agriculture

MDPI

Article

Robust Multi-Gateway Authentication Scheme for Agriculture Wireless Sensor Network in Society 5.0 Smart Communities

Haqi Khalid [1,*], Shaiful Jahari Hashim [1], Sharifah Mumtazah Syed Ahmad [1], Fazirulhisyam Hashim [1] and Muhammad Akmal Chaudhary [2]

[1] Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia; sjh@upm.edu.my (S.J.H.); s_mumtazah@upm.edu.my (S.M.S.A.); fazirul@upm.edu.my (F.H.)

[2] Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, Ajman City 346, United Arab Emirates; m.akmal@ajman.ac.ae

* Correspondence: haqikhalid1@gmail.com

Abstract: Recent Society 5.0 efforts by the Government of Japan are aimed at establishing a sustainable human-centered society by combining new technologies such as sensor networks, edge computing, Internet of Things (IoT) ecosystems, artificial intelligence (AI), big data, and robotics. Many research works have been carried out with an increasing emphasis on the fundamentals of wireless sensor networks (WSN) for different applications; namely precision agriculture, environment, medical care, security, and surveillance. In the same vein, almost all of the known authentication techniques rely on the single gateway node, which is unsuitable for the current sensor nodes that are broadly distributed in the real world. Despite technological advances, resource constraints and vulnerability to an attacker physically capturing some sensor nodes have remained an important and challenging research field for developing wireless sensor network user authentication. This work proposes a new authentication scheme for agriculture professionals based on a multi-gateway communication model using a fuzzy extractor algorithm to support the Society 5.0 environment. The scheme provides a secure mutual authentication using the well-established formal method called BAN logic. The formal security verification of the proposed scheme is validated with the AVISPA tool, a powerful validation method for network security applications. In addition, the security of the scheme was informally analyzed to demonstrate that the scheme is secure from different attacks, e.g., sensor capture, replay, and other network and physical attacks. Furthermore, the communication and computation costs of the proposed scheme are evaluated and show better performance than the existing authentication schemes.

Keywords: agriculture; Society 5.0; wireless sensor network; agriculture sensors; IoT; multi-gateway

Citation: 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. https://doi.org/10.3390/agriculture11101020

Academic Editor: Raul Morais

Received: 15 July 2021
Accepted: 1 September 2021
Published: 19 October 2021

1. Introduction

Society 5.0 has been launched by Japan for the perfect industrial structure and social system of the future. According to the Japan Cabinet Office (CAO), society 5.0 is "a human centred society that balances the economic development of a system by combining cyberspace and physical space to solve social problems" [1]. Figure 1 illustrates the evolution of societies from Society 1.0 to the new Society 5.0, in which everyone can live a safe and fulfilling life. Smart Food value chain of Society 5.0 with the National Organization for Agriculture and Food Research (NARO) addresses breeding, cultivation, harvesting, storage, processing, distribution, and consumption issues [2,3]. As a result, the process of establishing a "data-driven society", has begun, which now includes agriculture [4,5]. The ongoing evolution of information and communications technology (ICT) and digital technology of all kinds are the motivation behind Society 5.0 to offer individuals an enormous society of prospects for creativity, growth, unparalleled prosperity collaboration, and human to human, human to machine, and machine to device services [4,6].

Figure 1. New Society 5.0 [1].

By the end of 2025, the world market in smart agriculture is predicted to reach USD 15.3 billion compared to USD 5 billion in 2016, which is more than triple of the market size in just about ten years [7]. Smart agriculture in the agri-product exporting countries will become a critical IoT field [8]. At present, smart agriculture has been applied in IoT applications such as irrigation sensors [9], frost prediction of the event [10], farming of precision soil [11], identification of blind entity [12], smart farming [13], precision agriculture [14], so on. Terrestrial wireless sensor networks (TWSN) and wireless underground sensor networks (WUSN) are the two types of WSNs being utilized in agricultural fields. Wireless underground sensor networks [15] are planted inside the soil with higher frequencies being substantially reduced, while lower frequencies are allowed to permeate the soil [13,16–18]. The overview and the architecture of WSN in the agriculture environment is illustrated in Figure 2.

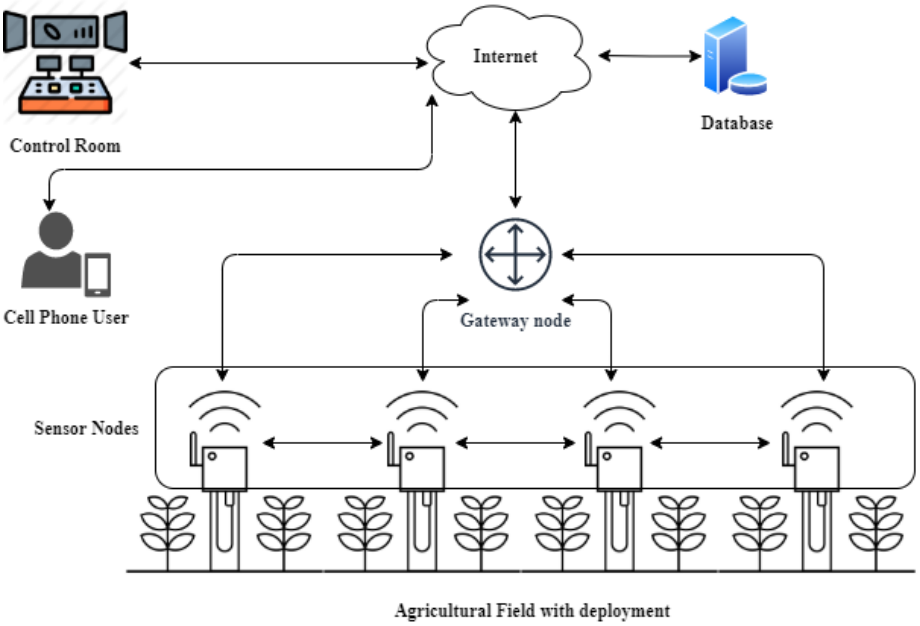

Figure 2. An overview of agriculture WSN environment.

The agriculture applications may transfer or monitor sensitive data via a public channel; thus securing the transmission and authenticating of highly sensitive information. Several gateways should also be included in dealing with a distributed environment to

avoid high computation costs in the entire network [19]. Additionally, different issues exist in IoT-based agriculture development, such as information security, privacy, data analysis, maintenance, mobility, and hardware [8,20,21]. The type of wireless communication (e.g., 4G, 5G, WiFi, 6LowPan, LoRa) used for connecting sensors distributed across a large area in the agriculture field may present a mobility challenge [22,23]. Implementing the IoT into the agriculture fields may allow the attacker to attack the agriculture systems; thus, the smart agriculture communication system needs to be secured [8]. Security concerns, such as eavesdropping, disruption, physical attack, and others, might compromise the data and structure of the network [24]. To address this, a data security architecture is constructed that protects data from sensors, wireless networks, and data processing applications through encryption, digital envelopes, digital signatures, and critical public key infrastructures (PKI) [25]. Generally, once the sensors and the gateway nodes are placed, they are stationary. In wireless environment, the cost of sending and receiving messages increases when the distance between the participants and the whole network increases. It is better to allow only the gateway nodes to communicate with the relatively far away users. However, a data flow with high speed may collide, and the performance of the WSN will be slowed down where there is only one gateway. More gateway nodes are needed when the sensors are distributed on a large scale. Thus, the costs of transmitting and receiving messages are much higher than the local computations at an entity in the network [26].

There have been many proposed authentication and key agreement (also known as the key establishment) schemes for WSNs in the literature. For instance, in [27] a lightweight authentication scheme (LAS) for IoT WSN users in a multi-gateway conception is proposed. Similarly, in [28] a three-factor mutual authentication protocol for multi-gateway IoT environments to solve the existing security weaknesses in two-factor authentication protocols is proposed. In 2014, a WSN with a lightweight authentication protocol was integrated with a fingerprint-based biological factor [29]. In [30], a new mechanism for user authentication and key agreement in heterogeneous ad hoc WSNs is proposed. In 2015, an authentication approach based on pseudo-identity temporal credentials in WSNs was devised [31]. In [32], a biometric-based user authentication solution for WSNs is suggested. In 2015, a new secure and more efficient authentication and key agreement scheme for agriculture monitoring using WSNs is proposed [33]. The work of [34] applied dark web technology to ensure the privacy of blockchain and servers. In 2017, work of [35] presented a confidentiality-preserving remote user authentication system for IoT users using WSN, which was more efficient than earlier comparable methods and could withstand all forms of security outbreaks. In [36], an authentication-based, smart-card, and password-based strategy for intelligent agriculture based on the use of fuzzy biometric extraction before providing users with required fields is developed. In [37], an elliptic curve-based user authentication mechanism based on symmetric cryptography (ECC) is presented. In [38], biometric-based authentication and key management services are discussed. In 2020, an Elliptic-Curve Diffie–Hellman authentication and key agreement approach for wireless sensor network (WSN) applications is suggested [39]. In [40], a new user authentication system based on signatures and the ECC in the IoT-enabled environment is presented. In [41], the WSN data protection, three-factor remote user authentication solution for increased security and efficient agricultural monitoring by ECC algorithm are also presented.

Table 1, shows that Turkanovic et al. in [30] did not secure forward privacy [42]. Amin and Biswas [43] found that the scheme in [33] is required to concentrate on redundancy. Wu et al. [31] highlighted specific security weaknesses such as sensor capture attacks, and impersonation attacks in He et al. [32]. On the other hand, Khalid et al. [33] revealed that Wu et al.'s [35] system lacks appropriate online registration and password change phases for sensor nodes. Ali et al. [36] and Lee et al. [28] are found to be vulnerable to impersonations, robbed smart-cards, ephemeral secret leaking (ESL), privileged insider attacks, and a Denial-of-Service (DoS) attack. Sadukhan et al. [37] does not lacks anonymity, traceability, and dynamic node addition. Furthermore, Yuan et al.'s [29] was found to be vulnerable to offline password guessing, privileged insider attacks, and gate-

way node impersonation attacks. In addition, it cannot provide query response protection. Moghadam et al. [39] and Haseeb et al. [44] are vulnerable to an insider attack, session key attack, and do not provide confidentiality. Vangala et al. [45] and Rangwani et al. [41] were subsequently broken by Ali et al. [36], who pointed out that the schemes are vulnerable to offline password guessing attack, identity guessing attack, and user tracking attack. Nevertheless, the analysis that identifies several attacks in the current authentication scheme has not been considered by the researchers (e.g., offline password/identity guessing attacks, sensor capture attacks, and impersonation attacks). In addition, the methods are vulnerable to inefficient authentication phases. In agriculture, wireless communication messages are transferred and received substantially at higher network entity computations than the local network entity computations. Therefore, transmission and reception expenses increase as network unit distance increases. As a result, the GWNs suffer high communication overhead, leading to slow shutdown or crash due to many users and sensors management in large-scale WSN.

