala, A.K.; Das, A.K.; Jo, M. Smart Contract-Based Blockchain-Envisioned Authentication Scheme for Smart Farming. *IEEE Internet Things J.* **2021**, *8*, 10792–10806. [CrossRef]
17. Wang, L.; Xu, L.; Zheng, Z.; Liu, S.; Li, X.; Cao, L.; Li, J.; Sun, C. Smart Contract-Based Agricultural Food Supply Chain Traceability. *IEEE Access* **2021**, *9*, 9296–9307. [CrossRef]
18. Bera, B.; Vangala, A.; Das, A.K.; Lorenz, P.; Khan, M.K. Private Blockchain-Envisioned Drones-Assisted Authenti-Cation Scheme in IoT-Enabled Agricultural Environment. *Comput. Stand. Interfaces* **2022**, *80*, 103567. [CrossRef]
19. Xue, L.; Huang, H.; Xiao, F.; Wang, W. A Cross-Domain Authentication Scheme Based on Cooperative Blockchains Functioning with Revocation for Medical Consortiums. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2409–2420. [CrossRef]
20. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Guizani, M. Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954. [CrossRef]
21. Yang, Y.; Wei, L.; Wu, J.; Long, C.; Li, B. A Blockchain-Based Multidomain Authentication Scheme for Conditional Privacy Preserving in Vehicular Ad-Hoc Network. *IEEE Internet Things J.* **2022**, *9*, 8078–8090. [CrossRef]
22. Bagga, P.; Sutrala, A.K.; Das, A.K.; Vijayakumar, P. Blockchain-Based Batch Authentication Protocol for Internet of Vehicles. *J. Syst. Arch.* **2021**, *113*, 101877. [CrossRef]
23. Nyangaresi, V.O.; Jasim, H.M.; Mutlaq, K.A.A.; Abduljabbar, Z.A.; Ma, J.; Abduljaleel, I.Q.; Honi, D.G. A Symmetric Key and Elliptic Curve Cryptography-Based Protocol for Message Encryption in Unmanned Aerial Vehicles. *Electronics* **2023**, *12*, 3688. [CrossRef]
24. Ali, R.; Pal, A.K.; Kumari, S.; Karuppiah, M.; Conti, M. A Secure User Authentication and Key-Agreement Scheme Using Wireless Sensor Networks for Agriculture Monitoring. *Future Gener. Comput. Syst.* **2018**, *84*, 200–215. [CrossRef]
25. Chen, M.; Lee, T.-F.; Pan, J.-I. An Enhanced Lightweight Dynamic PseudonymIdentity Based Authentication and Key AgreementScheme Using Wireless Sensor Networks for Agriculture Monitoring. *Sensors* **2019**, *19*, 1146. [CrossRef] [PubMed]
26. Alyahya, S.; Khan, W.U.; Ahmed, S.; Marwat, S.N.K.; Habib, S. Cyber Secure Framework for Smart Agriculture: Robust and Tamper-Resistant Authentication Scheme for IoT Devices. *Electronics* **2022**, *11*, 963. [CrossRef]
27. Bothe, A.; Bauer, J.; Aschenbruck, N. RFID-Assisted Continuous User Authentication for IoT-Based Smart Farming. In Proceedings of the 2019 IEEE International Conference on RFID Technology and Applications (RFID-TA), Pisa, Italy, 25–27 September 2019.
28. Hussien, Z.A.; Abdulmalik, H.A.; Hussain, M.A.; Nyangaresi, V.O.; Ma, J.; Abduljabbar, Z.A.; Abduljaleel, I.Q. Lightweight Integrity Preserving Scheme for Secure Data Exchange in Cloud-Based IoT Systems. *Appl. Sci.* **2023**, *13*, 691. [CrossRef]
29. Panda, S.S.; Jena, D.; Mohanta, B.K.; Ramasubbareddy, S.; Daneshmand, M.; Gandomi, A.H. Authentication and Key Management in Distributed IoT Using Blockchain Technology. *IEEE Internet Things J.* **2021**, *8*, 12947–12954. [CrossRef]
30. Vangala, A.; Das, A.K.; Lee, J.-H. Provably Secure Signature-based Anonymous User Authentication Protocol in an Internet of Things-enabled Intelligent Precision Agricultural Environment. *Concurr. Comput.* **2021**, *35*, e6187. [CrossRef]
31. Hassan, B.; Alsanad, A.A.; Ullah, I.; Amin, N.U.; Khan, M.A.; Uddin, M.I.; Wu, J.M.T. A Cost Effective Identi-Ty-Based Authentication Scheme for Internet of Things-Enabled Agriculture. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4275243. [CrossRef]
32. Harbi, Y.; Aliouat, Z.; Refoufi, A.; Harous, S.; Bentaleb, A. Enhanced Authentication and Key Management Scheme for Securing Data Transmission in the Internet of Things. *Ad Hoc Netw.* **2019**, *94*, 101948. [CrossRef]
33. Yuan, E.; Wang, L.; Cheng, S.; Ao, N.; Guo, Q. A Key Management Scheme Based on Pairing-Free Identity Based Digital Signature Algorithm for Heterogeneous Wireless Sensor Networks. *Sensors* **2020**, *20*, 1543. [CrossRef]

34. Mishra, D.; Dharminder, D.; Yadav, P.; Sreenivasa Rao, Y.; Vijayakumar, P.; Kumar, N. A Provably Secure Dynamic ID-Based Authenticated Key Agreement Framework for Mobile Edge Computing without a Trusted Party. *J. Inf. Secur. Appl.* **2020**, *55*, 102648. [CrossRef]
