sustainability

MDPI

Article

Elliptic Curve Cryptography-Based Scheme for Secure Signaling and Data Exchanges in Precision Agriculture

Zaid Ameen Abduljabbar [1,2,3,*], Vincent Omollo Nyangaresi [4], Hend Muslim Jasim [1], Junchao Ma [5,*], Mohammed Abdulridha Hussain [1,2], Zaid Alaa Hussien [6] and Abdulla J. Y. Aldarwish [1]

1 Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq
2 Technical Computer Engineering Department, AL-Kunooze University College, Basrah 61001, 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 & Technology, Bondo 40601, Kenya
5 College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
6 Information Technology Department, Management Technical College, Southern Technical University, Basrah 61005, Iraq
* Correspondence: zaid.ameen@uobasrah.edu.iq (Z.A.A.); majunchao@sztu.edu.cn (J.M.)

Abstract: Precision agriculture encompasses automation and application of a wide range of information technology devices to improve farm output. In this environment, smart devices collect and exchange a massive number of messages with other devices and servers over public channels. Consequently, smart farming is exposed to diverse attacks, which can have serious consequences since the sensed data are normally processed to help determine the agricultural field status and facilitate decision-making. Although a myriad of security schemes has been presented in the literature to curb these challenges, they either have poor performance or are susceptible to attacks. In this paper, an elliptic curve cryptography-based scheme is presented, which is shown to be formally secure under the Burrows–Abadi–Needham (BAN) logic. In addition, it is semantically demonstrated to offer user privacy, anonymity, unlinkability, untraceability, robust authentication, session key agreement, and key secrecy and does not require the deployment of verifier tables. In addition, it can withstand side-channeling, physical capture, eavesdropping, password guessing, spoofing, forgery, replay, session hijacking, impersonation, de-synchronization, man-in-the-middle, privileged insider, denial of service, stolen smart device, and known session-specific temporary information attacks. In terms of performance, the proposed protocol results in 14.67% and 18% reductions in computation and communication costs, respectively, and a 35.29% improvement in supported security features.

Keywords: Agriculture 4.0; precision agriculture; privacy; smart farming; security

check for updates

Citation: 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. https://doi.org/10.3390/su151310264

Academic Editors: Mourade Azrour, Azidine Guezzaz, Imad Zeroual, Azeem Irshad, Jamal Mabrouki, Said Benkirane and Shehzad Ashraf Chaudhry

Received: 9 May 2023
Revised: 23 June 2023
Accepted: 26 June 2023
Published: 28 June 2023

1. Introduction

Many economies in developing countries are dependent on agriculture as a source of income and contributions to gross domestic product (GDP) [1]. However, the majority of the farming practices are based on experience and ad hoc insights of the farmers. Consequently, there is little control on the agricultural produce quantity and hence financial profits. Fortunately, precision agriculture (PA) and the Internet of Things (IoT) can be deployed to address these issues [2,3]. As explained in [4], PA is part of Agriculture 3.0 in which farm yields are regularly monitored. In addition, PA involves automation and the application of information technology (IT) to improve farm output. In Agriculture 4.0, also referred to as smart agriculture or smart farming, additional technologies such as drones, artificial intelligence (AI), blockchain, big data, wireless sensor networks (WSN), and robotics are incorporated in agriculture. In PA, a number of sensors are deployed, such as radiation, air humidity, optimal, soil moisture, and ground sensors. According to [5], intelligent precision agriculture (IPA) encompasses the deployment of numerous IoT devices and drones to monitor agricultural surroundings. To boost productivity in the face of limited resources and protection from disasters, traditional agronomy needs to be replaced with smart agronomy [6]. As discussed in [7], there are fraud risks in the agricultural sector, especially concerning beverage and food packaging. Therefore, agricultural organizations require ideal certification of their products since these risks can impact negatively on the health of their consumers.

The smart devices deployed in PA and IPA exchange a massive number of messages. Therefore, insecure communication channels among IoT devices, unmanned aerial vehicles (UAVs), or drones can expose smart farming to diverse attacks [5,8]. For instance, Wi-Fi de-authentication and denial of device (DoS) can be launched on Raspberry Pi-based smart farms [9]. This can have serious consequences as the sensed data are normally processed to help determine the agricultural field status and facilitate decision-making, which may involve taking measures to maintain or enhance the farm status [10]. These attacks can also target drones deployed to monitor field conditions such as irrigation, spraying of pesticides, pollination, and planting of seeds [11]. On their part, WSNs offer monitoring, sensing, and a continuous supply of information regarding climatic conditions such as the chemical content of the soil, air humidity, temperature, light, water quality, and soil moisture. These parameters are then utilized to boost productivity, both qualitatively and quantitatively. According to [12], WSNs facilitate monitoring, data collection, and control of agricultural systems and hence ensure efficiency, minimal packet losses and economic overheads, better network control, and increased scalability and flexibility. However, threats such as interference, masquerading, interception, and message alteration can compromise these networks and harm crop production and other monitored agricultural practices [6]. The authors in [13] pointed out that issues such as sufficient energy resource utilization and secure data transmission are yet to be solved in WSN. This is because of the usage of open wireless networks during data transfers [14], which can potentially compromise the integrity, confidentiality, and authenticity of the exchanged data.

To address the above issues, there is a need for robust authentication and access control to secure the internet of drones, WSNs, IoT, and agricultural monitoring [15–17]. For instance, sufficient user authentication ensures that external users can use their mobile devices to securely access real-time data from the deployed agricultural smart devices [18,19]. There is also a need for robust source authentication, message authentication, and entity authentication.

1.1. Contributions

- A lightweight authentication scheme based on elliptic curve cryptography is developed for secure message exchange among the communicating smart devices in precision agriculture.
- Formal security analysis is carried out using BAN logic to demonstrate that a session key is derived from enciphering the exchanged data between the farmers and the agricultural service providers.
- Extensive semantic analysis is executed to show that the proposed scheme can withstand side-channeling, physical capture, eavesdropping, password guessing, spoofing, forgery, replay, session hijacking, impersonation, de-synchronization, man-in-the-middle, privileged insider, denial of service, stolen smart device, and known session-specific temporary information attacks. In addition, this protocol is demonstrated to support user privacy, anonymity, unlinkability, untraceability, robust authentication, session key agreement, and key secrecy and does not require the deployment of verifier tables.
- An elaborate performance evaluation is carried out to show that our scheme yields 14.67% and 18% reductions in computation and communication costs, respectively, and a 35.29% improvement in supported security features.

1.2. Problem Definition and Motivation

In precision agriculture, information technology plays a critical role in ensuring that farming activities obtain exact requirements, which boosts health, productivity, and agricultural outputs. In this way, environmental protection, sustainability, and profitability are assured in smart farms. On the flip side, the public channels deployed for message exchanges make these networks vulnerable to numerous attacks such as eavesdropping, message falsification, DoS, replay, MitM, impersonation, drones capture, ephemeral secret leakage (ESL), privileged insider, and physical smart devices capture attacks. Proper user and device authentication is one of the most promising solutions to these security and privacy challenges. In addition, communication attributes such as untraceability, unlinkability, anonymity, and user privacy need to be assured. For instance, the secrecy of trading transactions among farmers and agricultural firms needs to be upheld.

1.3. Security Requirements

Owing to the open communication channels deployed in smart agriculture, adversaries can hijack the session, take control of the communication process, and execute other malicious activities. Therefore, a secure authentication protocol should be resilient against a myriad of attacks. In addition, it should fulfill the following privacy and security requirements.

Untraceability and unlinkability: It should be cumbersome for the adversary to trace or link some captured messages to a particular network entity.

Robust authentication: To prevent illegitimate entities from joining the network or accessing the agricultural services and smart devices, all the entities must be validated.

Session key negotiation: Immediately after successful mutual authentication procedures, the communicating parties should agree on the session key to encipher the exchanged messages.

Anonymity and privacy: The real identities of the communicating parties should never be exchanged in plain text over public channels. This is to prevent attackers from eavesdropping them across the communication channel. This goes a long way in preserving the privacy of these parties.

Key secrecy: The session key should be computed in a manner that will make it cumbersome for the attacker to deploy the captured session key for the current communication process to derive the keys used in the previous or subsequent communication procedures.

Resistant to attacks: It should be difficult for the attacker to compromise the network and its smart devices through side-channeling, physical capture, eavesdropping, password guessing, spoofing, forgery, replay, session hijacking, impersonation, de-synchronization, man-in-the-middle (MitM), privileged insider, DoS, stolen smart device, and known session-specific temporary information (KSSTI) attacks.

1.4. Threat Modeling

In this paper, the adversary is assumed to have all the capabilities in the Dolev–Yao (DY) as well as Canetti and Krawczyk (CK) threat models. In the DY threat model, an attacker Ψ is capable of intercepting, altering, deleting, and injecting bogus messages into the communication channel. However, in the CK threat model, an adversary Ψ can compromise secret parameters, private keys, and session states that can be obtained from devices' memory. In addition, the communicating entities are assumed to be untrustworthy, and Ψ can physically capture the IoT devices and extract the secrets in their memories through power analysis. Using the extracted secrets, further attacks, such as impersonations, can be launched.

The rest of this paper is structured as follows: Section 2 discusses the related work, while Section 3 presents the proposed scheme. On the other hand, Section 4 discusses the security analysis of our scheme, while Section 5 presents its performance evaluation. Finally, Section 6 concludes this paper and provides some research directions.

2. Related Work

Many schemes have been developed to enhance security in the smart farm environment. For example, a novel private blockchain-based authentication scheme is presented in [5]. However, this protocol fails to protect against de-synchronization and session hijacking attacks. Similarly, blockchain-based schemes were developed in [20–24]. Although blockchain offers traceability, integrity protection, and shareability in the agricultural environment, such as agri-food supply chains, it has high storage and computation overheads [25]. Based on signatures, the authors of [18] present a three-factor user authentication protocol. Unfortunately, this scheme cannot prevent attacks such as eavesdropping and session hijacking. On the other hand, an identity-based scheme was introduced in [26]. Nevertheless, this technique is vulnerable to stolen smart cards, sensor node spoofing, impersonation, and stolen verifier attacks [27]. In addition, it cannot provide backward key secrecy. To address these challenges, two protocols were developed in [27]. Unfortunately, the authentication and password change phases of these schemes are inefficient [28]. To offer privacy protection, a remote user authentication protocol was presented in [6]. However, this scheme cannot withstand attacks such as eavesdropping, de-synchronization, and spoofing.