Table 1. Comparison of the existing WSN authentication scheme in the agricultural field.

Ref.	Communication Model	Method	Tool	Advantages	Limitations
[29]	Single gateway	RSA public key	GNY logic	Mutual authentication	Vulnerability to offline password guessing, insider, and gateway node impersonation attack.
[30]	Single gateway	ECC	Prototype (MICA2 sensor node)	Vulnerability to offline password guessing, insider, and gateway node impersonation attack.	Vulnerability to various impersonation attacks.
[31]	Single gateway	Hash function, and XOR	PBC library	Reduces the computation burden.	Node captured impersonation attack.
[32]	Single gateway	AES	PBC library	Denial-of-service attack and sensor node impersonation attack.	Fail to provide acclaimed security goals.
[33]	Single gateway	Dynamic pseudonym identity.	C/C++	Most suitable for agriculture monitoring.	Needs to concentrate on redundancy.
[34]	Single gateway	Blockchain	PBC library	Ensures data privacy and integrity.	Single point failure.
[35]	Single gateway	ECC	JPBC library	Provides mutual authentication between the user, the sensor, and the gateway.	Lack of user anonymity or smart card stolen attack.
[36]	Single gateway	PKI	Crypto ++	Agriculture field monitoring.	Packets drop and latency ratio.
[37]	Single gateway	ECC	PBC library	Node transfers data to the user without any interference from the gateway.	Vulnerable to user impersonation, stolen smart card, privileged insider attacks and does not support anonymity, and traceability.
[38]	Single gateway	ECC	Crypto ++	Computationally less expensive.	An adversary could gain unauthorized access to the device.
[39]	Single gateway	ECDH	JPBC library	Secure from numerous security attacks.	Vulnerability to attacks.
[44]	Single gateway	ECC	PBC library	Achieves essential security requirements like integrity, anonymity, forward secrecy.	Vulnerability to offline guessing attack, and sensor capture attack.
[41]	Single gateway	Fuzzy extractor.	Crypto ++	Secure monitoring.	High computation cost.

Several alternative architectures with single-gateway architecture for agriculture environment are proposed previously. These single-gateway systems have low fault tolerance, as the gateway acts as a single point of failure, thus making it vulnerable to external attacks. Therefore, an efficient multi-gateway authentication scheme for agriculture is needed to address these issues [7], because insecure communication between the smart devices, gateways, and users makes the IoT agriculture environment vulnerable to various potential attacks. Several Internet of Things (IoT) smart devices, e.g., sensor nodes, can be deployed to monitor the agricultural environment in smart farming. The drones can be further utilized to collect the data sensed by the IoT smart devices, and even sometimes, they can directly collect the information from the specific agriculture fields. However, inse-

cure communication between the sensor nodes, gateways, and agriculture professionals makes the IoT agriculture environment vulnerable to various potential attacks, including replay, impersonation, man-in-the-middle, privileged-insider, and physical smart devices and drones capture attacks [46]. Apart from these, anonymity and mutual authentication properties to be highly maintained is essentially required. An adversary cannot trace the entities sending the data securely to the control room via a gateway. Therefore, to address the above issues, we propose a multi-gateway authentication scheme with the three factors being the identity, password, and personal biometrics for agriculture WSN. The proposed scheme relies mainly on the fuzzy extractor method. We have also provided the simulation of our scheme using AVISPA, a powerful validation tool for network security applications, and showed that our scheme is safe against popularly known attacks. Similarly, the BAN logic is utilized to prove the secure mutual authentication between entities.

2. Materials and Methods

2.1. Security Requirements

The integration of WSN in low power agriculture for the internet and society 5.0 requires adequate security mechanisms, which can offer essential safety safeguards for WSN applications, equipment, and communications in agriculture. The conventional Internet connections require sufficient security to use end-to-end communications between low-power farm WSN sensing devices and other external or internet companies. According to recent studies in [2,3,5,16,21,40,47,48] about the security of smart agriculture, the agriculture WSN authentication scheme must satisfy the security and functionality of agriculture WSN in Society 5.0. These security and functionality requirements are as following:

- Mutual authentication: the agriculture professional Ui and the sensor node Sn should authenticate each other with the help of the gateway node (GWN).
- Anonymity: an adversary should not get the real identity of the agriculture professional U_i.
- Multi-gateway: agriculture WSN has many sensor nodes and IoT devices that are distributed over large agriculture fields. Hence, a single-gateway node can hardly manage this number of nodes, causing a single point of failure. Therefore, the agriculture environment should support multi-gateway communication.
- Physical Attack: The attacker disturbs the protocol by causing a collision packet, inserting and interrogating packets to obtain information about the communication template, or delaying communication. Thus, the WSN should withstand physical attacks, such as sensor capture attacks and gateway attacks.
- Network Attacks: The WSN authentication scheme for agriculture must resist several attacks, such as an offline password guess attack, the user impersonation attack, the node impersonation attack, the modification attack, the man-in-the-middle attack, and the replay attack.

2.2. Single-Gateway Model

Many researchers have utilized the single communication model to design a user authentication for WSN. The model, as shown in Figure 3, includes user, gateway, and sensor nodes. In the model, the user can access the desired sensor node after registering himself/herself into the GWN. However, the model user can only access the sensor nodes that are deployed within the local network. Furthermore, a user cannot access any sensor nodes that are deployed in the different agriculture fields, especially in large-scale environments. The user first sends an authentication message to the gateway; the gateway then sends the message to the deployed sensor. Later, the sensor node sends back the message to the gateway, and it forwards the message to the user whether the user was granted access or not.

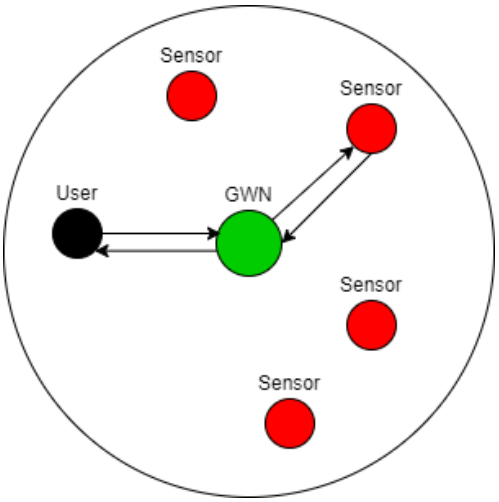

Figure 3. Single-gateway communication model.

2.3. Multi-Gateway Model

Amin and Biswas [43], and H. Guo [49] proposed a multi-gateway communication model, including users, gateway nodes (GWNs), and sensor nodes. Here, we divide gateway nodes into two categories: home gateway nodes (HGWN) and foreign gateway nodes (FGWN) according to the distance to other nodes—relatively close gateway nodes are called HGWNs, and the rest of them are called FGWNs. Sensors and gateway nodes are stationary after they are placed. The computing power of the gateway nodes is powerful, while sensors have low memory, low bandwidth, low battery power and limited computing power. Sensor nodes monitor and collect data, then send the sensed data to the nearest gateway node, i.e., HGWN. The HGWN forwards the received data to other FGWNs, users or sensors. For example, when a user wants to communicate with a sensor node, they need to authenticate each other.

As shown in Figure 4, if the user and the sensor belong to a home network managed by the same HGWN, the authentication process is as follows: Case 1; Firstly, the user sends a login message to HGWN. Second, HGWN authenticates the user and sends a message to the sensor node. Then, the sensor authenticates HGWN and returns messages to HGWN. After HGWN completes the authentication, it returns messages to the user. Finally, the user completes the authentication of HGWN and computes a session key with the sensor and HGWN. When a user wants to communicate with a sensor node in different networks, the detailed steps are shown in Figure 3. We describe the process as follows: Case 2; The user Ui sends a login message to its HGWN. The HGWN then broadcasts request messages to the sensor node that the user wants to request for a communication. After FGWN receives the broadcast messages, it checks whether the sensor node is in its database. If so, FGWN sends a message to HGWN. The HGWN returns reply messages to the user. Finally, the user and FGWN perform mutual authentication and negotiate the session key as shown in Figure 4.

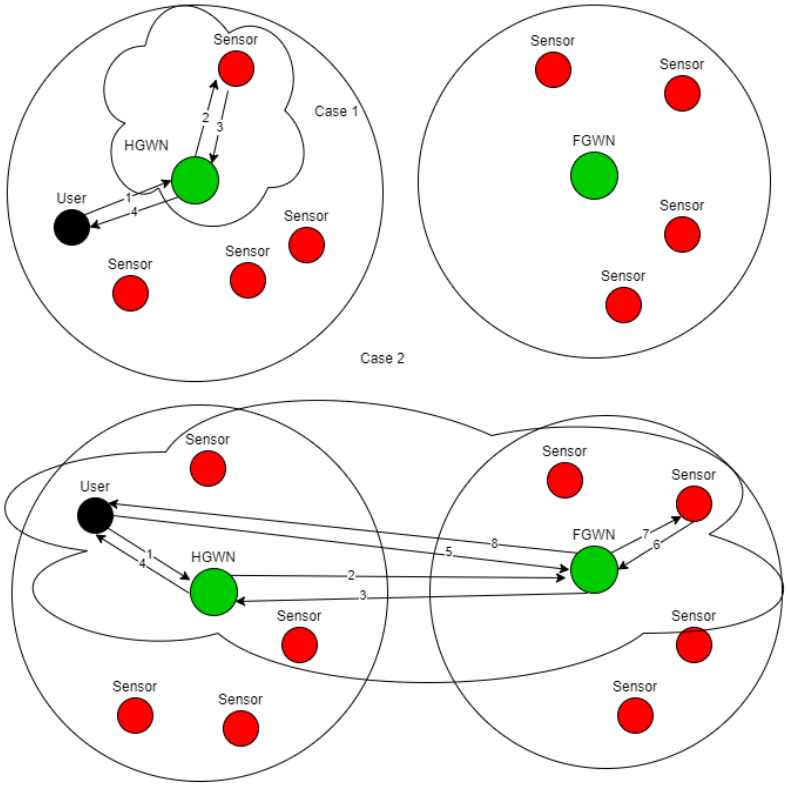

Figure 4. Multi-gateway communication model.

2.4. Fuzzy Extractor

This section provides a brief explanation of fuzzy extractors to clarify the procedure of the algorithm. In a Fuzzy extractor, there are two main procedures: a reproduction procedure referred to as (Rep), and a generation procedure referred to as (Gen). The two procedures are described as follows:

- *Gen*: the input of this procedure is the user biometric BIO_i. Furthermore, the outputs are the key to the biometric σ_i and the public parameter. Thus, the procedure function can be represented as $Gen(BIO_i) = (\sigma_i, \tau_i)$ where τ_i is the error tolerance threshold.
- *Rep*: This procedure retrieves the biometric key σ_i form corresponding auxiliary string τ_i and the user biometric $BIO_i^{'}$, where the function can be represented as $Rep(BIO_i^{'}, \tau_i) = \sigma_i$. This provides the error tolerance threshold τ greater than the Hamming distance between the original input of BIO_i and the retrieved biometric $BIO_i^{'}$.