35. Andola, N.; Prakash, S.; Gahlot, R.; Venkatesan, S.; Verma, S. An Enhanced Smart Card and Dynamic ID Based Remote Multi-Server User Authentication Scheme. *Cluster Comput.* **2022**, *25*, 3699–3717. [CrossRef]
36. Gupta, D.S.; Islam, S.K.H.; Obaidat, M.S.; Vijayakumar, P.; Kumar, N.; Park, Y. A Provably Secure and Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for IIoT Environments. *IEEE Syst. J.* **2021**, *15*, 1732–1741. [CrossRef]
37. Liu, J.; Liu, R.; Lai, Y. Risk-Based Dynamic Identity Authentication Method Based on the UCON Model. *Secur. Commun. Netw.* **2022**, *2022*, 1–13. [CrossRef]
38. Sibahee, A.; Nyangaresi, M.A.; Abduljabbar, V.O.; Luo, Z.A.; Zhang, C.; Ma, J. Two-Factor Privacy Preserving Protocol for Efficient Authentication in Internet of Vehicles Networks. *IEEE Internet Things J.* **2023**, *11*, 14253–14266. [CrossRef]
39. Li, Y.; Cheng, Q.; Shi, W. Security Analysis of a Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for IIoT Environments. *Secur. Commun. Netw.* **2021**, *2021*, 1–6. [CrossRef]
40. Li, X.; Niu, J.; Bhuiyan, M.Z.A.; Wu, F.; Karuppiah, M.; Kumari, S. A Robust ECC-Based Provable Secure Authen-Tication Protocol with Privacy Preserving for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3599–3609. [CrossRef]
41. Eddine, M.S.; Ferrag, M.A.; Friha, O.; Maglaras, L. EASBF: An Efficient Authentication Scheme over Blockchain for Fog Computing-Enabled Internet of Vehicles. *J. Inf. Secur. Appl.* **2021**, *59*, 102802. [CrossRef]
42. Tomar, A.; Tripathi, S. Blockchain-Assisted Authentication and Key Agreement Scheme for Fog-Based Smart Grid. *Clust. Comput.* **2022**, *25*, 451–468. [CrossRef]
43. Itoo, S.; Khan, A.A.; Kumar, V.; Alkhayyat, A.; Ahmad, M.; Srinivas, J. CKMIB: Construction of Key Agreement Protocol for Cloud Medical Infrastructure Using Blockchain. *IEEE Access* **2022**, *10*, 67787–67801. [CrossRef]
44. Jia, X.; Luo, M.; Wang, H.; Shen, J.; He, D. A Blockchain-Assisted Privacy-Aware Authentication Scheme for Internet of Medical Things. *IEEE Internet Things J.* **2022**, *9*, 21838–21850. [CrossRef]
45. Shuai, M.; Xiong, L.; Wang, C.; Yu, N. A Secure Authentication Scheme with Forward Secrecy for Industrial Internet of Things Using Rabin Cryptosystem. *Comput. Commun.* **2020**, *160*, 215–227. [CrossRef]
46. Fan, Q.; Chen, J.; Deborah, L.J.; Luo, M. A Secure and Efficient Authentication and Data Sharing Scheme for Internet of Things Based on Blockchain. *J. Syst. Arch.* **2021**, *117*, 102112. [CrossRef]
47. Mutlaq, K.A.A.; Nyangaresi, V.O.; Omar, M.A.; Abduljabbar, Z.A. Symmetric Key Based Scheme for Veri-Fication Token Generation in Internet of Things Communication Environment. In *EAI International Conference on Applied Cryptography in Computer and Communications*; Springer Nature: Cham, Switzerland, 2022; pp. 46–64.
48. Khalid, H.; Hashim, S.J.; Ahmad, S.M.S.; Hashim, F.; Chaudhary, M.A. Robust Multi-Gateway Authentication Scheme for Agriculture Wireless Sensor Network in Society 5.0 Smart Communities. *Agriculture* **2021**, *11*, 1020. [CrossRef]
49. Kumar, R.; Singh, S.; Singh, D.; Kumar, M.; Gill, S.S. A Robust and Secure User Authentication Scheme Based on Multifactor and Multi-gateway in IoT Enabled Sensor Networks. *Secur. Priv.* **2024**, *7*, e335. [CrossRef]
50. Irshad, A.; Alreshoodi, M. SEMS-5G: A Secure and Efficient Multi-Server Authentication Scheme for 5G Networks. *IEEE Access* **2024**, *12*, 49062–49077. [CrossRef]
51. Taji, K.; Ghanimi, F. Enhancing Security and Privacy in Smart Agriculture: A Novel Homomorphic Signcryption System. *Results Eng.* **2024**, *22*, 102310. [CrossRef]
52. Itoo, S.; Khan, A.A.; Ahmad, M.; Idrisi, M.J. A Secure and Privacy-Preserving Lightweight Authentication and Key Exchange Algorithm for Smart Agriculture Monitoring System. *IEEE Access* **2023**, *11*, 56875–56890. [CrossRef]
53. Nyangaresi, V.O. Provably Secure Authentication Protocol for Traffic Exchanges in Unmanned Aerial Vehicles. *High-Confid. Comput.* **2023**, *3*, 100154. [CrossRef]
54. Bahache, A.N.; Chikouche, N.; Mezrag, F. Authentication Schemes for Healthcare Applications Using Wireless Medical Sensor Networks: A Survey. *SN Comput. Sci.* **2022**, *3*, 382. [CrossRef]