Based on a public-key-based cryptosystem, an authentication scheme was developed in [29]. Although this approach protects against MitM and replay attacks, it cannot withstand privileged insider, user impersonation, and ephemeral secret leakage (ESL) attacks [5]. In addition, it does not include biometric change and user device revocation phases. The signature-based privacy-preserving protocol in [30] can address some of these issues. However, it is still susceptible to ESL attacks and cannot assure the untraceability and anonymity of the communicating parties [5]. Similarly, the protocol in [31] does not provide user and device anonymity since their internet protocol (IP) addresses incorporated in messages are exchanged publicly. In addition, it has high computation overheads due to the utilization of public key cryptography for its digital signatures and certificates [32]. Moreover, it is prone to replay, physical device capture, MitM, user and device impersonation, and attacks. On its part, the scheme in [33] cannot protect against user anonymity violation, user impersonation, and smart card loss attacks. Similarly, the protocol in [34] is vulnerable to physical sensing device capture, untraceability violation, and smart card loss attacks [5]. Using some bilinear pairing operations, authentication and key establishment protocols were introduced in [35,36]. However, the utilization of pairing operations increased the computation costs of these protocols [37]. Since the trusted authority in [36] has access to user identity and password, it is susceptible to privileged insider attacks. In addition, it cannot withstand replay, disclosure of sensor data, offline password guessing, and stolen smart card and verifier attacks [38]. As such, an improved elliptic curve cryptography (ECC)-based scheme was developed in [38]. However, this protocol has an inefficient and delayed authentication phase. In addition, it is not robust against DoS and replay attacks [39]. Although the protocol in [40] addresses some of these issues, its bilinear pairing operations result in high computation costs [41].

To offer security in a heterogeneous IoT environment, an authentication technique was presented in [42]. Unfortunately, this protocol is vulnerable to physical device capture, privileged insider, and ESL attacks. In addition, it cannot preserve untraceability and anonymity [5]. Similarly, a remote user authentication protocol was developed in [43], which was shown to be lightweight. However, it failed to protect against ESL and privileged insider attacks. It also failed to support untraceability and anonymity [5]. On its part, the scheme in [43] was not resilient against privileged insider and sensor node capture attacks. It also failed to preserve forward key secrecy [6]. The authors in [44,45] designed identity-based signature protocols to protect message exchanges in mobile devices. However, identity-based schemes have key escrow problems [46]. Based on ECC and symmetric key encryption, a security technique was presented in [47]. Although it was shown to be robust against MitM and replay attacks, it was vulnerable to ESL, privileged insider, and user impersonation attacks. It also failed to incorporate device revocation, node addition, and password and user biometric change phases [5]. Similarly, the biometric-based scheme in [48] did not include device revocation, user passwords, and biometric update phases. It was also vulnerable to privileged insider, user impersonation, ESL, DoS, and stolen smart card attacks [49]. On its part, the protocol in [50] was susceptible to DoS attacks and could not offer forward key secrecy [51]. Similarly, the scheme in [52] did not support forward key secrecy and was prone to stolen verifier attacks [53]. As such, an enhanced ECC-based protocol was introduced in [53], while a privacy-preserving scheme was developed in [54]. The scheme in [54] was demonstrated to be resilient against eavesdropping, DoS, masquerade, privileged insider, and forgery attacks. It also supports secret key updates, traceability, and anonymity. However, it cannot withstand MitM attacks [20].

It is evident that numerous schemes have been proposed to improve the security posture in precision agriculture. However, it has been shown that these techniques face a number of security, privacy, and performance challenges. The proposed scheme is shown to solve some of these challenges as described in Sections 4 and 5 below.

3. The Proposed Scheme

The farmer smart devices SD_j and the agricultural service providers ASP_i are the main components of this scheme. As shown in Figure 1, the registration phase occurs over secure channels, while the SD_j and ASP_i exchange the data over the insecure public channels in an ad hoc manner. As such, the goal of the proposed protocol is to enhance the privacy and security of the transmitted information.

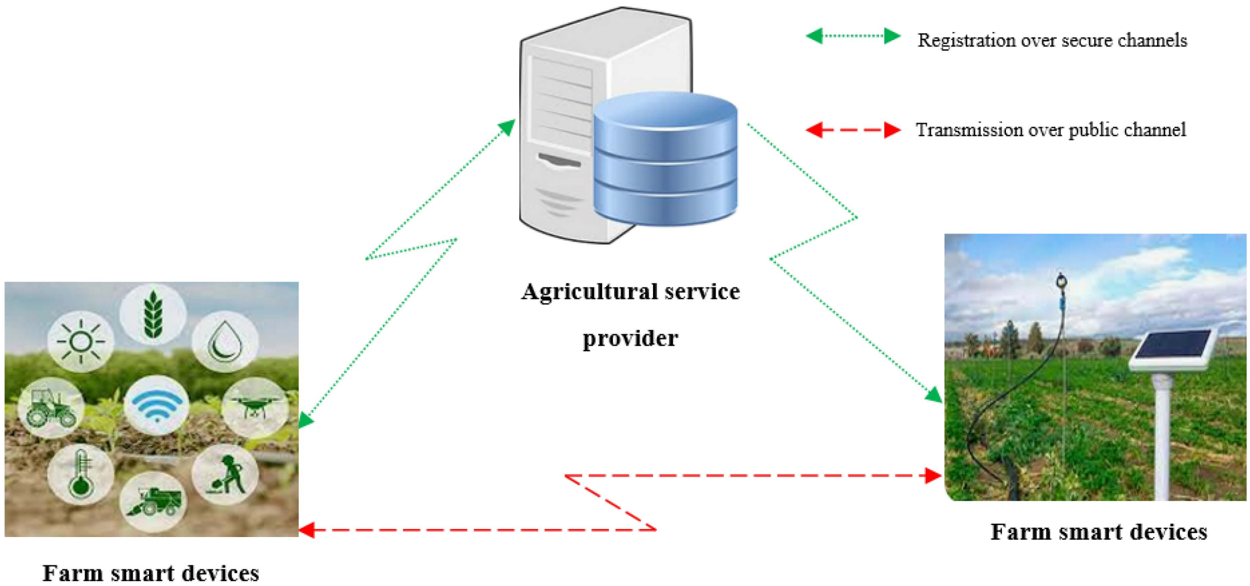

Figure 1. Network model.

The proposed scheme comprises four major phases, which include system initialization, registration, login, and authentication phases. Table 1 presents the notations used throughout this paper.

The subsections below provide detailed descriptions of the various major phases of the proposed scheme.

3.1. System Initialization

In this phase, the agricultural service provide ASP_i executes the following three steps to generate the security parameters that will be utilized during the other three phases. These steps are described in detail, as shown in Figure 2.

Table 1. Deployed symbols.

Symbol	Description
ASP_i	Agricultural service provider i
MK_A	Master key for ASP_i
F_j	Farmer j
SD_j	Smart device for F_j
FID_j	Unique identity for F_j
FPW_j	Login password for F_j
R_i	Random nonce i
$\|\|$	Concatenation operation
P_B	Padding bits
$\oplus$	XOR operation
ϕ_A	Session key computed at ASP_i
ϕ_S	Session key computed at SD_j

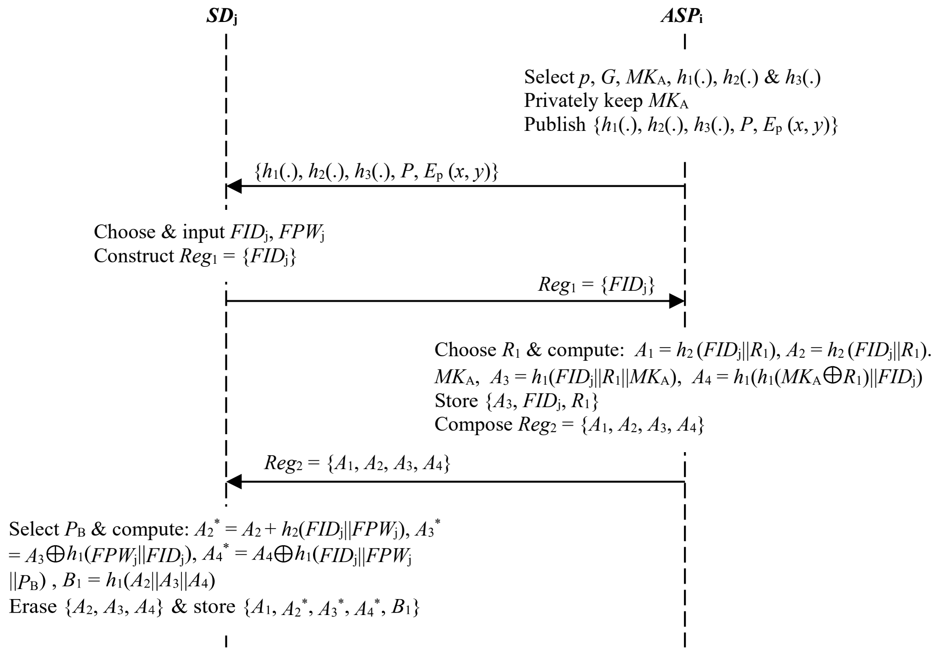

Figure 2. System initialization and registration phases.

Step 1: The ASP_i selects the prime number p, whose length is k-bits. It also chooses some elliptic curve group G whose base point is P and whose order is q.

Step 2: The ASP_i selects a random parameter MK_A from {1, $q - 1$} and deploys it as its master secret key. In addition, it chooses three collision-resistant one-way hashing functions, $h_1(.)$, $h_2(.)$, and $h_3(.)$, where $h_2(.)$ serves as the map-to-point hashing function. Therefore, $h_2(.)$: $\{0,1\}^* \rightarrow$ G, $h_1(.)$: $\{0,1\}^* \rightarrow \{0,1\}^k$, and $h_3(.)$: G $\rightarrow \{0,1\}^k$.

Step 3: Parameter MK_A is secretly retained by the ASP_i, while parameter set $\{h_1(.)$, $h_2(.)$, $h_3(.)$,P, E_p $(x, y)\}$ is publicly made available to all smart devices.

3.2. Registration Phase

It is required that all farmers register with the ASP_i and obtain some security tokens before being allowed to access some services from the ASP_i. This is a four-step process, as described below.