However, the polynomial-time running of the Gen and Rep procedures is efficiently robust to the fuzzy extractor algorithm. Furthermore, recovering σ_i from the input of the biometric $BIO_i^{'}$ alongside the string of the auxiliary τ_i by an attacker is difficult. Thus, the fuzzy extractor algorithm is highly secured.

2.5. Proposed Scheme

The following section proposes a new multi-gateway authentication scheme for agriculture wireless sensor networks, as shown in Figure 5. The proposed scheme uses the smart card, password, and personal biometrics as authentication factors. There are four phases involved in the proposed scheme (e.g., pre-deployment phase, agriculture/sensor registration phase, login phase, and authentication phase). In Table 2, the used notations in the proposed scheme are illustrated. As mentioned in the literature review, a unique property of biometrics enables its use in authentication protocols. Using biometric keys with low-entropy passwords makes it difficult to fake or exchange, including the inability to be lost or forgotten. As a result, guessing biometric keys becomes a complex problem. This

study makes use of a robust fuzzy extractor. Finally, the WSN's sensor nodes, the GWNs, and the users are synchronized and use the timestamp to withstand the replay attack.

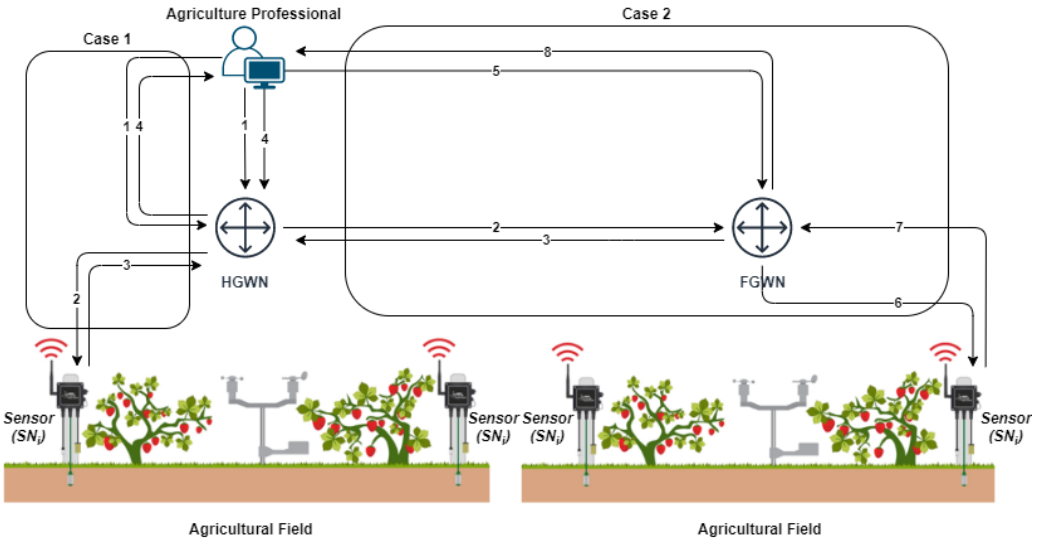

Figure 5. The proposed scheme communication models.

Table 2. Notations.

Notations	Description
SA	System administrator.
HGWN	Home gateway.
FGWN	Foreign gateway.
SNi	Sensor.
U_i	Agriculture professional.
SN_{ID}	Sensor identity.
SN_{MSK}	Sensor master key.
GW_{ID}	Gateway identity.
GW_{MSK}	Gateway master key.
HGW_{ID}	Home gateway identity.
HGW_{MSK}	Home gateway master key.
FGW_{ID}	Foreign gateway identity.
FGW_{MSK}	Foreign gateway master key.
U_{PW}	Agriculture profession Password.
U_{BIO}	Agriculture profession biometric.
PID_i	Pseudo-identity.
PWR_i	Pseudo-password.
SN_{ID}^n	New sensor identity.
SK_i	Secret key.
$h(.)$	One-way hash function.
(Gen)	Generation procedure of fuzzy extractor.
(Rep)	Reproduction procedure of fuzzy extractor.
$A \parallel B$	Concatenation.
$A \oplus B$	Exclusive-OR.

2.5.1. Pre-Deployment Phase

The system parameters are selected in this phase, and it pre-loads information in deployed sensor nodes and gateways before being deployed in a target field. This phase is carried out in a stand-alone mode. The system administrator (SA) is responsible for and manages the pre-deployment phase. Each cluster has n sensor nodes that are deployed randomly or manually in the preceding stage with a target field; each cluster also contains (HGWN). In this work, we assume that every sensor node chooses the nearest HGWN. The SA, on the other hand, selects the system parameters in the following manner:

1. Sensor node pre-deployment:
   - The SA randomly chooses a unique identity SN_{ID} and master key SN_{MSK}. For each deployed sensor node in the cluster $(1 \leq j \leq m)$, then, SA calculates $A_j = h(SN_{ID} \parallel SN_{MSK})$ for each sensor node. It also generates a distinct master key SN_{MSK}, with all the generated A_j, which are distinct throughout the WSN. Now, the credentials (SN_{ID}, A_j) are pre-loaded into the sensor node memory within its corresponding cluster priorly.
2. Gateway Pre-deployment:
   - First, the gateway selects an identity GW_{ID}, and GW_{MSK} as gateway master key for the deployed GWNs in the cluster. In the proposed scheme, there are two different GWNs: HGWNs, those located in a specific cluster, and those located outside a cluster called FGWN. The SA then generates an identity HGW_{ID} and HGW_{MSK} as gateway master key. The same goes for the FGWN generating FGW_{ID} and FGW_{MSK}.
   - Later, the SA computes $A_{HGWN} = h(HGW_{ID} \parallel SN_{ID} \parallel HGW_{MSK}) \oplus h(SN_{ID} \parallel SN_{MSK})$ for all n sensor nodes SN_i within HGWN, for example. The SA finally pre-loads the information $HGW_{ID}, (SN_{ID}, A_{HGWN}) \leq j \leq m, HGW_{MSK}$ into the memory of the HGWN priorly to its deployment in the target field.

2.5.2. Registration Phase

After the pre-deployment phase of the sensor nodes in the targeted agriculture field, the sensors are transmitted to the registered legal professional via HGWN and FGWN. The sensors and agriculture professionals must be registered with SA to access the desired services. The following sections outline how to register a sensor node and an agriculture professional:

- **User/agriculture professional registration:** Before participating in any communication during this phase, the user or agriculture professional must register with one of the GWNs. Assuming that the user chooses to register with HGWN, he or she must follow the steps outlined in Figure 6:
  - Agriculture Professional chooses U_{ID} as an identity and U_{PW}, which is the password, and a random number R to computes $PID_i = h(U_{ID} \parallel R)$ and $PWR_i = h(U_{PW} \parallel R)$. Then, the parameters PID_i, PWR_i are securely transmitted to the SA as a registration request.
  - The SA receives the message and generates an identity TID_i, which is 160-bit to compute $A_{th} = h(HGW_{ID} \parallel PID_i \parallel HGW_{MSK}) \oplus h(PID_i \parallel PWR_i \parallel TID_i)$ for each user U_i in HGWN. It also computes $A_{tf} = h(FGW_{ID} \parallel PID_i \parallel FGW_{MSK}) \oplus h(PID_i \parallel PWR_i \parallel TID_i)$ for the FGWN. The SA then issues an embedded smart-card $SC_i : (HGW_{ID}, A_{th}), (FGW_{ID}, A_{tf}), TID_i, Gen(.), Rep(.), h(.), t$, where t is the error tolerance threshold. Finally, it sends the message to the user Ui via a secure channel.
  - Now that the user U_i receives the embedded smart-card from SA securely, the U_i imprints their fingerprint U_{BIO} at the sensor of a specific terminal and computes $Gen(U_{BIO}) = (\sigma, \tau)$, which σ is the key of the biometric data and σ is the parameter. Then, the U_i computes $T_i = h(U_{ID} \parallel \sigma) \oplus R, S_i = h(PID_i \parallel PWR_i \parallel \sigma)$ and $A^*_{th} = A_{th} \oplus h(\sigma \parallel R) = (HGW_{ID} \parallel PID_i \parallel HGW_{MSK}) \oplus h(PID_i \parallel PWR_i \parallel TID_i) \oplus h(\sigma \parallel R)$. The U_i then computes $A^*_{if} = A_{if} \oplus h(\sigma \parallel R) = h(FGW_{ID} \parallel PID_i \parallel FGW_{MSK}) \oplus h(PID_i \parallel PWR_i \parallel TID_i) \oplus h(\sigma \parallel R)$. U_i stores τ, T_i and S_i in the smart-card SC. The U_i then replaces $A_{th} with A^*_{th}$, and A_{tf} with A^*_{if} in the stored information of SC. The stored data will be as $(HGW_{ID}, A^*_{th}), (FGW_{ID}, A^*_{if}), TID_i, Gen(.), Rep(.), h(.), t, \tau, T_i, S_i$.

However, the pair (PID_i, TID_i) are stored in the database of the corresponding HGWNs to the Ui and also sores them into all FGWNs by the SA if the user desires to access services from any sensor node through the FGWNs.

- **Newly Joined Sensors:** The newly joined sensor node must be registered with the SA for further communication services in this phase. The phase is performed after being deployed priorly in the pre-deployment phase. Figure 7 shows the steps of newly joined sensors. As we mentioned above, each sensor in the cluster has the information (SN_{ID}, A_j) in its memory. Thus, to register the sensor node SNi into the SA, the sensor is required to apply the following steps:
  - Firstly, the sensor SN_i chooses an identity (SN^n_{ID}), and a random number r_{sn} is generated for each sensor to compute $N_{SN} = h(SN^n_{ID} \parallel r_{sn})$, and $M_{SN} = h(N_{SN} \parallel r_{sn})$. Then, the sensor sends N_{SN} to the SA securely.
  - Now, the SA receives the message and obtain a new sensor identity SN^n_{ID} and generate a master key SN^n_{MSK} for the newly joined sensor. Then, it calculates $A^n_j = h(SN^n_{ID} \parallel SN^n_{MSK})$ and loads the (A^n_j, SN^n_{ID}) into the sensor memory within its corresponding cluster.

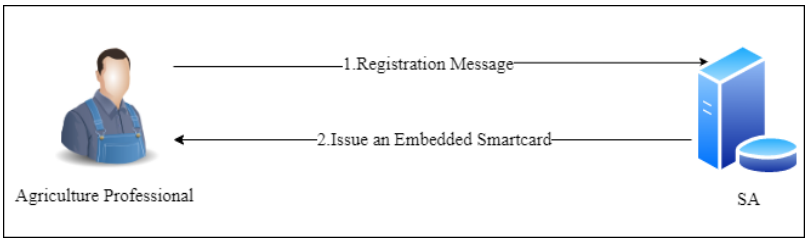

Figure 6. Agriculture professional registration phase.