Step 1: The farmer F_j selects a unique identity FID_j and password FPW_j that are input to the SD_j. Next, a registration message $Reg_1 = \{FID_j\}$ is constructed that is forwarded to the ASP_i over secured communication channels.

Step 2: Upon receipt of Reg_1, the ASP_i selects random nonce R_1. Next, it derives security values $A_1 = h_2\ (FID_j\,||\,R_1)$, $A_2 = h_2\ (FID_j\,||\,R_1)$. MK_A, $A_3 = h_1(FID_j\,||\,R_1\,||\,MK_A)$, and $A_4 = h_1(h_1(MK_A \oplus R_1)\,||\,FID_j)$.

Step 3: The ASP_i stores parameter set $\{A_3, FID_j, R_1\}$ in its database for later use during the login and authentication phases. Finally, it constructs registration message $Reg_2 = \{A_1, A_2, A_3, A_4\}$, which is forwarded to F_j over secure channels, as shown in Figure 2.

Step 4: After receiving message Reg_2, the SD_j generates fixed-bit padding parameter P_B. This is followed by the computation of values $A_2{}^* = A_2 + h_2(FID_j\,||\,FPW_j)$, $A_3{}^* = A_3 \oplus h_1(FPW_j\,||\,FID_j)$, $A_4{}^* = A_4 \oplus h_1(FID_j\,||\,FPW_j\,\,||\,P_B)$, and $B_1 = h_1(A_2\,||\,A_3\,||\,A_4)$. Finally, it erases parameter set $\{A_2, A_3, A_4\}$ and stores value set $\{A_1, A_2{}^*, A_3{}^*, A_4{}^*, B_1\}$ in its memory.

3.3. Login

The goal of this phase is to validate the farmer password and unique identity that are input to the smart device SD_j. To accomplish this, the following two steps are executed:

Step 1: The farmer F_j inputs the unique identity FID_j and password FPW_j into the SD_j. Next, the SD_j derives values $A_2 = A_2{}^* - h_2(FID_j\,||\,FPW_j)$, $A_3 = A_3{}^* \oplus h_1(FPW_j\,||\,FID_j)$, and $A_4 = A_4{}^* \oplus h_1(FID_j\,||\,FPW_j\,||\,P_B)$.

Step 2: The SD_j computes $B_1{}^* = h_1(A_2\,||\,A_3\,||\,A_4)$ and verifies if $B_1{}^* \stackrel{?}{=} B_1$. the session is terminated if these two values are not identical. Otherwise, F_j has logged in successfully and can now proceed to the authentication phase.

3.4. Authentication and Key Agreement

In this phase, the farmer F_j, through the SD_j, generates and exchanges a number of security tokens with the agricultural service provider ASP_i, through which these two entities verify one another before the onset of agricultural data exchanges. In addition, the session keys for data encryption are derived as described below.

Step 1: The SD_j generates random nonces R_2 and R_3, where $R_2 \in \{1, q-1\}$ and $R_3 \in \{0,1\}^k$. Next, it derives parameters $B_2 = A_1$. R_2, $B_3 = A_2$. R_2, $B_4 = A_3 \oplus h_3\ (B_3)$, $C_1 = A_4 \oplus R_3$, and $C_2 = h_1(B_2\,||\,B_3\,||\,B_4\,||\,C_1\,||\,R_3\,||\,A_4)$. Lastly, it composes authentication message $Auth_1 = \{B_2, B_4, C_1, C_2\}$, which is transmitted to the ASP_i over public channels, as shown in Figure 3.

Step 2: Upon receiving the authentication message $Auth_1$ message from SD_j, the ASP_i derives $B_3{}^* = MK_A$. B_2 and $A_3{}^{**} = B_4 \oplus h_3\ (B_3{}^*)$. Next, it confirms whether parameter set $\{A_3{}^{**}, FID_j{}^*, R_1{}^*\}$ is in its database. Here, the session is terminated if this verification fails. Otherwise, the ASP_i proceeds to compute values $A_4{}^{**} = h_1(h_1(MK_A \oplus R_1{}^*)\,||\,FID_j{}^*)$ and $R_3{}^* = C_1 \oplus A_4{}^{**}$.

Step 3: The ASP_i derives values $C_2{}^* = h_1(B_2\,||\,B_3{}^*\,||\,B_4\,||\,C_1\,||\,R_3{}^*\,||\,A_4{}^{**})$ and confirms whether $C_2{}^* \stackrel{?}{=} C_2$. The authentication session is essentially terminated if this verification flops. Otherwise, the ASP_i chooses random nonce$R_4 \in \{0,1\}^k$, which is utilized in deriving parameters $C_3 = R_4 \oplus A_4{}^{**}$, session key $\phi_A = h_1(R_3{}^*\,||\,R_4\,||\,B_3{}^*\,||\,A_4{}^{**})$, and $C_4 = h_1(FID_j{}^*\,||\,\phi_A)$. Finally, it constructs an authentication message $Auth_2 = \{C_3, C_4\}$ that it forwards to SD_j over public communication channels.

Step 4: After obtaining message $Auth_2$ from ASP_i, the SD_j derives values $R_4{}^* = C_3 \oplus A_4$, session key $\phi_S = h_1(R_3\,||\,R_4{}^*\,||\,B_3\,||\,A_4)$, and $C_4{}^* = h_1(FID_j\,||\,\phi_S)$. This is followed by the validation of whether $C_4{}^* \stackrel{?}{=} C_4$. The authentication session is aborted if these two parameters are unequal. Otherwise, SD_j deploys ϕ_S as the session key to encipher all the exchanged messages.

3.5. Password Renewal Phase

The proposed scheme allows the farmer F_j to change his/her password FPW_j. This may be prompted by the loss of FPW_j or when they suspect that this password might have been compromised. This is attained by executing the following steps.

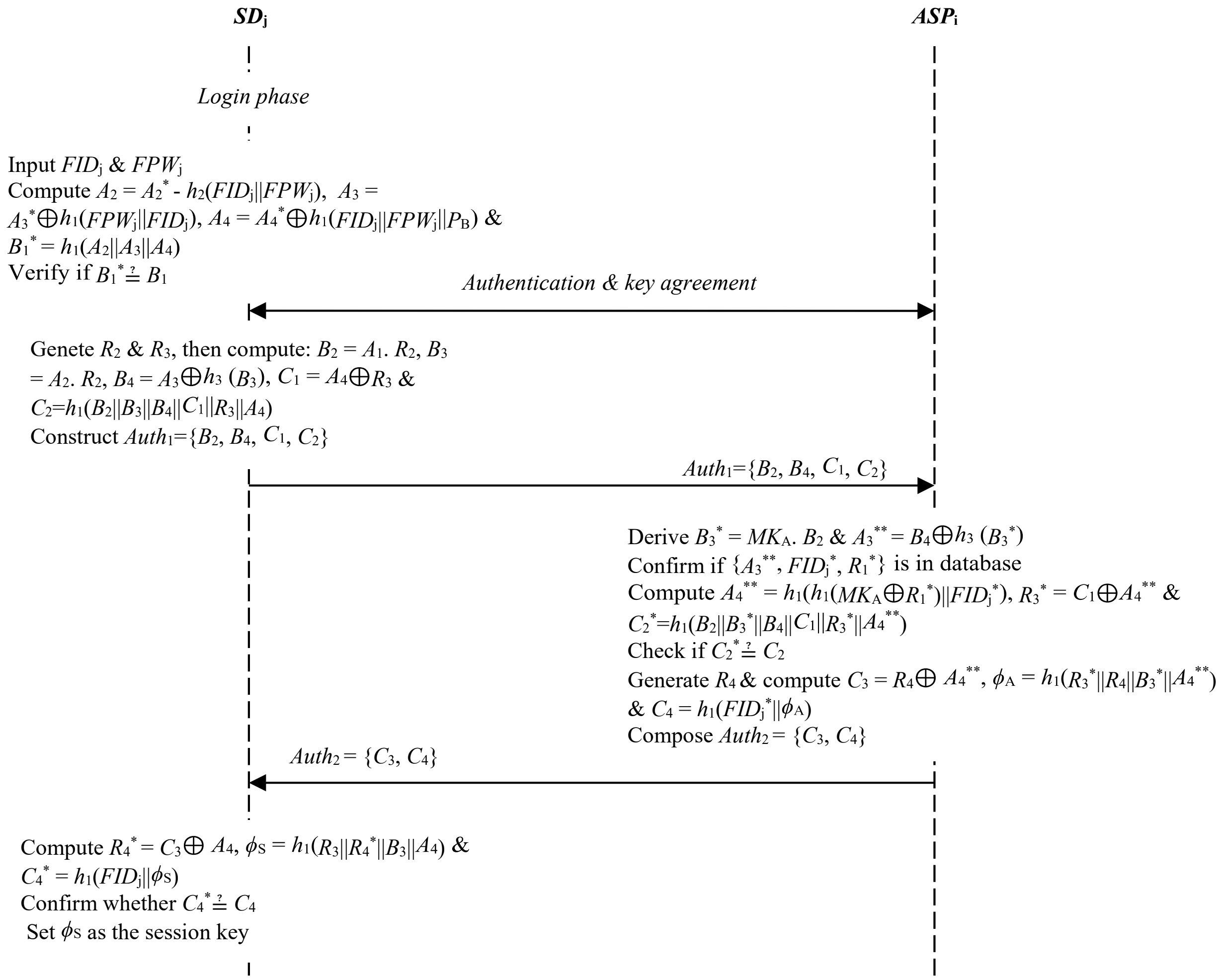

Figure 3. Login, authentication, and key negotiation phase.

Step 1: The farmer F_j inputs the current password FPW_j into the SD_j. Next, SD_j deploys the stored parameter set $\{A_1, A_2^*, A_3^*, A_4^*, B_1\}$ in its memory to derive $A_2 = A_2^* - h_2(FID_j || FPW_j)$, $A_3 = A_3^* \oplus h_1(FPW_j || FID_j)$, and $A_4 = A_4^* \oplus h_1(FID_j || FPW_j || P_B)$.

Step 2: F_j selects the new password FPW_j^*. This is followed by the computation of parameters $A_2^{New} = A_2 + h_2(FID_j || FPW_j^*)$, $A_3^{New} = A_3 \oplus h_1(FPW_j^* || FID_j)$, and $A_4^{New} = A_4 \oplus h_1(FID_j || FPW_j^* || P_B)$.