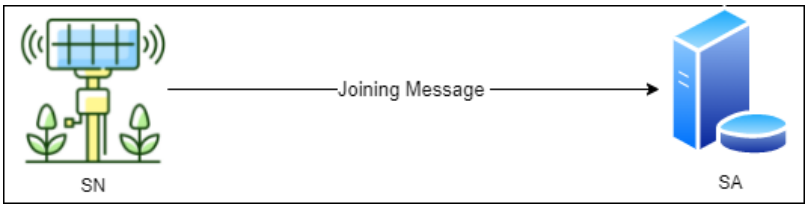

Figure 7. Newly joined sensors phase.

2.5.3. Login Phase

This phase enables the agriculture professional to authenticate to HGWNs using the smart-card SC. After inserting the smart-card into a specific card reader terminal, the SC transmits the login request message to the HGWNs by performing the following steps, which are shown in Figure 8:

- Firstly, the agriculture user inserts their smart-card and inputs the username U_{ID}, password U_{PW} and imprints their biometric U_{BIO} at the sensor. Then, the smart-card calculates using the error tolerance thresholds value $\tau, \sigma^*_i = Rep(U_{BIO}, \sigma)$, $R^* = T_i \oplus h(U_{ID} \parallel \sigma^*_i)$, $PID^*_i = h(U_{ID} \parallel R^*)$, $PWR^*_i = h(U_{PW} \parallel R^*)$, and $R^*_i = h(PID^*_i \parallel PWR^*_i \parallel \sigma^*_i)$. Then, it checks the condition of $R^*_i \neq R_i$, if invalid, terminates the session.
- Otherwise, the SC authenticates the user and generates a random nonce N_i and calculates a secret key $SK_i = A^*_{th} \oplus h(PID^*_i \parallel PWR^*_i \parallel TID_i) \oplus h(\sigma^*_i \parallel R^*_i) = h(HGW_{ID} \parallel PID_i \parallel HGW_{MSK})$ consistent with the HGWN of the U_i. The U_i selects SN_j to have access to WSN services. Furthermore, smart-card computes $W_i = h(PID^*_i \parallel TID_i \parallel N_i)$ and $CT_i = E_{SK_i}[HGW_{ID}, SN_{ID}, W_i, N_i, TS_1]$, where $E_{SK_i}(M)$ signifies the plaintext message M's symmetric key encryption (e.g., AES) using the key SK_i, and TS_1 current timestamp. The SC finally sends $M_1 = [SN_{ID}, TID_i, CT_i]$ to HGWN publicly.

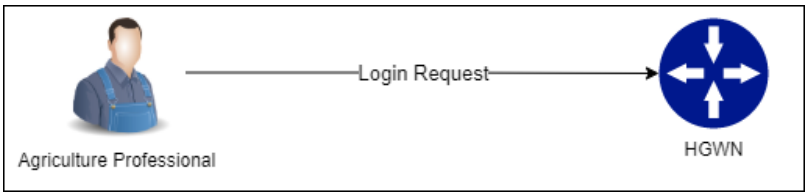

Figure 8. Login phase.

2.5.4. Authentication Phase

When the HGWN receives the login message, it checks to see if the SN_{ID} is stored in the HGWN database. If SN_{ID} is in the database, Case 1 will be down. Otherwise, it performs Case 2. Figures 9 and 10 depict distinct procedures individually for the two cases.
Case 1:

- The HGWN verifies the TS_1 by selecting a new timestamp TS_2 to check the freshness $|TS_2 - TS_1| \leq \triangle T$, where $\triangle T$ is the current timestamp. It calculates $SK_i^* = h(HGW_{ID} \parallel PID_i \parallel HGW_{MSK})$ based on the stored information in its database. After that, it decrypts $CT_i = D_{SK_i}[HGW_{ID}^*, SN_{ID}^*, W_i^*, N_i^*, TS_1^*]$, where D_{SK_i} depicts the decryption of a symmetric key using the key SK_i. After retrieving the information, HGWN verifies the timestamp $|TS_1^* - TS_1^{'}| \leq \triangle T$, where $TS_1^{'}$ is the message receiving time. If it holds, HGWN checks $HGW_{ID}^* \neq HGW_{ID}$, and $SN_{ID}^* \neq SN_{ID}$, and if these parameters are valid, it computes $W_i^{(**)} = h(PID_i \parallel TID_i \parallel N_i^*)$ based on the stored PID_i, and TID_i, then checks $W_i^{(**)} \neq W_i^*$, if it does not hold, it terminates the session. Otherwise, it selects a random nonce N_j to compute a shared secret key with the sensor node $SK_j = h(HGW_{ID} \parallel SN_{ID} \parallel HGW_{MSK}) = h(SN_{ID} \parallel SN_{MSK})$, $P_i = h(PID_i \parallel N_j \parallel N_i^* \parallel TS_2)$, and $CT_j = E_{SK_j}[HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2]$, and sends an authentication message $M_2 = SN_{ID}, CT_j$ to the sensor node via a public channel.
- The sensor node SN_i receives the message and decrypts $CT_j = D_{A_j}[HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2]$ using the stored key $A_j = h(SN_{ID} \parallel SN_{MSK})$ stored in the memory to obtain the information. Later, SNi checks the freshness of the timestamp $|TS_2^{'} - TS_2^*| \leq$, where TS_2^* is the message M_2 received time, if not fresh, terminates the session; otherwise, it computes $P_i^* = h(PID_i \parallel N_j^{'} \parallel N_i^{'} \parallel TS_2^{'})$ and checks$P_i^* \neq P_i^{'}$, if it does not hold, SN_i terminates the session. After that, SN_i calculates $Z_i = h(PID_i \oplus N_i^* \oplus N_j)$, $SK_{U \to SN} = h(HGW_{ID} \parallel SN_{ID} \parallel PID_i \parallel N_j)$, where $SK_{U ß SN}$ is a shared key between user and sensor node, and $B_i = h(SK_{U \to SN} \parallel TS_3)$. Then, TS_2 sends a replay authentication message $M_3 = SN_{ID}, B_i, Z_i, TS_3$ to the user via an open channel.
- Upon receiving M_3, it checks the freshness of the timestamp $|TS_3 - TS_3^*| \leq \triangle T$, if not fresh, it terminates the session; otherwise, it computes $N_j^{'} = Z_i \oplus h(PID_i^* \oplus N_i)$ using the previously computed $PID_i^* = h(U_{ID} \parallel R^*)$, $SK_{U \to SN}^* = h(HGW_{ID} \parallel SN_{ID} \parallel PID_i^* \parallel N_j \parallel N_j^{'})$, and $B_i^* = h(SK_{U \to SN}^* \parallel TS_3)$. Finally, U_i verifies $B_i^* \neq B_i$, if it holds, it ensures that U_i and SN_i share the same session key and store it for the future communication.

Case 2:

- The FGWN calculates $SK_{if} = A_{tf}^* \oplus h(FGW_{ID} \parallel SN_{ID} \parallel FGW_{MSK}) = h(SN_{ID} \parallel SN_{MSK})$, then it extracts PID_i corresponding to TID_i and generates a nonce number N_f, and computes $V_i = h(TID_i \parallel N_f \parallel TS_4)$, $CT_f = E_{SK_{if}}[HGW_{ID}, FGW_{ID}, SN_{PID}, PID_i, N_f, V_i, TS_4]$, then it sends $M_5 = SN_{ID}, CT_f$ to the sensor node SN_i.
- Upon receiving M_5, the SN_i decrypts the message $CT_f = E_{A_j}[HGW_{ID}, FGW_{ID}, SN_{ID}, PID_i, N_f, V_i, TS_4]$, using the key A_j to obtain information. Then, it checks the freshness of the timestamp $|TS_4^* - TS_4^{'}| \leq \triangle T$, and checks $SN_{ID}^* \neq SN_{ID}$. If holds, SN_i calculates $V_i^{'} = h(TID_i \parallel N_f^* \parallel TS_4^*)$ and verifies $V_i^{'} \neq V_i^*$, if does not hold,

ends session. Otherwise; SN_i generates a random nonce Nc_f, and computes $F_i = h(PID_i \oplus N_f \oplus Nc_f)$, $Q_i = h(FGW_{ID}^* \parallel Nc_f \parallel TS_5)$ and sends $M_6 = SN_{ID}, F_i, Q_i, TS_5$ to the FGWN.

- The FGWN receives M_6, it checks the freshness of the timestamp TS_5, and computes $Nc_f^* = F_i \oplus h(PID_i \oplus N_f)$, and $Q_i^* = h(FGW_{ID} \parallel SN_{ID} \parallel PID_i \parallel N_f \parallel Nc_f^* \parallel TS_5)$. Then, it validates $Q_i^* \neq Q_i$, if holds, the FGWN calculates $SK_{ef} = h(FGW_{ID} \parallel SN_{ID} \parallel PID_i \parallel N_f \parallel Nc_f^* \parallel TS_5)$, and $CT_{if} = E_{SK_{ef}}[FGW_{ID}, SN_{ID}, PID_i, N_f, Nc_f^*, T S_6]$. Finally, FGWN prepares and sends the $M_7 = SN_{ID}, CT_{if}$ to the U_i.
- After receiving M_7, the U_i calculates key $SK_u = A_{if}^* \oplus h(PID_i^* \parallel PWR_i^* \parallel TID_i) \oplus h(\sigma_i^* \parallel R_i^*) = h(FGW_{ID} \parallel PID_i \parallel FGW_{MSK})$, and decrypts $CT_{if} = D_{SK_u}[FGW_{ID}, S N_{ID}, PID_i, N_f, Nc_f^*, TS_6]$ to obtain information. After retrieving the data, it checks the freshness of the timestamps TS_6. Furthermore, the sensor identity is SN_{ID}. If it holds, U_i generates a random nonce Nc_u and computes $D_i = h(PID_i \parallel N_f \parallel Nc_f^* \parallel Nc_u)$, and also computes shared session key as $SK_{U \to SN} = h(HGW_{ID} \parallel FGW_{ID}^* \parallel SN_{ID} \parallel PID_i^* \parallel Nc_f \parallel N_j \parallel N_j^{'})$, and$J_i = h(SK_{U \to SN} \parallel TS_7)$. Finally, U_i sends $M_8 = SN_{ID}, D_i, J_i, TS_7$ to the sensor node.
- The SN_i receives M_8 and starts checking the freshness of the timestamp TS_7, then calculates $Nc_u^* = D_i \oplus h(PID_i \parallel N_f \parallel N_f)$ and session key $SK_{U \to SN}^* = h(HGW_{ID}^* \parallel FGW_{ID}^* \parallel SN_{ID} \parallel PID_i \parallel N_f \parallel Nc_f^* \parallel Nc_u)$, and $J_i^* = h(SK_{U \to SN}^* \parallel TS_7)$. After that, the SN_i checks the condition $J_i^* \neq J_i$, and if it holds, the agriculture professional and sensor node are successfully and mutually authenticated.

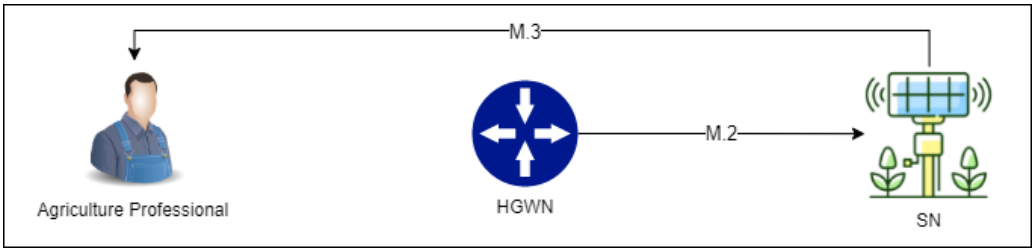

Figure 9. Authentication phase (Case 1).

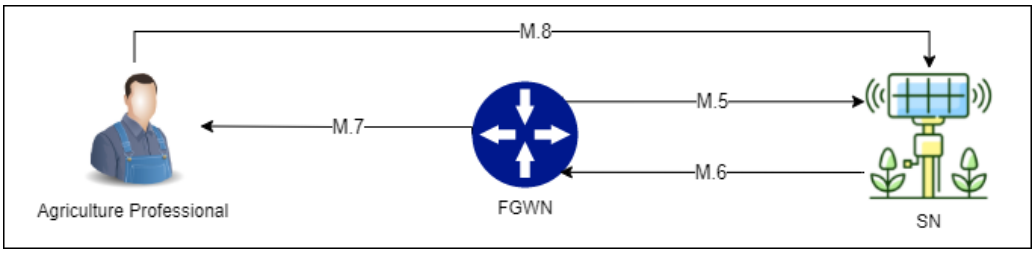

Figure 10. Authentication phase (Case 2).

2.6. Proof of Authentication Using BAN Logic

This section applies the Burrows–Abadi–Needham logic (BAN) to the proposed scheme to conduct a formal analysis. The BAN logic [50,51] is used widely to ensure the security of the key agreement-based authentication protocol [24,29,44]. First, communication parties establish the protocol's accuracy: the user Ui and the sensor node Sn, which exchange a freshly formed session key after the execution of the protocol. We begin by illustrating the BAN logic with the following specific notations:

- $P| \equiv X$: The principal P is convinced that the announcement X is valid.
- $P \triangleleft X$: P examines X, which indicates that P has received a message containing X that can be read by P.
- $P| \sim X$: : P once stated X, which signifies that P ∣ X as P once said it sometime.
- $P| \Rightarrow X$: P commands X completely, believing X is trustworthy (Jurisdiction over X).

- $\#(X)$: Because the message X is new, no entity has previously sent a message containing X.
- $P| \equiv Q \overset{SK}{\longleftrightarrow} P$: P and Q communicate via SK (shared key).
- $P \overset{SK}{\longleftrightarrow} Q$: P and Q share SK as a secret.
- $< X >_Y$: In conjunction with the formula Y, the formula X is utilized.
- (X): X is a hashed value in the formula.
- (X, Y): After that, the X and Y formulae are concatenated and hashed.
- $(X, Y)_k$: Using the key k to hash the formulae X and Y.

In the light of forgoing explanation of specific notations, we present the following rules for formalizing the logical postulates of BAN logic:
Message meaning rule: For shared secret keys (Rule 1):

$$\frac{P| \equiv Q \overset{k}{\longleftrightarrow} P, P \triangleleft X_k}{P| \equiv Q| \sim X}$$

P trusts Q if it believes k is shared with Q and sees X is encrypted under k.
Nonce verification rule (Rule 2):

$$\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$$

If P believes X was recently expressed (freshness) and Q once said X, P believes that Q believes X.
Jurisdiction rule (Rule 3):

$$\frac{P| \equiv Q)| \equiv X, P| \equiv Q| \Rightarrow X}{P| \equiv X}$$

If P believes that Q has jurisdiction over X and Q believes that a file contains X, P believes X as well.
Freshness rule (Rule 4):

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$$

If one of the components in the formula is known to be fresh, the complete formula must be fresh.
Belief rule (Rule 5):

$$\frac{P| \equiv Q| \equiv (X, Y)}{P| \equiv Q| \equiv (X)}$$

If P believes that Q believes in the message set (X, Y), then P also believes that Q believes in message X. Session key rule: For shared secret keys (Rule 6):

$$\frac{P| \equiv \#(X), P| \equiv Q| \equiv X}{P| \equiv P| \overset{k}{\longleftrightarrow} Q}$$

If P believes the shared session key is fresh, P and Q are said to believe X. The session key k with Q is then believed by P. Hence, the proposed scheme should meet the following goals, according to the BAN logic's analytic procedures:
Goal 1. $HGWN| \equiv (U_i \overset{SK}{\longleftrightarrow} HGWN)$;
Goal 2. $HGWN| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} HGWN)$;
Goal 3. $SN_j| \equiv (HGWN \overset{SK}{\longleftrightarrow} SN_j)$;
Goal 4. $SN_j| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} SN_j)$.
Goal 5. $HGWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} HGWN)$;

Goal 6. $HGWN| \equiv SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} HGWN)$.
Goal 7. $U_i| \equiv (HGWN \overset{SK}{\longleftrightarrow} U_i)$;
Goal 8. $U_i| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} U_i)$;
Goal 9. $FGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} FGWN)$.
Goal 10. $FGWN| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} FGWN)$.
Goal 11. $HGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} HGWN)$.
Goal 12. $HGWN| \equiv FGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} HGWN)$.
Goal 13. $FGWN| \equiv (U_i \overset{SK}{\longleftrightarrow} FGWN)$;
Goal 14. $FGWN| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} FGWN)$.
Goal 15. $SN_j| \equiv (FGWN \overset{SK}{\longleftrightarrow} SN_j)$;
Goal 16. $SN_j| \equiv FGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} SN_j)$;
Goal 17. $FGWN| \equiv (SN_j \overset{SK}{\longleftrightarrow} FGWN)$;
Goal 18. $FGWN| \equiv SN_j| \equiv (SN_j \overset{SK}{\longleftrightarrow} FGWN)$.
Goal 19. $U_i| \equiv (FGWN \overset{SK}{\longleftrightarrow} U_i)$;
Goal 20. $U_i| \equiv FGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} U_i)$.

To simplify the analysis between U_i and SN_j, we first idealize the transmitted messages of our proposed scheme, which are as follows:

Message 1: $U_i \rightarrow HGWN$: $HGW_{ID}, SN_{ID}, W_i, N_i, TS_1$.
Message 2: $HGWN \rightarrow SN_j$: $HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2$.
Message 3: $SN_j \rightarrow HGWN$: SN_{ID}, B_i, Z_i, TS_3.
Message 4: $HGWN \rightarrow Ui$: $HGW_{ID}, SN_{ID}, PID_i^*, N_j, N_j^{'}$.
Message 5: $HGWN \rightarrow FGWN$: $SN_{ID}, PID_i, HGW_{ID}, A_{th}$.
Message 6: $FGWN \rightarrow HGWN$: $A_{th}, HGW_{ID} :< PID_i, FGW_{ID} > A_{tf}, PID_i$.
Message 7: $HGWN \rightarrow U_i$: $A_{th} :< HGW_{ID}, SN_{ID}, PID_i^*, N_j, N_j^{'} > A_{tf}, FGW_{ID}$.
Message 8: $U_i \rightarrow FGWN$: $FGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2$.
Message 9: $FGWN \rightarrow SN_j$: $FGW_{ID}, SN_{ID}, PID_i, N_f, Nc_f^*, TS_6$.
Message 10: $SN_j \rightarrow FGWN$: SN_{ID}, F_i, Q_i, TS_5.
Message 11: $FGWN \rightarrow U_i$: $FGW_{ID}, SN_{ID}, PID_i, N_f, Nc_f^*, TS_6$.

Based on our proposed scheme, we made the following initial state assumptions:

A1: $U_i| \equiv \#(N_i^*, N_j, N_f, Nc_f^*)$;
A2: $HGWN| \equiv \#(N_i^*, N_j, N_f)$;
A3: $SN_j| \equiv \#(N_i^*, N_j, N_f, Nc_f^*)$;
A4: $FGWN| \equiv \#(N_i^*, N_j, N_f)$;
C1: $U_i| \equiv (U_i \overset{K_i}{\longleftrightarrow} HGWN)$;
C2: $HGWN| \equiv (HGWN \overset{P_j}{\longleftrightarrow} SN_j)$;
C3: $SN_j| \equiv (SN_j \overset{P_j}{\longleftrightarrow} HGWN)$;
C4: $HGWN| \equiv (HGWN \overset{PID_i}{\longleftrightarrow} U_i)$;
C5: $U_i| \equiv (U_i \overset{SK_{if}, A3}{\longleftrightarrow} FGWN)$;
C6: $FGWN| \equiv (FGWN \overset{SK_{if}, N_f, A5}{\longleftrightarrow} SN_j)$;
C7: $SN_j| \equiv (SN_j \overset{A6}{\longleftrightarrow} SN_jFGWN)$;
C8: $FGWN| \equiv (FGWN \overset{N_f, A7}{\longleftrightarrow} Ui)$;

C9: $HGWN| \equiv (HGWN \overset{Nc_f^*}{\longleftrightarrow} FGWN)$;
C10: $FGWN| \equiv (FGWN \overset{SK_{if}}{\longleftrightarrow} HGWN)$;
B1: $HGWN| \equiv (U_i \rightarrow N_i^*)$;
B2: $SN_j| \equiv (HGWN \rightarrow N_j)$;
B3: $HGWN| \equiv (SN_j \rightarrow N_i^{'})$;
B4: $U_i| \equiv (HGWN \rightarrow N_j^{'})$;
B5: $FGWN| \equiv (U_i \rightarrow N_f)$;
B6: $SN_j| \equiv (FGWN \rightarrow N_f)$;
B7: $FGWN| \equiv (SN_j \rightarrow Nc_f^*)$;
B8: $U_i| \equiv (FGWN \rightarrow N_f)$;
B9: $HGWN| \equiv (FGWN \rightarrow N_i^*)$;
B10: $FGWN| \equiv (HGWN \rightarrow N_f)$;

Additionally, we demonstrate the robustness of the present scheme based on BAN logic rules by showing that Ui and SNj have the same shared SK session key to communicate securely while still accomplishing the required goals under initial assumptions. The following are the descriptions of the inside information:

We might get the following based on message 1:

$$S1 : HGWN \triangleleft HGW_ID, SN_{ID}, W_i, N_i, TS_1.$$

We apply the message meaning rule to S1 and Assumption C1 to get:

$$S2 : HGWN| \equiv U_i| \sim N_i.$$

Then, we use the freshness conjunctenation rule and the nonce verification rule to get the final observations based on assumptions A2 and Step 2:

$$S3 : HGWN| \equiv U_i| \equiv N_i.$$

When B1, S3, and the jurisdiction rule are applied, we get:

$$S4 : HGWN| \equiv N_i.$$

We apply the session key rule to the assumptions A2 and S3 to get:

$$S5 : HGWN| \equiv (U_i \overset{SK}{\longleftrightarrow} HGWN). \textbf{(Goal 1)}$$

The nonce verification rule and jurisdiction rule are applied to S5 and assumption A2 to obtain:

$$S6 : HGWN| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} HGWN). \textbf{(Goal 2)}$$

We could obtain the following from message 2:

$$S7 : SNj \triangleleft SN_{ID}, B_i, N_j, Z_i, TS_3$$

If we apply the message meaning rule to C2, S7, we get:

$$S8 : SN_j| \equiv HGWN| \sim N_j$$

The freshness conjunctenation rule and nonce verification rule are applied to assumptions A3 and S8 to obtain:

$$S9 : SNj| \equiv HGWN| \equiv N_j.$$

We use the jurisdiction rule to get the following results from Step 9 and B2:

$$S10 : SN_j| \equiv N_j$$

S9 and A3 are combined with the session key rule to produce:

$$S11 : SNj|HGWN(SK)SNj. \textbf{ (Goal 3)}$$

S11 and assumption A3 are applied to the nonce verification rule to achieve:

$$S12 : SNj| \equiv HGWN| \equiv HGWN \overset{SK}{\longleftrightarrow} SNj. \textbf{ (Goal 4)}$$

We may get the following from message 3:

$$S13 : HGWN \triangleleft SN_{ID}, B_i, N_j, Z_i, TS_3.$$

We apply the message meaning rule to S13 and assumption C3 to get:

$$S14 : HGWN| \equiv SNj| \sim N_j.$$

The freshness conjunctenation rule and nonce verification rule are applied to assumptions A2 and S14 to obtain:

$$S15 : HGWN| \equiv SN_j| \equiv N_j.$$

We apply the jurisdiction rule in S15 and B3 to obtain:

$$S16 : HGWN| \equiv N_j.$$

We apply the session key rule to the assumptions A2 and S15 to obtain:

$$S17 : HGWN| \equiv (SNj \overset{SK}{\longleftrightarrow} HGWN). \textbf{ (Goal 5)}$$

We use the nonce verification rule to derive the following result from S17 and assumption A2:

$$S18 : HGWN| \equiv SNj| \equiv SNj \overset{SK}{\longleftrightarrow} HGWN. \textbf{ (Goal 6)}$$

We might get the following based on message 4:

$$S19 : Ui \triangleleft HGW_{ID}, SN_{ID}, PID_i^*, N_j, N_j^{'}, TS_4$$

We apply the message meaning rule to S19 and assumption C4 to obtain:

$$S20 : Ui| \equiv HGWN| \sim N_j^{'}.$$

Applying the freshness concatenation and nonce verification rules to assumptions A1 and S20, we obtain:

$$S21 : Ui| \equiv HGWN| \equiv N_j^{'}.$$

We obtain jurisdiction by using the jurisdiction rule in line with B4 and S21:

$$S22 : Ui| \equiv N_j^{'}.$$

We use the session key rule-following A1 and S21 to achieve:

$$S23 : Ui|HGWN \overset{SK}{\longleftrightarrow} Ui. \textbf{ (Goal 7)}$$

We use the nonce verification rule to derive the following result from S23 and assumption A1:

$$S24 : Ui| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} Ui).$$ **(Goal 8)**

According to message 5, we might be able to obtain:

$$S25 : FGWN \triangleleft SN_{ID}, PID_i, HGW_{ID}, A_{th}.$$

S25 and assumption C9 are subjected to the message meaning rule to obtain:

$$S26 : FGWN| \equiv HGWN| \sim A_{th}$$

From S26, we apply the nonce verification rule to get:

$$S27 : FGWN| \equiv HGWN| \equiv A_{th}.$$

We use the jurisdiction rule to get the following from S27 and B8:

$$S28 : FGWN| \equiv A_{th}.$$

According to S27 and S28, the session key rule is applied, and we get:

$$S29 : FGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} FGWN).$$ **(Goal 9)**

According to S29, we apply the nonce verification rule to get:

$$S30 : FGWN| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} FGWN). (Goal10)$$

We might be able to access this based on message 6:

$$S31 : HGWN \triangleleft A_{th}, HGW_{ID} :< PID_i, FGW_{ID} > A_{tf}, PID_i.$$

Using the message meaning rule with S31 and Assumption C10, and we get at:

$$S32 : HGWN| \equiv FGWN| \sim A_{tf}.$$

We use the nonce verification rule to get the following observations from S32:

$$S33 : HGWN| \equiv FGWN| \equiv A_{tf}.$$

We apply the jurisdiction rule to S33 and B9 to obtain:

$$S34 : HGWN| \equiv A_{tf}.$$

According to S33 and S34, the session key rule is applied, we get:

$$S35 : HGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} HGWN).$$ **(Goal 11)**

We apply the nonce verification procedure following S35 to obtain:

$$S36 : HGWN| \equiv FGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} HGWN).$$ **(Goal 12)**

We might be able to obtain this based on message 7:

$$S37 : Ui \triangleleft A_{th} :< HGW_{ID}, SN_{ID}, PID_i^*, N_j, N_j^{'} > A_{tf}, FGW_{ID}.$$

We apply the message meaning rule to S37 and assumption C5 to obtain:

$$S38 : Ui| \equiv HGWN| \sim N_j^{'}.$$

Applying the freshness conjuncatenation and nonce verification rules to assumptions A4 and S38, we obtain:

$$S39 : Ui| \equiv HGWN| \equiv N_j^{'}.$$

We apply the jurisdiction rule by B9 and S39 to get:

$$S40 : Ui| \equiv N_j^{'}.$$

We apply the session key rule-following A1, A4, and Step 40 to obtain:

$$S41 : Ui| \equiv HGWN \overset{SK}{\longleftrightarrow} Ui. \textbf{ (Goal 7)}$$

We apply the nonce verification rule to Step 41 and assumption A1 to get:

$$S42 : Ui| \equiv HGWN| \equiv (HGWN \overset{SK}{\longleftrightarrow} Ui). (Goal8)$$

We may obtain according to message 8:

$$S43 : FGWN \triangleleft FGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2.$$

We use the message meaning rule from S43 and assumption C10:

$$S44 : FGWN| \equiv Ui| \sim N_i^*.$$

We apply the freshness conjuncatenation rule and the nonce verification rule from assumptions A4 and S44 to:

$$S45 : FGWN| \equiv Ui| \equiv N_i^*$$

From S45 and B5, we apply the rule of competence to obtain:

$$S46 : FGWN| \equiv N_i^*.$$

The session key rule is applied according to A4 and S45 and 46; thus, we obtain:

$$S47 : FGWN| \equiv (Ui \overset{SK}{\longleftrightarrow} FGWN). \textbf{ (Goal 13)}$$

Under A4 and S47, the nonce verification rule is to be established:

$$S48 : FGWN| \equiv Ui| \equiv (Ui \overset{SK}{\longleftrightarrow} FGWN). \textbf{ (Goal 14)}$$

We may obtain according to message 9:

$$S49 : SNj \triangleleft FGW_{ID}, SN_{ID}, PID_i, N_f, Nc_f^*, TS_6$$

We apply the message meaning rule according to C2, B5, and S49:

$$S50 : SNj| \equiv FGWN| \sim N_f, Nc_f^*$$

We apply the freshness conjuncatenation rules and the nonce verification rule for assumptions A3 and S50:

$$S51 : SNj| \equiv FGWN| \equiv N_f, Nc_f^*.$$

We use the rule of jurisdiction to obtain the following S51 and B6 results:

$$S52 : SNj| \equiv N_f, Nc_f^*.$$

From S52 and A3 and to obtain the key session rule:

$$S53 : SNj| \equiv FGWN \overset{SK}{\longleftrightarrow} SNj. \textbf{ (Goal 15)}$$

From S52 and A3 and to obtain the key session rule:

$$S54 : SNj| \equiv FGWN| \equiv FGWN \overset{SK}{\longleftrightarrow} SNj. \textbf{ (Goal 16)}$$

According to message 10, we could get:

$$S55 : FGWN \triangleleft SN_{ID}, F_i, N_j, Q_i, TS_5.$$

From S55 and assumption C7, we use the message meaning rule:

$$S56 : FGWN| \equiv SNj| \sim N_j.$$

We obtain by applying the freshness conjuncatenation and nonce verification rules to assumptions A4 and S56:

$$S57 : FGWN| \equiv SNj| \equiv N_j,$$

We apply the jurisdiction rule-following S57 and B7 to obtain:

$$S58 : FGWN| \equiv N_j.$$

We apply the session key rule to the assumptions A4 and S58 to obtain:

$$S59 : FGWN| \equiv SNj \overset{SK}{\longleftrightarrow} FGWN). \textbf{ (Goal 17)}$$

We obtain by applying the nonce verification rule to S59 and assumption A4:

$$S60 : FGWN| \equiv SNj| \equiv SNj \overset{SK}{\longleftrightarrow} FGWN. \textbf{ (Goal 18)}$$

According to message 11, we might be able to obtain:

$$S61 : Ui \triangleleft SN_{ID}, F_i, Q_i, Nc_f^*, TS_5.$$

We apply the message meaning rule to Step 61 and assumption C8 to obtain:

$$S62 : Ui| \equiv FGWN| \sim Nc_f^*.$$

We obtain by applying the freshness conjuncatenation and nonce verification rules to the assumptions A1 and S62:

$$S63 : Ui| \equiv FGWN| \equiv Nc_f^*.$$

We apply the jurisdiction rule by A1 and S63 to get:

$$S64 : Ui| \equiv Nc_f^*.$$

We apply the session key rule by A1 and S63 and S64 to achieve:

$$S65 : Ui| \equiv FGWN \overset{SK}{\longleftrightarrow} Ui. \textbf{ (Goal 19)}$$

Applying the nonce verification rule to S65 and assumption A1, we obtain:

$$S66 : Ui| \equiv FGWN| \equiv (FGWN \overset{SK}{\longleftrightarrow} Ui). \textbf{ (Goal 20)}$$

From Steps 5–6, 11, 12, 17, 18, 23, 24, 29, 30,35, 36, 41, 42, 48, 47, 53, 54, 59, 60, 65, and 66, it is obvious that our scheme accomplishes all of the goals (Goals 1–20). Both Ui and SNj

assume they share a secure session key $SK_j = h(HGW_{ID} \parallel SN_{ID} \parallel HGW_{MSK}) = h(SN_{ID} \parallel SN_{MSK})$, via HGWN/FGWN.