Step 3: The SD_j erases value set $\{A_2, A_3, A_4, A_2^*, A_3^*, A_4^*\}$ from its memory and stores parameter set $\{A_1, A_2^{New}, A_3^{New}, A_4^{New}, B_1\}$ in its memory.

4. Security Analysis

In this section, the proposed scheme's security and privacy are analyzed using both formal and semantic techniques described below.

4.1. Formal Security Analysis

In this sub-section, we deploy the BAN logic to demonstrate that the farmer F_j and agricultural service provider ASP_i interact to set up a session key between them. This key is then utilized to encipher the data exchanged between these two entities. Suppose that A and B are principals, M and N are statements, and μ is the encryption key. The notations used in this analysis are described below.

$A \mid\equiv M$: A trusts M.

$\{M\}$: Statement M is enciphered using key μ.

$A \triangleleft M$: A receives M.

$<M>_N$: Statement M is combined statement N.
(M): M is fresh.
$A \Rightarrow M$: A has control.
(M, N): Statement M or N is part of (M, N).
$A \overset{\mu}{\leftrightarrow} B$: Principals A and B deploy shared key μ during communication.
$A \mid \sim M$: Principal A once said statement M.
$(M)_h$: Statement M is hashed using hashing function h.
During the formal security verification, the following BAN logic rules are utilized.
Freshness Rule (F-R)
$\frac{A\ believes\ fresh\ M}{A\ believes\ fresh\ (M,N)}$, mathematically represented as $\frac{A\ |\equiv\#\ (M)}{A\ |\equiv\#\ (M,\ N)}$.
The Message-Meaning Rule (M-M-R)
$\frac{A\ believes\ A\overset{\mu}{\leftrightarrow}B,\ A\ sees\ \{M\}_\mu}{A\ believes\ B\ once\ said\ M}$, which can be mathematically expressed as $\frac{A\ |\equiv A\overset{\mu}{\leftrightarrow}B,\ A\triangleleft\{M\}_\mu}{A\ |\equiv B|\sim M}$.
Jurisdiction-Rule (J-R)
$\frac{A\ believes\ B\ control\ M,\ A\ believes\ B\ believes\ M}{A\ believes\ M}$, mathematically denoted as $\frac{A\ |\equiv B \Rightarrow M,\ A\ |\equiv B\ |\equiv M}{A\ |\equiv M}$.
Believe–Rule (B-R)
$\frac{A\ believes\ B\ believes\ (M,\ N)}{A\ believes\ B\ believes\ M}$, also expressed as $\frac{A\ |\equiv B\ |\equiv (M,\ N)}{A\ |\equiv B\ |\equiv M}$.
Nonce Verification Rule (N-V-R)
$\frac{A\ believes\ fresh\ (M),\ A\ believes\ B\ once\ said\ M}{A\ believes\ B\ believes\ M}$, which can be denoted as $\frac{A\ |\equiv \#(M),\ A\ |\equiv B|\sim M}{A|\equiv B|\equiv M}$.

To show the establishment of session key μ between provider ASP_i and farmer F_j, the following two goals are formulated.

Goal 1: $ASP_i \mid\equiv (F_j \overset{\mu}{\leftrightarrow} ASP_i)$.

Goal 2: $F_j \mid\equiv (F_j \overset{\mu}{\leftrightarrow} ASP_i)$.

To achieve this, the following initial assumptions are made:

IA$_1$: $F_j \mid\equiv R_2$;
IA$_2$: $F_j \mid\equiv R_3$;
IA$_3$: $F_j \mid\equiv B_2$;
IA$_4$: $F_j \mid\equiv B_3$;
IA$_5$: $F_j \mid\equiv F_j \overset{A_4}{\leftrightarrow} ASP_i$;
IA$_6$: $F_j \mid\equiv ASP_i \Rightarrow (R_4)$;
IA$_7$: $ASP_i \mid\equiv B_2$;
IA$_8$: $ASP_i \mid\equiv MK_A$;
IA$_9$: $ASP_i \mid\equiv R_4$;
IA$_{10}$: $ASP_i \mid\equiv F_j \overset{A_4}{\leftrightarrow} ASP_i$;
IA$_{11}$: $ASP_i \mid\equiv F_j \Rightarrow (R_3, B_2)$.

In the proposed protocol, two messages are exchanged during the authentication and key agreement phase. These messages include $Auth_1$ and $Auth_2$, transmitted by the SD_j and ASP_i, respectively. For efficient analysis, these messages are transformed into idealized designs, as described below.

$\boldsymbol{SD_j \rightarrow ASP_i}$**:**
$Auth_1 = \{B_2, B_4, C_1, C_2\}$;
Idealized form: $\{B_2, B_4, C_1, C_2, \langle B_3\rangle_{B_2}, \langle A_3\rangle_{B_3}, \langle R_3\rangle_{A_4}\}$.
$\boldsymbol{ASP_i \rightarrow SD_j}$**:**
$Auth_2 = \{C_3, C_4\}$;
Idealized form: $\{C_3, C_4, \langle R_4\rangle_{A_4}\}$.

Next, the above BAN logic notations, rules, and initial state assumptions are deployed to demonstrate that the farmer F_j and the agricultural service provider ASP_i derive and share similar session key μ to encipher the exchanged messages. This procedure proceeds as described below.

Based on $Auth_1$, the following is obtained:

DM_1: $ASP_i \triangleleft \{B_2, B_4, C_1, C_2, \langle B_3\rangle_{B_2}, \langle A_3\rangle_{B_3}, \langle R_3\rangle_{A_4}\}$.
Using *M-M-R*, DM_1, IA_7, and IA_8, DM_2 is yielded.
DM_2: $ASP_i \mid\equiv F_j \mid \sim\{B_2, B_4, C_1, C_2, \langle B_3\rangle_{B_2}, \langle A_3\rangle_{B_3}, \langle R_3\rangle_{A_4}\}$.
Since $C_2^* = h_1(B_2 \mid\mid B_3^* \mid\mid B_4 \mid\mid C_1 \mid\mid R_3^* \mid\mid A_4^{**})$, from DM_2,
DM_3: $ASP_i \mid\equiv F_j \mid \equiv\{B_2, B_4, C_1, C_2, \langle B_3\rangle_{B_2}, \langle A_3\rangle_{B_3}, \langle R_3\rangle_{A_4}\}$.
Using the *B-R* and DM_3, the following is obtained:
DM_4: $ASP_i \mid\equiv F_j \mid \equiv (R_3, B_2)$
Based on IA_{11} and DM_4, we obtain:
DM_5: $ASP_i \mid\equiv (R_3, B_2)$.
On the other hand, the application of *B-R* on DM_5 yields:
DM_6: $ASP_i \mid\equiv (R_3)$ and $ASP_i \mid\equiv (B_2)$.
Based on IA_8 and IA_9, the following is obtained:
DM_7: $ASP_i \mid\equiv MK_A$ and $ASP_i \mid\equiv (R_4)$,
Since $\phi_A = h_1(R_3^* \mid\mid R_4 \mid\mid B_3^* \mid\mid A_4^{**})$, then from DM_6 and DM_7,
DM_8: $ASP_i \mid\equiv (F_j \overset{\mu}{\leftrightarrow} ASP_i)$, hence **Goal 1** is attained.
From $Auth_2$, the following is obtained:
DM_9: $F_j \triangleleft \{C_3, C_4, \langle R_4\rangle_{A_4}\}$.
Using the M-M-R on IA_{10} results in the following:
DM_{10}: $F_j \mid\equiv ASP_i \mid\sim\{C_3, C_4, \langle R_4\rangle_{A_4}\}$.
Based on DM_{10}, IA_5, and IA_9,
DM_{11}: $F_j \mid\equiv ASP_i \mid\equiv R_4$.
The application of the *J-R* on DM_{11} and IA_6 results in the following:
DM_{12}: $F_j \mid\equiv R_4$.
Based on DM_{12} and IA_2–IA_5, we obtain:
DM_{13}: $F_j \mid\equiv (F_j \overset{\mu}{\leftrightarrow} ASP_i)$; therefore, **Goal 2** is accomplished.
The attainment of these two security goals confirms that farmer F_j and agricultural service provider ASP_i strongly trust that they share session key μ for traffic protection.

4.2. Semantic Security Analysis

The objective of this subsection is the formulation and proofing of some hypotheses regarding the supported security features in the proposed scheme.

Hypothesis 1: *Farmer privacy and anonymous communication are achieved.*

Proof: In the proposed scheme, the real identity of the farmer is FID_j. This identity is incorporated in values such as $A_1 = h_2\,(FID_j \mid\mid R_1)$, $A_2 = h_2\,(FID_j \mid\mid R_1)\ MK_A$, $A_3 = h_1(FID_j \mid\mid R_1 \mid\mid MK_A)$, $A_4 = h_1(h_1(MK_A \oplus R_1) \mid\mid FID_j)$, and $C_4 = h_1(FID_j^* \mid\mid \phi_A)$. In all these parameters, FID_j is encapsulated in other parameters before being hashed. During the authentication and key agreement phase, messages $Auth_1 = \{B_2, B_4, C_1, C_2\}$ and $Auth_2 = \{C_3, C_4\}$ are transmitted over public channels. Here, $B_2 = A_1$. R_2, $B_4 = A_3 \oplus h_3\,(B_3)$, $C_1 = A_4 \oplus R_3$, $C_2 = h_1(B_2 \mid\mid B_3 \mid\mid B_4 \mid\mid C_1 \mid\mid R_3 \mid\mid A_4)$, $C_3 = R_4 \oplus A_4^{**}$, and $C_4 = h_1(FID_j^* \mid\mid \phi_A)$. It is evident that of all these parameters, it is only C_4 that directly incorporates farmer identity FID_j. However, this identity is encapsulated in session key ϕ_A before being hashed. Due to the difficulty of reversing the hashing function, it is difficult for adversary Ψ to obtain this identity from message $Auth_2$. □

Hypothesis 2: *Side-channeling and physical attacks are prevented.*

Proof: Suppose that attacker Ψ has physically captured the farmer's smart device SD_j. The objective is to extract the security tokens in its memory to compute the session key $\phi_S = h_1(R_3 \mid\mid R_4^* \mid\mid B_3 \mid\mid A_4)$. During the registration phase, the SD_j stores parameter set $\{A_1, A_2^*, A_3^*, A_4^*, B_1\}$ in its memory. Here, $A_1 = h_2\,(FID_j \mid\mid R_1)$, $A_2^* = A_2 + h_2(FID_j \mid\mid FPW_j)$, $A_3^* = A_3 \oplus h_1(FPW_j \mid\mid FID_j)$, $A_4^* = A_4 \oplus h_1(FID_j \mid\mid FPW_j \mid\mid P_B)$, $B_1 = h_1(A_2 \mid\mid A_3 \mid\mid A_4)$, and $B_3 = A_2.R_2$. As such, although the attacker may have access to value A_4^*, random nonces