2.7. Formal Security Verification Using AVISPA Tool

This section demonstrates the proposed scheme's security validation using the AVISPA tool, a widely used and well-known security validation tool [52]. The security verification code was written using the AVISPA tool based on High-Level Protocol Specification Language (HLPSL). It is a role-oriented language composed of primary roles that define each participant system and composition roles representing scenarios connected to fundamental roles [53]. The intruder, who is always represented by "I" and explained using the Dolev–Yao model, also plays a special role. The intruder plays a critical part in implementing the protocol and interacts with several other functions in the system. Using the HLPSL2IF translator, the HLPSL protocol specification is transformed into an intermediate format (IF). After that, the intermediate format is examined using one of four different backends: CL-AtSe, OFMC, SATMC, or TA4SP. Each backend uses a variety of automated analytical tools to detect potential attacks against known models.

Specifying Scheme Roles

This section shows our scheme employing HLPSL in two scenarios. The first scenario, as shown in Figures 11–14, carries out the basic functions of UI's, SA's, HWGN, and SNj sensor nodes during the user registration, log-in and authentication, and key agreement phases (Case 1). In the second scenario, we integrated user roles Ui, SA, HWGN, and sensor node SNj throughout the user registration, log-in and authentication phase, and key agreement phase (Case 2).

The details of the role of the initiator, the user Ui, are shown in Figure 11 for Case 1. The start signal is first received by Ui, which changes its state from 0 to 1. The variable status is used to keep track of the current state. Using the SND() function, Ui securely provides PIDi, PWRi to the SA during the registration phase of the user. The SCi smart card is received in Ui from the SA containing information (IDGWNh, Aih), (IDGWNf, Aif), TIDi, Gen (.), Rep(.), h (.) and t, changing the status from 1 to 2. Ui delivers the log-in request message M1 = IDSNj, TIDi, Ci to the HGWN across an open channel during the log-in phase. The secret declaration (X, id, A) states that the protocol identification of agent A is id. The information X is kept secret from agent A. For example, secret (IDi, PWRi, BIOi, sp1, Ui) implies that IDi, PWi, and secret number R are kept secret from Ui only, as determined by the protocol identifier sp1. Declaration witness (Ui, HGWN, ui hgwn ru, TS1') implies that Ui recently generated the HGWN timestamp TS1. During the authentication and key agreement phases, Ui gets the acknowledgment message M3 = hIDSNj, Gi, Hi, TS3i through a public channel from the sensor node SNj and updates its state from 2 to 3. Finally, Ui checks SNj's authenticity by comparing SNj's timestamp TS3 to the randomly generated nonce RNj generated by the declaration request (SNj, Ui, sn ui rk, RNj'). Notably, the type declaration channel (dy) reflects the communication channel using the Dolev–Yao threat model, implying that an intruder can view, intercept, or change messages sent via an insecure public channel. Sentence A denotes that the function is carried out by the agent identified by variable A.

```
role user (Ui, SA, HGWN, SNj : agent, H: hash_func,
SKuisa : symmetric_key, Snd, Rcv: channel(dy))
played_by Ui
def=
local State: nat,
FGWN : agent,
PIDi, PWRi, Di, IDi, RIDi, PWi, BIOi, R: text,
Aih, Aif, T , RTSi, TIDj,TIDi, IDGWNh, IDGWNf, TS1, TS2, TS3: text,
Gen, Rep: hash_func,
MKGWNh,MKGWNf,IDSNj, MKSNj,Ki, Kj, Bi, Ci, RNi, RNk, RNj: text,
F: hash_func
const sp1, sp2, sp3, sp4, sp5, ui_hgwn_ru, ui_hgwn_tsu,
sn_ui_rk, sn_ui_tsk : protocol_id
init State := 0
transition
1. State = 0 /\ Rcv(start) =|>
State' := 1 /\ PIDi':= H(IDi.R)
            /\ PWRi' := H(PWi.R)
            /\ Snd({PIDi'.PWRi'}_SKuisa)
            /\ secret({IDi,PWRi,BIOi}, sp1, {Ui})
% Receive registration reply from the SA securely
2. State = 1 /\ Rcv({IDGWNh.xor(H(IDGWNh.H(IDi.R).MKGWNh).H(H(IDi.R).
H(PWi.R).TIDi')).IDGWNf.xor(H(IDGWNf.H(IDi.R).MKGWNf), H(H(IDi.R).
H(PWi.R).TIDi')).TIDi'.Gen.Rep.H.T}_SKuisa) =|>
State' := 2 /\ secret({IDGWNh}, sp2, {Ui,SA,HGWN})
/\ secret({MKGWNh}, sp3, {SA,HGWN})
/\ secret({IDGWNf}, sp4,{Ui,SA,FGWN})
/\ secret({MKGWNf}, sp5, {SA,FGWN})
% Login phase
% Send message Ml = cIDSNj, Ci> to HGWN via public channel
/\ RNi':= new()
/\ TS1' := new()
/\ Ki':= H(IDGWNh.H(IDi.R).MKGWNh)
/\ Bi' :=H(H(IDi.R).TIDi.RNi')
/\ Ci' := {IDGWNh.IDSNj.Bi'.RNi'.TS1' }_Ki'
/\ Snd(IDSNj.TIDi'.Ci')
% Ui has freshly generated the value TS1 for HGWN
/\ witness(Ui, HGWN, ui_hgwn_ru, TS1')
% Ui has freshly generated the value RNi for HGWN
/\ witness(Ui, HGWN, ui_hgwn_tsu, RNi')
3. State = 2 /\ Rcv(IDSNj.xor(H(xor(H(IDi.R),RNi')),RNj').
H(H(IDGWNh.IDSNj.H(xor(H(IDi.R),RNi')).RNj').TS3')) =|>
State' := 3 /\ request(SNj, Ui, sn_ui_rk, TS3')
/\ request(SNj, Ui, sn_ui_tsk,RNj')
end role
```

Figure 11. role in HLPSL.

Similarly, Figure 12 shows the role of the home gateway role in HLPSL. The role starts by receiving the message (IDSNj.TIDi'.IDGWNh.IDSNj.H(H(IDi.R).TIDi'.RNi'). RNi'.TS1' _H(IDGWNh.H(IDi.R).MKGWNh)) from the user Ui using the operation Rcv (). The declaration secret (IDi, PWi, R, sp1, Ui) indicates that the values IDi, PWi, R sent secularly to the user Ui using the protocol sp1. While the statement secret (IDGWNh, sp2, Ui, SA, HGWN) specifies the identity of the home gateway among the Ui and SA by the HGWN using protocol ID sp2. Furthermore, the indication secret (MKGWNh, sp3, SA, HGWN) shows that the master key is shared between the SA and the HGWN. While the identity of the foreign gateway is shared securely using the declaration secret (IDGWNf, sp4, Ui, SA, FGWN) amongst the Ui, and SA using the protocol ID sp4. The foreign gateway shares its master key with SA using the declaration secret (MKGWNf, sp5, SA, FGWN). Later, the home gateway sends the message (IDSNj.Fi') to the sensor using Snd(). The declaration witness (HGWN, SNj, hgwn_sn_rf, TS2') indicates that the HGWN freshly generates TS2' for the SNj. Furthermore, the HGWN is freshly generating random nonce RNi' for the sensor using the declaration witness (HGWN, SNj, hgwn_sn_tsf, RNk'). The HGWN accepts the legitimacy of the Ui by checking the freshness of the TS1 using the declaration request (Ui, HGWN, ui_hgwn_ru, TS1'), and also accepts the legitimacy of the user by checking RNi' through the indication request (Ui, HGWN, ui_hgwn_tsu, RNi').

```
role homegateway (Ui, SA, HGWN, SNj : agent, H: hash_func,
 Snd, Rcv: channel(dy))
played_by HGWN
def=
 local State: nat,
FGWN : agent,
 PWi, BIOi, S, Di, IDi,TIDi, IDGWNh, IDGWNf, TS1, TS2, TS3: text,
 F: hash_func,
R: text,
MKGWNh, MKGWNf, IDSNj, MKSNj, Ki, Kj, Bi, Ci, Fi, Gi, RNi, RNk, RNj, Gj: text
 const sp1, sp2, sp3, sp4, sp5, ui_hgwn_ru, ui_hgwn_tsu,
 hgwn_sn_rf, hgwn_sn_tsf: protocol_id
init State := 0
transition
% Login and authentication phases
% Recive message Msg1 from Ui via public channel
1. State = 0 /\ Rcv(IDSNj.TIDi'.{IDGWNh.IDSNj.H(H(IDi.R).TIDi'.RNi').
RNi'.TS1' }_H(IDGWNh.H(IDi.R).MKGWNh)) =|>
State' := 3 /\ secret({IDi,PWi,R}, sp1, {Ui })
/\ secret({IDGWNh}, sp2, {Ui,SA,HGWN})
/\ secret({MKGWNh}, sp3, {SA,HGWN})
/\ secret({IDGWNf}, sp4, {Ui,SA,FGWN})
/\ secret({MKGWNf}, sp5, {SA,FGWN})
% Authentication and key agreement phase % Send message M2 =clDSNj, Fi>
to sensor SNj via public channel
/\ TS2' := new()
/\ RNk' := new()
/\ Kj' := H(IDSNj.MKSNj)
/\ Di' := H(H(xor(H(IDi.R),RNi')).RNi' .RNk' .TS2')
/\ Fi' := {IDGWNh.IDSNj.H(xor(H(IDi.R),RNi')). RNi'.RNk'.Di'.TS2'}_Kj'
% Send message Msg2 to SNj via public channel
 /\ Snd(IDSNj.Fi')
% hgwn has freshly generated the values rf and TSf for SNj
 /\ witness(HGWN, SNj,  hgwn_sn_rf, TS2')
 /\ witness(HGWN, SNj, hgwn_sn_tsf, RNk')
% hgwns acceptance of the values ru and TSu generated for hgwn by Ui
 /\ request(Ui, HGWN, ui_hgwn_ru, TS1')
 /\ request(Ui, HGWN, ui_hgwn_tsu, RNi')
end role
```

Figure 12. The home gateway role in HLPSL.

In Figure 13, the role of the sensor node in HLPSL is illustrated. The role starts by receiving the message (IDSNj.IDGWNh.IDSNj.H(xor(H(IDi.K), RNi')). RNi' .RNk'.H(H(xor (H(IDi.K), RNi')). RNi'.RNk'.TS2').TS2'_H(IDSNj.MKSNj)) from the HGWN using the operation Rcv (). However, the role indicates the values IDi, PWi, R are shared securely to the user using the declarations secret (IDi, PWi, R, sp1, Ui). The declarations secret (IDGWNh, sp2, Ui, SA, HGWN), and secret (IDGWNf, sp4, Ui, SA, FGWN) specify that the identity of the home and foreign gateway is shared secretly among the Ui, and the SA. While the expressions secret (MKGWNh, sp3, SA, HGWN, and secret(MKGWNf, sp5, SA, FGWN) show that the master key of the home and foreign gateways is shared securely to the user Ui. Likewise, the user believes that the sensor freshly generates TS3', and RNj' for user. The user also acknowledges the HGWN's legality by confirming the TS2' timestamp using the declaration request (HGWN, SNj, hgwn sn rf, TS2'), and by validating the RNk's random nonce with the declaration request (HGWN, SNj, hgwn sn tsf, RNk').