R_3 and $R_4{}^*$, as well as parameter B_3, cannot be recovered from SD_j's memory. As such, the derivation of session key ϕ_S flops. □

Hypothesis 3: *This scheme is robust against eavesdropping and password-guessing attacks.*

Proof: Let us assume that adversary Ψ is interested in capturing the farmer's password FPW_j for malicious login into SD_j. To achieve this, an attempt is made to eavesdrop FPW_j from the exchanged messages $Auth_1 = \{B_2, B_4, C_1, C_2\}$ and $Auth_2 = \{C_3, C_4\}$. Here, $B_2 = A_1$. R_2, $A_1 = h_2$ $(FID_j || R_1)$, $B_4 = A_3 \oplus h_3$ (B_3), $C_1 = A_4 \oplus R_3$, $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$, $C_3 = R_4 \oplus A_4{}^{**}$, and $C_4 = h_1(FID_j{}^* || \phi_A)$. Evidently, none of the components of these two messages contains plain-text FPW_j. The only parameters incorporating this password are $A_2 = A_2{}^* - h_2(FID_j || FPW_j)$, $A_3 = A_3{}^* \oplus h_1(FPW_j || FID_j)$, and $A_4 = A_4{}^* \oplus h_1(FID_j || FPW_j || P_B)$, which are never sent directly over the public channels. In addition, FPW_j is encapsulated in other values before being hashed. Due to the difficulty of reversing or colliding the hashing function, any guessing of FPW_j from these parameters will fail. □

Hypothesis 4: *This scheme upholds unlinkability and untraceability.*

Proof: During the authentication phase, the SD_j generates random nonce R_2 and R_3, where $R_2 \in \{1, q-1\}$ and $R_3 \in \{0,1\}^k$. These nonces are utilized to construct authentication message $Auth_1 = \{B_2, B_4, C_1, C_2\}$, where $B_2 = A_1$. R_2, $A_1 = h_2$ $(FID_j || R_1)$, $B_4 = A_3 \oplus h_3$ (B_3), $C_1 = A_4 \oplus R_3$, and $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$. Similarly, ASP_i chooses random nonce$R_4 \in \{0,1\}^{\mathrm{k}}$, which is used in the derivation of authentication response message $Auth_2 = \{C_3, C_4\}$, where), $C_3 = R_4 \oplus A_4{}^{**}$, and $C_4 = h_1(FID_j{}^* || \phi_A)$. Consequently, messages $Auth_1{}^{\mathrm{Sub}}$ and $Auth_1{}^{\mathrm{Sub}}$ for the subsequent communication session will be different from those of the current session. This lack of correlation among authentication messages implies that Ψ is incapable of tracking F_j using any captured messages. □

Hypothesis 5: *Spoofing and forgery attacks are thwarted.*

Proof: Let us assume that attacker Ψ is attempting to forge message $Auth_1 = \{B_2, B_4, C_1, C_2\}$ sent from SD_j towards the ASP_i, as well as response message $Auth_2 = \{C_3, C_4\}$ forwarded back to the SD_j from ASP_i. Here, $B_2 = A_1$. R_2, $A_1 = h_2$ $(FID_j || R_1)$, $B_4 = A_3 \oplus h_3$ (B_3), $A_3 = h_1(FID_j || R_1 || MK_A)$, $C_1 = A_4 \oplus R_3$, $A_4 = A_4{}^* \oplus h_1(FID_j || FPW_j || P_B)$, $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$, $C_3 = R_4 \oplus A_4{}^{**}$, and $C_4 = h_1(FID_j{}^* || \phi_A)$. Clearly, this requires random nonces such as R_1, R_2, and R_3, farmer's real identity FID_j and password FPW_j, master key for ASP_iMK_A, and padding bits P_B, among other parameters. *Hypothesis 1* illustrates the difficulty of obtaining FID_j, *Hypothesis 3* demonstrates the difficulty of obtaining FPW_j, while *Hypothesis 4* shows the difficulty of obtaining random nonces. In addition, Ψ cannot obtain master key MK_A since it is randomly selected from $\{1, q-1\}$ by ASP_i. □

Hypothesis 6: *This scheme can withstand session hijacking attacks.*

Proof: Suppose that adversary Ψ has captured random nonces R_1, R_2, R_3, and R_4. Next, an attempt is made to compute session parameters $B_3 = A_2$. R_2, $C_1 = A_4 \oplus R_3$, $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$, $A_4{}^{**} = h_1(h_1(MK_A \oplus R_1{}^*) || FID_j{}^*)$, and $C_3 = R_4 \oplus A_4{}^{**}$ used in messages $Auth_1 = \{B_2, B_4, C_1, C_2\}$ and $Auth_2 = \{C_3, C_4\}$. Here, $B_2 = A_1$. R_2, $A_1 = h_2$ $(FID_j || R_1)$, $B_4 = A_3 \oplus h_3$ (B_3), $A_3 = h_1(FID_j || R_1 || MK_A)$, $C_1 = A_4 \oplus R_3$, $A_4 = A_4{}^* \oplus h_1(FID_j || FPW_j || P_B)$, $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$, $C_3 = R_4 \oplus A_4{}^{**}$, and $C_4 = h_1(FID_j{}^* || \phi_A)$. To hijack the session, other parameters are required apart from these random nonces, as illustrated in *Hypothesis 5*. Since these values are unavailable to Ψ, session hijacking is not possible. □

Hypothesis 7: *Impersonation attacks are prevented.*

Proof: Upon receiving message $Auth_1$, the ASP_i confirms whether parameter set {A_3^{**}, FID_j^*, R_1^*} is in its database. The aim is to abort the session if this verification fails. In addition, it derives value $C_2^* = h_1(B_2 || B_3^* || B_4 || C_1 || R_3^* || A_4^{**})$ and checks if $C_2^* \stackrel{?}{=} C_2$. Here, the authentication session is terminated if this verification flops. On its part, the SD_j computes parameters $R_4^* = C_3 \oplus A_4$, session key $\phi_S = h_1(R_3 || R_4^* || B_3 || A_4)$, and $C_4^* = h_1(FID_j || \phi_S)$ upon receiving message $Auth_2$. This is followed by the verification of whether $C_4^* \stackrel{?}{=} C_4$. Essentially, the authentication session is aborted if these two parameters are not the same. As such, the legitimacy of all the communicating entities is verified to thwart impersonations. □

Hypothesis 8: *Robust authentication is executed.*

Proof: At the SD_j side, noncesR_2 and $R3$ are generated and parameters $B_2 = A_1.\ R_2$, $B_3 = A_2.\ R_2$, $B_4 = A_3 \oplus h_3\ (B_3)$, $C_1 = A_4 \oplus R_3$, and $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$ are computed. These parameters are deployed to construct authentication message $Auth_1$ = {B_2, B_4, C_1, C_2} forwarded to the ASP_i. Similarly, the ASP_i generates random nonce R_4 utilized to derive values $C_3 = R_4 \oplus A_4^{**}$, session key $\phi_A = h_1(R_3^* || R_4 || B_3^* || A_4^{**})$, and $C_4 = h_1(FID_j^* || \phi_A)$. Lastly, authentication message $Auth_2$ = {C_3, C_4} is composed and forwarded to SD_j. During this process of authentication procedures, the legitimacy of SD_j is verified at the ASP_i using parameters {A_3^{**},FID_j^*, R_1^*}, C_2^*, and C_2, as demonstrated in Hypothesis 7. Similarly, the authenticity of ASP_i is verified at the SD_j using parameters $C_4^*$and C_4, as illustrated in Hypothesis 7. □

Hypothesis 9: *This protocol prevents de-synchronization and DoS attacks.*

Proof: Most of the authentication protocols incorporate timestamps in the exchanged messages, which renders them susceptible to de-synchronization and DoS attacks. The aim of these timestamps is to uphold the freshness of the transmitted messages. In the proposed scheme, random nonces are utilized to preserve the freshness of the exchanged messages. For instance, the SD_j generates random nonces R_2 and R_3 that are used to derive parameters $B_2 = A_1.\ R_2$, $B_3 = A_2.\ R_2$, $B_4 = A_3 \oplus h_3\ (B_3)$, $C_1 = A_4 \oplus R_3$, and $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$ of authentication message $Auth_1$ = {B_2, B_4, C_1, C_2} forwarded to the ASP_i. On its part, the ASP_i chooses random nonce R_4, which is incorporated in values $C_3 = R_4 \oplus A_4^{**}$, session key $\phi_A = h_1(R_3^* || R_4 || B_3^* || A_4^{**})$, and $C_4 = h_1(FID_j^* || \phi_A)$ of message $Auth_2$ = {C_3, C_4} forwarded to SD_j. □

Hypothesis 10: *This scheme eliminates the need for verifier tables.*

Proof: Some authentication schemes require that the communicating parties maintain verifier tables, which are queried during the authentication process. If the attackers gain access to these verifier tables, the entire network can be compromised and brought down. In the proposed scheme, the ASP_i authenticates the SD_j using parameter set {A_3^{**},FID_j^*, R_1^*}, and $C_2^*$and C_2. Whereas values A_3^{**},FID_j^* and R_1^* are re-computed and compared to the ones in its database, parameter C_2^* is re-computed and compared to the one received in authentication message $Auth_1$ = {B_2, B_4, C_1, C_2} received from the SD_j. On the other hand, the SD_j authenticates ASP_i using value $C_4^* = h_1(FID_j || \phi_S)$, which is re-calculated and compared with its equivalent C_4 received from ASP_i in authentication message $Auth_2$ = {C_3, C_4}. This eliminates the need for the ASP_i and SD_j to maintain verifier tables. □

Hypothesis 11: *Man-in-the-middle and replay attacks are thwarted.*

Proof: Suppose that the adversary is interested in computing and replaying bogus authentication parameters A_3^{**}, FID_j^*, R_1^*, C_2^*, C_3, and C_4 needed to successfully authenticate ASP_i and SD_j. Here, $A_3^{**} = B_4 \oplus h_3\ (B_3^*)$, $B_3 = A_2.\ R_2$, $B_3^* = MK_A.\ B_2$, $B_4 = A_3 \oplus h_3\ (B_3)$, $A_3 = A_3^* \oplus h_1(FPW_j\,||\,FID_j)$, $C_2^* = h_1(B_2\,||\,B_3^*\,||\,B_4\,||\,C_1\,||\,R_3^*\,||\,A_4^{**})$, $C_3 = R_4 \oplus A_4^{**}$, and $C_4 = h_1(FID_j^*\,||\,\phi_A)$. Based on *Hypothesis 1*, Ψ has no access to FID_j, while according to *Hypothesis 3*, Ψ has no access to FPW_j. Similarly, it has been shown in *Hypothesis 4* that Ψ does not have access to random nonces incorporated in these parameters. Based on *Hypothesis 5*, master key MK_A is never available to Ψ. Therefore, our scheme can withstand MitM attacks. □

Hypothesis 12: *The session key is set up for message encryption.*

Proof: Upon receiving message $Auth_1$ from SD_j, the ASP_i generates nonce R_4 and computes values $B_3^* = MK_A.\ B_2$, $A_3^{**} = B_4 \oplus h_3\ (B_3^*)$, $A_4^{**} = h_1(h_1(MK_A \oplus R_1^*)\,||\,FID_j^*)$, and $R_3^* = C_1 \oplus A_4^{**}$. These parameters are utilized to derive session key $\phi_A = h_1(R_3^*\,||\,R_4\,||\,B_3^*\,||\,A_4^{**})$. Similarly, after receiving message $Auth_2 = \{C_3, C_4\}$ from ASP_i, the SD_j computes value $R_4^* = C_3 \oplus A_4$ and session key $\phi_S = h_1(R_3\,||\,R_4^*\,||\,B_3\,||\,A_4)$. These keys are employed to encipher the exchanged messages. □

Hypothesis 13: *Known session-specific temporary information attacks are prevented.*

Proof: During the authentication phase, the ASP_i computes session key $\phi_A = h_1(R_3^*\,||\,R_4\,||\,B_3^*\,||\,A_4^{**}$, while the SD_j derives session key $\phi_S = h_1(R_3\,||\,R_4^*\,||\,B_3\,||\,A_4)$. Here, $R_3^* = C_1 \oplus A_4^{**}$, $C_1 = A_4 \oplus R_3$, $A_4^{**} = h_1(h_1(MK_A \oplus R_1^*)\,||\,FID_j^*)$, $B_3^* = MK_A.\ B_2$, $R_4^* = C_3 \oplus A_4$, $C_3 = R_4 \oplus A_4^{**}$, $A_4 = h_1(h_1(MK_A \oplus R_1)\,||\,FID_j)$, $B_3 = A_2.\ R_2$, and $A_2 = A_2^* - h_2(FID_j\,||\,FPW_j)$. It was demonstrated in *Hypothesis 11* that attacker Ψ has no access to FID_j, MK_A, FPW_j, and random nonces used in these session keys. In addition, the computation of parameters, such as $B_3^* = MK_A.\ B_2 = MK_A.\ A_1.\ R_2 = MK_A.\ h_2\ (FID_j\,||\,R_1).\ R_2 = A_2.\ R_2$, even when B_2 and A_2 are known, is difficult due to the intractability of the computational Diffie–Hellman (CDH) problem. □

Hypothesis 14: *Key secrecy is upheld.*

Proof: Suppose that attacker Ψ has access to private values such as random nonces R_2, R_3, and R_4. Let us also assume that authentication messages $Auth_1 = \{B_2, B_4, C_1, C_2\}$ and $Auth_2 = \{C_3, C_4\}$ have been captured by the adversary. Using these parameters, an attempt is made to derive messages $Auth_1^{Sub}$ and $Auth_1^{Sub}$ for the subsequent communication session. Here, $B_2 = A_1.\ R_2$, $A_1 = h_2\ (FID_j\,||\,R_1)$, $B_3 = A_2.\ R_2$, $A_2 = h_2\ (FID_j\,||\,R_1).MK_A$, $B_4 = A_3 \oplus h_3\ (B_3)$, $A_3 = h_1(FID_j\,||\,R_1\,||\,MK_A)$, $C_1 = A_4 \oplus R_3$, $A_4 = h_1(h_1(MK_A \oplus R_1)\,||\,FID_j)$, $C_2 = h_1(B_2\,||\,B_3\,||\,B_4\,||\,C_1\,||\,R_3\,||\,A_4)$,$\phi_A = h_1(R_3^*\,||\,R_4\,||\,B_3^*\,||\,A_4^{**})$, and $C_4 = h_1(FID_j^*\,||\,\phi_A)$. It is clear that even with the captured random nonces, the computation of these authentication messages will still fail. This is because Ψ still needs other parameters, such as FID_j and MK_A. According to *Hypothesis 1*, FID_j is unavailable to Ψ. Similarly, *Hypothesis 5* has shown the difficulty of obtaining master key MK_A. Moreover, *Hypothesis 13* has demonstrated the difficulty of deriving B_3 since it requires solving the CDH problem. □

Hypothesis 15: *Privileged insider and stolen smart device attacks are prevented.*

Proof: Let us assume that Ψ has stolen the farmer's smart device SD_j. Thereafter, the security tokens $\{A_1, A_2^*, A_3^*, A_4^*, B_1\}$ stored in its memory are extracted. This can also happen when Ψ has some privileged access to these parameters. Here, $A_1 = h_2\ (FID_j\,||\,R_1)$, $A_2^* = A_2 + h_2(FID_j\,||\,FPW_j)$, $A_3^* = A_3 \oplus h_1(FPW_j\,||\,FID_j)$, $A_4^* = A_4 \oplus h_1(FID_j\,||\,FPW_j\,||\,P_B)$, and $B_1 = h_1(A_2\,||\,A_3\,||\,A_4)$. The aim of the attacker is to access the secret value set $\{A_2, A_3, A_4\}$, where $A_2 = h_2\ (FID_j\,||\,R_1).\ MK_A$, $A_3 = h_1(FID_j\,||\,R_1\,||\,MK_A)$, and $A_4 = h_1(h_1(MK_A$

$\oplus R_1)||FID_j)$. However, all these parameters are encapsulated in other values such FID_j, FPW_j, and P_B; hence, their recovery is challenging. □

Hypothesis 16: *The proposed scheme is highly scalable and adaptable.*

Proof: In the proposed scheme, farmer F_j communicates directly to the service provider ASP_i devoid of any centralized entity. In addition, *Hypothesis 10* describes how the proposed scheme eliminates the need for verifier tables. As such, any farmer smart device SD_K can seamlessly join and leave the network without affecting the performance of the already existing devices. □

5. Performance Evaluation

In this section, three common metrics deployed in the performance evaluation of authentication protocols are used to gauge the proposed scheme. These metrics include computation and communication costs, as well as the supported security characteristics. The specific details about the evaluation procedures are described in the following sub sections.

5.1. Computation Costs

To determine the execution time for the various cryptographic operations, ASP_i is emulated in a multi-precision integer and rational arithmetic cryptographic library (MIRACL) in a server with the specifications in Table 2.

Table 2. Server specifications.

Feature	Description
Operating system	Ubuntu 22.04 LTS
RAM	8 GB
Processor	Intel Core i7-8565U
Operating system type	64-bit
Clock frequency	3.2 GHz

On the other hand, the farmer's SD_j is emulated using Raspberry Pi 3 Model B Rev 1.2, whose specifications are presented in Table 3.

Table 3. Smart device specifications.

Feature	Description
Operating system	Ubuntu 20.04 LTS
RAM	1 GB
Processor	Quad-core
Operating system type	64 bit
Clock frequency	1.4 GHz

Under these conditions, the average execution times for various cryptographic primitives are presented in Table 4.

During the authentication and key negotiation phase, the SD_j executes a single T_{MTP}, six T_H, a single T_{PS}, and two T_{SM} operations. On the other hand, the ASP_i carries out a single T_{SM} and six T_H operations. Table 5 presents the comparisons of the computation cost of the proposed scheme with other related protocols.

Based on the values in Table 5, the protocol in [18] has a computation cost of 31.847 ms, while the scheme in [6] has a computation overhead of 14.838 ms. Similarly, the computation costs for the protocols in [5,20,40] and the proposed scheme are 33.692 ms, 14.97 ms, 77.102 ms, and 12.662 ms, respectively. As shown in Figure 4, the protocol in [40] incurs the highest computation costs.

Table 4. Execution time for various cryptographic operations.

Cryptographic Operation	Time (ms) SD_j	Time (ms) ASP_i
Hashing operation (T_H)	0.314	0.056
Bilinear pairing (T_{BP})	33.051	4.715
Elliptic curve scalar multiplications (T_{SM})	2.256	0.654
Symmetric encryption/Decryption (T_{ED})	0.019	0.002
Elliptic curve point subtraction (T_{PS})	0.0115	0.003
Modular exponentiation (T_{ME})	0.325	0.083
Modular multiplication (T_{MM})	0.015	0.002
Modular addition (T_{MA})	0.012	0.001
Fuzzy extraction (T_{FE})	2.253	0.674
t-degree univariate polynomial evaluation (T_{PL})	13.3	0.3
Map-to-point hashing (T_{MTP})	5.264	2.853
Elliptic curve point addition (T_{PA})	0.017	0.004

Table 5. Computation costs comparisons.

Scheme	Derivations: User/Smart Device/Sensor	Derivations: Server/Gateway Node	Total (ms)
Vangala et al. [18]	$22\ T_H + 8\ T_{SM} + 2\ T_{PA} + T_{FE} = 27.243$	$12\ T_H + 6\ T_{SM} + 2\ T_{PA} = 4.604$	31.847
Rangwani et al. [6]	$8\ T_H + 5\ T_{SM} = 13.792$	$7\ T_H + T_{SM} = 1.046$	14.838
Bera et al. [5]	$7\ T_H + 6\ T_{SM} + 2\ T_{PA} + T_{PL} = 29.068$	$7\ T_H + 6\ T_{SM} + 2\ T_{PA} + T_{PL} = 4.624$	33.692
Vangala et al. [20]	$9\ T_H + 4\ T_{SM} = 11.85$	$9\ T_H + 4\ T_{SM} = 3.12$	14.970
Wu et al. [40]	$2\ T_{BP} + 2\ T_{ME} + 2\ T_{ED} + T_H = 67.446$	$2\ T_{BP} + 2\ T_{ME} + 2\ T_{ED} + T_H = 9.656$	77.102
Proposed	$T_{MTP} + 6\ T_H + 2\ T_{SM} + T_{PS} = 11.672$	$T_{SM} + 6\ T_H = 0.99$	12.662

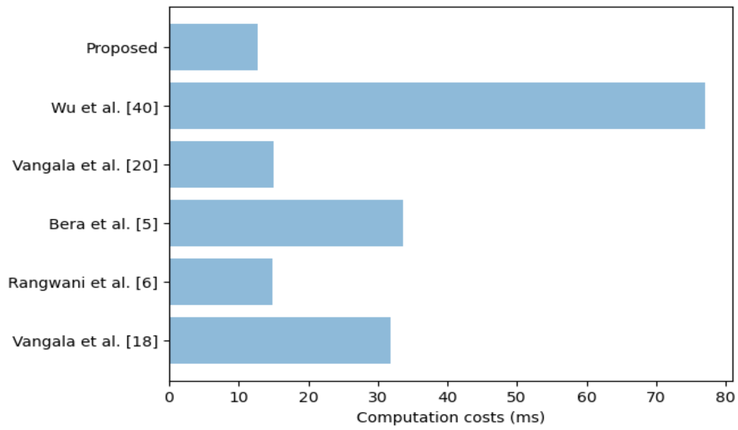

Figure 4. Computation costs comparisons [5,6,18,20,40].

This is attributed to the time-consuming bilinear pairing operations executed in this scheme. This is followed by the schemes in [5,6,18,20] and the proposed protocols in that order. The high computation overhead in [40] is attributed to the time-consuming bilinear pairing operations executed in this scheme. Since the farmer's smart device is battery-powered, our scheme is the most efficient and ensures that the battery for SD_j lasts longer. On the other hand, deploying the protocol in [40] in SD_j will drain its battery within a short time.

5.2. Communication Costs

To derive the number of bits used in the proposed protocol, the sizes of the messages exchanged between the SD_j and ASP_i during the authentication and key agreement phase are taken into consideration. For fair comparison, the values in [5] are used, in which the output sizes of the various cryptographic operations are presented in Table 6 below.

Table 6. Parametric sizes.

Operation	Size (bits)
Real identity	160
Random nonce	160
Hashing output	256
Points in finite group	512
Timestamp	32
Password	160

In our scheme, two messages are exchanged during the authentication and key negotiation phase. Whereas message $Auth_1 = \{B_2, B_4, C_1, C_2\}$ is sent from the SD_j towards the ASP_i, message $Auth_2 = \{C_3, C_4\}$ is transmitted from ASP_i towards SD_j. Here, $B_2 = A_1$. R_2, $B_4 = A_3 \oplus h_3\ (B_3)$, $C_1 = A_4 \oplus R_3$, $C_2 = h_1(B_2 || B_3 || B_4 || C_1 || R_3 || A_4)$, $C_3 = R_4 \oplus A_4^{**}$, and $C_4 = h_1(FID_j^{*} || \phi_A)$. Table 7 illustrates the derivation of the communication cost of this scheme.

Table 7. Message sizes.

Message	Size (bits)
$SD_j \rightarrow ASP_i$	
$Auth_1$:$\{B_2, B_4, C_1, C_2\}$	992
$B_4 = C_1 = C_2 = 160$; $B_2 = 512$	
$ASP_i \rightarrow SD_j$	
$Auth_2$: $\{C_3, C_4\}$	320
$C_3 = C_4 = 160$	
Total	1312

On the other hand, the protocol in [18] exchanges four messages, while the scheme in [6] requires five messages during the authentication process, as shown in Table 8. On their part, the schemes in [5,20,40] exchange 2 messages, 3 messages, and 10 messages, respectively. In terms of the total message sizes, the schemes in [5,6,18,20,40] require 5792 bits, 4128 bits, 2016 bits, 2305 bits, and 1600 bits, respectively.

Table 8. Communication costs comparisons.

Scheme	Number of Exchanged Messages	Size (bits)
Vangala et al. [18]	4	5792
Rangwani et al. [6]	5	4128
Bera et al. [5]	2	2016
Vangala et al. [20]	3	2305
Wu et al. [40]	10	1600
Proposed	2	1312

As shown in Figure 5, the scheme in [18] has the highest communication cost of 5792 bits, followed by the protocols in [5,6,20,40] and the proposed scheme, respectively.

Since the farmer's smart device is battery-powered, it has limited communication capability and hence the proposed protocol is the most efficient.

5.3. Security Characteristics

The goal of this section is to compare the security characteristics of the proposed scheme with other related protocols. Table 9 presents the results of this comparative evaluation.

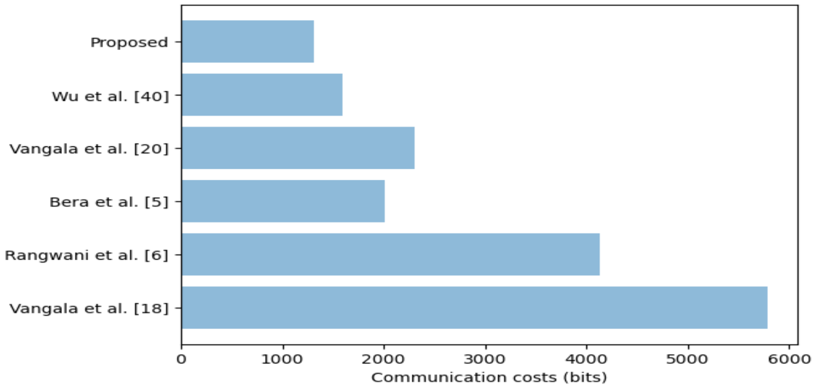

Figure 5. Communication costs comparisons [5,6,18,20,40].

Table 9. Security characteristics comparisons.

	[18]	[6]	[5]	[20]	[40]	Proposed
Security features						
User privacy	√	√	-	√	√	√
Anonymity	√	√	-	√	√	√
Unlinkability	-	-	-	-	-	√
Untraceability	√	√	-	√	√	√
Robust authentication	√	√	√	√	√	√
No verifier tables	×	√	√	×	-	√
Session key agreement	√	√	√	√	√	√
Key secrecy	√	√	√	-	-	√
Robust against:						
Side-channeling	√	√	√	√	×	√
Physical capture	√	√	√	√	×	√
Eavesdropping	×	×	√	×	×	√
Password guessing	√	√	×	×	√	√
Spoofing	×	×	×	×	×	√
Forgery	×	×	√	×	×	√
Replay	√	√	√	√	√	√
Session hijacking	×	×	×	×	×	√
Impersonation	√	√	√	√	×	√
De-synchronization	×	×	×	×	×	√
MitM	√	√	√	√	√	√
Privileged insider	√	√	√	√	√	√
KSSTI	×	√	√	√	×	√
DoS	√	√	-	√	√	√
Stolen smart device	√	√	√	√	×	√

√: supported; ×: not supported; -: not considered.

As shown in Table 9, the schemes in [5,20] each support 14 security characteristics, while the protocol in [18] offers support for 15 security features. On the other hand, the scheme in [6] supports 17 features, while the proposed protocol supports all 23 security features. Therefore, our scheme is the most secure and privacy-preserving.

Based on the results above, it is evident that the proposed scheme results in significant improvements in computation costs, communication costs, and supported security characteristics. Regarding computation overheads, the protocol in [6] with a cost of 14.838 m is used as the baseline. On the hand, the scheme in [40] with a communication cost of 1600 bits is used as the baseline. Similarly, the protocol in [18], which offers support for 15 security features, is deployed as the baseline. Using these baseline values, the proposed protocol results in 14.67% and 18% reductions in computation and communication costs, respectively, and a 35.29% improvement in supported security features.

6. Conclusions

In precision agriculture, numerous sensors such as radiation, air humidity, optimal, soil moisture, and ground sensors are deployed. In addition, intelligent precision agriculture utilizes numerous IoT devices and drones to monitor agricultural surroundings. Although these technologies help boost productivity in the face of limited resources, they are exposed to threats such as eavesdropping, message falsification, DoS, replay, MitM, and impersonations. Therefore, past researchers have seen the development of many security solutions for this environment. However, the attainment of perfect privacy and security at low computation and communication overheads still remains a mirage. The developed scheme has been shown to solve some of these challenges. For example, it has been shown to be resilient against side-channeling, physical capture, eavesdropping, password guessing, spoofing, forgery, replay, session hijacking, impersonation, de-synchronization, man-in-the-middle, privileged insider, denial of service, stolen smart device, and known session-specific temporary information attacks. Using the values in [6,18,40] as baselines, the proposed scheme leads to 14.67% and 18% reductions in computation and communication costs, respectively, and a further 35.29% improvement in supported security features. Future research will revolve around further enhancements of its performance as well as evaluation using metrics that were out of the scope of the current work.

Author Contributions: Z.A.A. and V.O.N.; methodology, and writing—original draft preparation, H.M.J.; software, data curation, validation, and writing—review and editing, J.M. and Z.A.A.; formal analysis, investigation, supervision, project administration, and funding acquisition, M.A.H., Z.A.H. and A.J.Y.A.; resources, visualization, and formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Natural Science Foundation of Top Talent of SZTU (Grant No. 20211061010016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. 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.* **2022**, *18*, 904–919. [CrossRef]
2. Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision Agriculture Techniques and Practices: From Considerations to Applications. *Sensors* **2019**, *19*, 3796. [CrossRef] [PubMed]
3. Shi, X.; An, X.; Zhao, Q.; Liu, H.; Xia, L.; Sun, X.; Guo, Y. State-of-the-Art Internet of Things in Protected Agriculture. *Sensors* **2019**, *19*, 1833. [CrossRef]
4. Vangala, A.; Das, A.K.; Chamola, V.; Korotaev, V.; Rodrigues, J.J. Security in IoT-enabled smart agriculture: Architecture, security solutions and challenges. *Cluster Comput.* **2022**, *26*, 879–902. [CrossRef]
5. Bera, B.; Vangala, A.; Das, A.K.; Lorenz, P.; Khan, M.K. Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment. *Comput. Stand. Interfaces* **2022**, *80*, 103567. [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. Lan, G.; Brewster, C.; Spek, J.; Smeenk, A.; Top, J. *Blockchain for Agriculture and Food*; Findings from the Pilot Study, Report; Wageningen Economic Research: Wageningen, The Netherlands, 2017; p. 34.
8. Nyangaresi, V.O.; Ibrahim, A.; Abduljabbar, Z.A.; Hussain, M.A.; Al Sibahee, M.A.; Hussien, Z.A.; Ghrabat, M.J.J. Provably Secure Session Key Agreement Protocol for Unmanned Aerial Vehicles Packet Exchanges. In Proceedings of the 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 9–10 December 2021; pp. 1–6.
9. Sontowski, S.; Gupta, M.; Chukkapalli, S.S.L.; Abdelsalam, M.; Mittal, S.; Joshi, A.; Sandhu, R. Cyber attacks on smart farming infrastructure. In Proceedings of the 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC), Atlanta, GA, USA, 1–3 December 2020; pp. 135–143.
10. Khanna, A.; Kaur, S. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Comput. Electron. Agric.* **2019**, *157*, 218–231. [CrossRef]

11. Van der Merwe, D.; Burchfield, D.R.; Witt, T.D.; Price, K.P.; Sharda, A. Drones in agriculture. *Adv. Agron.* **2020**, *162*, 1–30.
12. Dagar, R.; Som, S.; Khatri, S.K. Smart farming–IoT in agriculture. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 1052–1056.
13. Sanjeevi, P.; Prasanna, S.; Kumar, B.S.; Gunasekaran, G.; Alagiri, I.; Anand, R.V. Precision agriculture and farming using Internet of Things based on wireless sensor network. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3978. [CrossRef]
14. Nyangaresi, V.O.; Abduljabbar, Z.A.; Refish, S.H.A.; Al Sibahee, M.A.; Abood, E.W.; Lu, S. Anonymous Key Agreement and Mutual Authentication Protocol for Smart Grids. In *Cognitive Radio Oriented Wireless Networks and Wireless Internet, Proceedings of the 16th EAI International Conference, CROWNCOM 2021, Virtual Event, 11 December 2021, and 14th EAI International Conference, WiCON 2021, Virtual Event, 9 November 2021*; Springer International Publishing: Cham, Switzerland, 2022; pp. 325–340.
15. Wazid, M.; Das, A.K.; Bhat, V.; Vasilakos, A.V. LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. *J. Netw. Comput. Appl.* **2020**, *150*, 102496. [CrossRef]
16. Wang, D.; Li, W.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4081–4092. [CrossRef]
17. Challa, S.; Das, A.K.; Gope, P.; Kumar, N.; Wu, F.; Vasilakos, A.V. Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems. *Futur. Gener. Comput. Syst.* **2018**, *108*, 1267–1286. [CrossRef]
18. Vangala, A.; Das, A.K.; Lee, J. Provably secure signature-based anonymous user authentication protocol in an Internet of Things-enabled intelligent precision agricultural environment. *Concurr. Comput. Prac. Exp.* **2021**, *35*, e6187. [CrossRef]
19. Alsamhi, S.H.; Shvetsov, A.V.; Kumar, S.; Shvetsova, S.V.; Alhartomi, M.A.; Hawbani, A.; Rajput, N.S.; Srivastava, S.; Saif, A.; Nyangaresi, V.O. UAV Computing-Assisted Search and Rescue Mission Framework for Disaster and Harsh Environment Mitigation. *Drones* **2022**, *6*, 154. [CrossRef]
20. 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]
21. Akram, S.V.; Malik, P.K.; Singh, R.; Anita, G.; Tanwar, S. Adoption of blockchain technology in various realms: Opportunities and challenges. *Secur. Priv.* **2020**, *3*, e109. [CrossRef]
22. Lin, Y.-P.; Petway, J.R.; Anthony, J.; Mukhtar, H.; Liao, S.-W.; Chou, C.-F.; Ho, Y.-F. Blockchain: The Evolutionary Next Step for ICT E-Agriculture. *Environments* **2017**, *4*, 50. [CrossRef]
23. Almadhoun, R.; Kadadha, M.; Alhemeiri, M.; Alshehhi, M.; Salah, K. A user authentication scheme of IoT devices using blockchain-enabled fog nodes. In Proceedings of the 2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA), Aqaba, Jordan, 28 October–1 November 2018; pp. 1–8.
24. 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]
25. Al Sibahee, M.A.; Nyangaresi, V.O.; Ma, J.; Abduljabbar, Z.A. Stochastic Security Ephemeral Generation Protocol for 5G Enabled Internet of Things. In *IoT as a Service, Proceedings of the 7th EAI International Conference, IoTaaS 2021, Sydney, Australia, 13–14 December 2021*; Springer International Publishing: Cham, Switzerland, 2022; pp. 3–18.
26. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* **2014**, *20*, 96–112. [CrossRef]
27. Chang, C.-C.; Le, H.-D. A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2015**, *15*, 357–366. [CrossRef]
28. Das, A.K.; Kumari, S.; Odelu, V.; Li, X.; Wu, F.; Huang, X. Provably secure user authentication and key agreement scheme for wireless sensor networks. *Secur. Commun. Netw.* **2016**, *9*, 3670–3687. [CrossRef]
29. 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]
30. Tian, Y.; Yuan, J.; Song, H. Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones. *J. Inf. Secur. Appl.* **2019**, *48*, 102354. [CrossRef]
31. Chae, C.-J.; Cho, H.-J. Enhanced secure device authentication algorithm in P2P-based smart farm system. *Peer-to-Peer Netw. Appl.* **2018**, *11*, 1230–1239. [CrossRef]
32. Nyangaresi, V.O.; Abduljabbar, Z.A.; Mutlaq, K.A.-A.; Ma, J.; Honi, D.G.; Aldarwish, A.J.Y.; Abduljaleel, I.Q. Energy Efficient Dynamic Symmetric Key Based Protocol for Secure Traffic Exchanges in Smart Homes. *Appl. Sci.* **2022**, *12*, 12688. [CrossRef]
33. Wu, F.; Xu, L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. *Peer-to-Peer Netw. Appl.* **2015**, *10*, 16–30. [CrossRef]
34. Srinivas, J.; Mukhopadhyay, S.; Mishra, D. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. *Ad Hoc Netw.* **2017**, *54*, 147–169. [CrossRef]
35. Zeng, X.; Xu, G.; Zheng, X.; Xiang, Y.; Zhou, W. E-AUA: An Efficient Anonymous User Authentication Protocol for Mobile IoT. *IEEE Internet Things J.* **2018**, *6*, 1506–1519. [CrossRef]
36. Liu, C.-H.; Chung, Y.-F. Secure user authentication scheme for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2017**, *59*, 250–261. [CrossRef]
37. Nyangaresi, V.O.; Abduljabbar, Z.A.; Ma, J.; Al Sibahee, M.A. Verifiable Security and Privacy Provisioning Protocol for High Reliability in Smart Healthcare Communication Environment. In Proceedings of the 2022 4th Global Power, Energy and Communication Conference (GPECOM), Cappadocia, Turkey, 14–17 June 2022; pp. 569–574.

38. Challa, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan, M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* **2018**, *69*, 534–554. [CrossRef]
39. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.H.; Giri, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 102502. [CrossRef]
40. Wu, H.-T.; Tsai, C.-W. An intelligent agriculture network security system based on private blockchains. *J. Commun. Netw.* **2019**, *21*, 503–508. [CrossRef]
41. Abduljaleel, I.Q.; Abduljabbar, Z.A.; Al Sibahee, M.A.; Ghrabat, M.J.J.; Ma, J.; Nyangaresi, V.O. A Lightweight Hybrid Scheme for Hiding Text Messages in Colour Images Using LSB, Lah Transform and Chaotic Techniques. *J. Sens. Actuator Netw.* **2022**, *11*, 66. [CrossRef]
42. Tai, W.-L.; Chang, Y.-F.; Li, W.-H. An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *J. Inf. Secur. Appl.* **2017**, *34*, 133–141. [CrossRef]
43. 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. *Futur. Gener. Comput. Syst.* **2018**, *84*, 200–215. [CrossRef]
44. He, D.; Zhang, Y.; Wang, D.; Choo, K.-K.R. Secure and Efficient Two-Party Signing Protocol for the Identity-Based Signature Scheme in the IEEE P1363 Standard for Public Key Cryptography. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 1124–1132. [CrossRef]
45. Feng, Q.; He, D.; Liu, Z.; Wang, D.; Choo, K.K.R. Multi-party signing protocol for the identity-based signature scheme in IEEE P1363 standard. *IET Inf. Secur.* **2020**, *1*, 1–10.
46. Nyangaresi, V.O. Terminal independent security token derivation scheme for ultra-dense IoT networks. *Array* **2022**, *15*, 100210. [CrossRef]
47. Sadhukhan, D.; Ray, S.; Biswas, G.P.; Khan, M.K.; Dasgupta, M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. *J. Supercomput.* **2020**, *77*, 1114–1151. [CrossRef]
48. Dhillon, P.K.; Kalra, S. A lightweight biometrics based remote user authentication scheme for IoT services. *J. Inf. Secur. Appl.* **2017**, *34*, 255–270. [CrossRef]
49. Chang, C.-C.; Nguyen, N.-T. An Untraceable Biometric-Based Multi-server Authenticated Key Agreement Protocol with Revocation. *Wirel. Pers. Commun.* **2016**, *90*, 1695–1715. [CrossRef]
50. Amin, R.; Islam, S.H.; Kumar, N.; Choo, K.-K.R. An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks. *J. Netw. Comput. Appl.* **2018**, *104*, 133–144. [CrossRef]
51. Li, X.; Niu, J.; Alam Bhuiyan, Z.; Wu, F.; Karuppiah, M.; Kumari, S. A Robust ECC-Based Provable Secure Authentication Protocol With Privacy Preserving for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3599–3609. [CrossRef]
52. Alotaibi, M. An Enhanced Symmetric Cryptosystem and Biometric-Based Anonymous User Authentication and Session Key Establishment Scheme for WSN. *IEEE Access* **2018**, *6*, 70072–70087. [CrossRef]
53. Moghadam, M.F.; Nikooghadam, M.; Al Jabban, M.A.B.; Alishahi, M.; Mortazavi, L.; Mohajerzadeh, A. An Efficient Authentication and Key Agreement Scheme Based on ECDH for Wireless Sensor Network. *IEEE Access* **2020**, *8*, 73182–73192. [CrossRef]
54. Fadi, A.T.; Deebak, B.D. Seamless authentication: For IoT-big data technologies in smart industrial application systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2919–2927.