The role of system administrator in HLPSL is shown in Figure 14. After the message (H(IDi.K).H(PWi.K)_SKuisa) is received from the user, it shares the values IDi, PWi, K securely using the protocol ID sp1. Later, it sends (IDGWNh.Aih'.IDGWNf.Aif'.TIDi'.Gen.Rep. H.T_SKuisa) and encrypts the message using the SKuisa secret key. Furthermore, it indicates that the identities by the declarations secret (IDGWNh, sp2, Ui, SA, HGWN), and secret (IDGWNf, sp4, Ui, SA, FGWN) are shared among Ui, SA, HGWN, and FGWN. The master keys MKGWNh, and MKGWNf are share secretly by the declarations secret (MKGWNh, sp3, SA, HGWN), and secret (MKGWNf, sp5, SA, FGWN) to the SA.

```
role sensornode (Ui, SA, HGWN, SNj : agent, H: hash_func,
 Snd, Rcv: channel(dy))
played_by SNj
def=
 local State: nat,
FGWN : agent,
 IDi, PWi, K, DIDi, PWRi, Aih, Aif, T: text, F: hash_func,
 TIDi, IDGWNh, IDGWNf, TS1, TS2, TS3, SKuisnj: text,
Gen, Rep: hash_func,
MKGWNh, MKGWNf, IDSNj, MKSNj, Pj, Ki, Kj, Bi, Ci, Di, Fi, Gi, Hi, RNi, RNk, RNj, BIOi: text
 const sp1, sp2, sp3, sp4, sp5, hgwn_sn_rf, hgwn_sn_tsf,
 sn_ui_rk, sn_ui_tsk: protocol_id
init State := 0
transition
% Login and authentication phases
% Recive message Msg2 from HGWN via public channel
1. State = 0 /\ Rcv(IDSNj.{IDGWNh.IDSNj.H(xor(H(IDi.K), RNi')).
RNi' .RNk'.H(H(xor(H(IDi.K),RNi')). RNi'.RNk'.TS2').TS2'}_H(IDSNj.MKSNj)) =|>
State' := 2 /\ secret({IDi,PWi,K}, sp1, {Ui})
/\ secret({IDGWNh}, sp2, {Ui,SA,HGWN})
/\ secret({MKGWNh}, sp3, {SA,HGWN})
/\ secret({IDGWNf}, sp4, {Ui,SA,FGWN})
 /\ secret({MKGWNf}, sp5, {SA,FGWN})
% Send M3 = 4DSNj, Gi, Hi, TS3> to Ui via public channel
/\ RNj' := new()
/\ TS3' := new()
/\ Gi' := xor(H(xor(H(IDi.K),RNi')),RNj')
 /\ SKuisnj' := H(IDGWNh.IDSNj.H(xor(H(IDi.K), RNi')).RNj')
/\ Hi' := H(SKuisnj.TS3')
 /\ Snd(IDSNj.Gi'.Hi'.TS3')
/\ witness(SNj, Ui, sn_ui_rk, TS3')
/\ witness(SNj, Ui, sn_ui_tsk, RNj')
/\ request(HGWN, SNj, hgwn_sn_rf, TS2')
/\ request(HGWN, SNj, hgwn_sn_tsf, RNk')
end role
```

Figure 13. Sensor node role in HLPSL.

```
role sensornode (Ui, SA, HGWN, SNj : agent, H: hash_func,
 Snd, Rcv: channel(dy))
played_by SNj
def=
 local State: nat,
FGWN : agent,
 IDi, PWi, K, DIDi, PWRi, Aih, Aif, T: text, F: hash_func,
 TIDi, IDGWNh, IDGWNf, TS1, TS2, TS3, SKuisnj: text,
Gen, Rep: hash_func,
MKGWNh, MKGWNf, IDSNj, MKSNj, Pj, Ki, Kj, Bi, Ci, Di, Fi, Gi, Hi, RNi, RNk, RNj, BIOi: text
 const sp1, sp2, sp3, sp4, sp5, hgwn_sn_rf, hgwn_sn_tsf,
 sn_ui_rk, sn_ui_tsk: protocol_id
init State := 0
transition
% Login and authentication phases
% Recive message Msg2 from HGWN via public channel
1. State = 0 /\ Rcv(IDSNj.{IDGWNh.IDSNj.H(xor(H(IDi.K), RNi')).
RNi' .RNk'.H(H(xor(H(IDi.K),RNi')). RNi'.RNk'.TS2').TS2'}_H(IDSNj.MKSNj)) =|>
State' := 2 /\ secret({IDi,PWi,K}, sp1, {Ui})
/\ secret({IDGWNh}, sp2, {Ui,SA,HGWN})
/\ secret({MKGWNh}, sp3, {SA,HGWN})
/\ secret({IDGWNf}, sp4, {Ui,SA,FGWN})
 /\ secret({MKGWNf}, sp5, {SA,FGWN})
% Send M3 = 4DSNj, Gi, Hi, TS3> to Ui via public channel
/\ RNj' := new()
/\ TS3' := new()
/\ Gi' := xor(H(xor(H(IDi.K),RNi')),RNj')
 /\ SKuisnj' := H(IDGWNh.IDSNj.H(xor(H(IDi.K), RNi')).RNj')
/\ Hi' := H(SKuisnj.TS3')
 /\ Snd(IDSNj.Gi'.Hi'.TS3')
/\ witness(SNj, Ui, sn_ui_rk, TS3')
/\ witness(SNj, Ui, sn_ui_tsk, RNj')
/\ request(HGWN, SNj, hgwn_sn_rf, TS2')
/\ request(HGWN, SNj, hgwn_sn_tsf, RNk')
end role
```

Figure 14. System administrator role in HLPSL.

Figure 15 shows the session, goal, and environmental roles of the proposed scheme. All primary roles of the session, including user, sa, hgwn, and sensor, are instances with concrete arguments. The HLPSL specification continually defines the top-level role (envi-

ronment). In addition, the proposed scheme has implemented five secrecy goals and three authentication goals:

Secrecy Goals:

secrecy_of sp1: Indicates that the IDi, PWRi, and BIOi are kept secret to the Ui.
secrecy_of sp2: States that the IDGWNh is shared securely to the Ui, SA, and HGWN.
secrecy_of sp3: This shows that the MKGWNh is kept secret to the SA and HGWN.
secrecy_of sp4: Indicates that the IDGWNf is shared among Ui, SA, and FGWN.
secrecy_of sp5: Indicates that the MKGWNf is kept secret to the SA and FGWN.

Authentication Goals:

authentication_on ui_hgwn_ru, ui_hgwn_tsu: It indicates that the user Ui generates TS1′ and RNi; which are freshly generated and perform a strong authentication with HGWN-based validity of these values.
authentication_on hgwn_sn_rf, hgwn_sn_tsf: It indicates that HGWN generates TS2′ and RNK′ freshly for the sensor and performs a strong authentication of the parameter's freshness.
authentication_on sn_ui_rk, sn_ui_tsk: It shows that the sensor generates a fresh TS3′ and RNj′ for the user and performs a strong authentication based on the validity of the values.

```
role session(Ui, SA, HGWN, SNj : agent, H: hash_func,
 SKuisa : symmetric_key)
def=
 local Tx1, Rv1, Tx2, Rv2, Tx3, Rv3, Tx4, Rv4: channel(dy)
 composition
 user (Ui, SA, HGWN, SNj, H, SKuisa, Tx1, Rv1)
 /\ systemadminstrator (Ui, SA, HGWN, SNj, H, SKuisa, Tx2, Rv2)
 /\ homegateway (Ui, SA, HGWN, SNj, H, Tx3, Rv3)
 /\ sensornode (Ui, SA, HGWN, SNj, H, Tx4, Rv4)
end role
role environment()
def=
 const ui, sa, hgwn, snj: agent, h, f: hash_func,
 skuisa: symmetric_key, sp1, sp2, sp3, sp4, sp5, ui_hgwn_ru,
 ui_hgwn_tsu, hgwn_sn_rf, hgwn_sn_tsf,
 sn_ui_rk, sn_ui_tsk: protocol_id, tsu, tsf, tsk: text
intruder_knowledge ={h, f, tsu, tsf, tsk}
 composition
 session(ui, sa, hgwn, snj, h, skuisa)
/\ session(i, sa, hgwn, snj, h, skuisa)
/\ session(ui, i, hgwn, snj, h, skuisa)
/\ session(ui, sa, i, snj, h, skuisa)
/\ session(ui, sa, hgwn, i, h, skuisa)
end role
goal
 secrecy_of sp1, sp2, sp3, sp4, sp5
 authentication_on ui_hgwn_ru, ui_hgwn_tsu
 authentication_on hgwn_sn_rf, hgwn_sn_tsf
 authentication_on sn_ui_rk, sn_ui_tsk
end goal
environment()
```

Figure 15. The role session, environment, and goals in HLPSL.

3. Results and Discussion

In this section, we provide a comprehensive discussion on the security and functional results of the proposed scheme with the related user authentication schemes applied for agriculture WSNs, such as D. Rangwani et al. [41], Dhillon and Kalra [38], J. Lee et al. [28], and A.Vangala et al. [45]. We first provide the results of the AVISPA tool presented in the earlier Section 2.7. Then, a theoretical security analysis on the way of providing security protection against various attacks is discussed. Finally, it illustrates the functionality of the proposed scheme in terms of communication and computation costs against other exiting authentication schemes.

3.1. The AVISPA Results

In the OFMC and CL-AtSe back-ends, the SPAN tool simulated the proposed scheme for both cases (Case 1 and Case 2) using AVISPA tool. The following evaluations are carried out in our scheme in both cases:

- Executability check on non-trivial HLPSL specifications: The proposed protocol model may not be completed due to modeling errors. As a result, the state unreachability of critical states in which an attack can occur, the AVISPA back-ends may not identify an attack, as mentioned in the protocol model. Consequently, an executability test is essential. Our initial HLPSL implementation shows that the executability test objectives in Figures 11–14 are met in both cases.
- Replay attack check: The OFMC and CL-AtSe back-ends search for a passive intruder to determine whether authentic agents can execute the specified protocol. The simulation results shown in Figures 16 and 17 reveal that our scheme is secure against replay attacks in both cases.
- Dolev–Yao model check: The AVISPA simulation, built on the OFMC and CLAtSe back-ends, detects man-in-the-middle and replay attacks. Figures 16 and 17 indicate indisputably that our scheme is secure when employed with these back-ends.

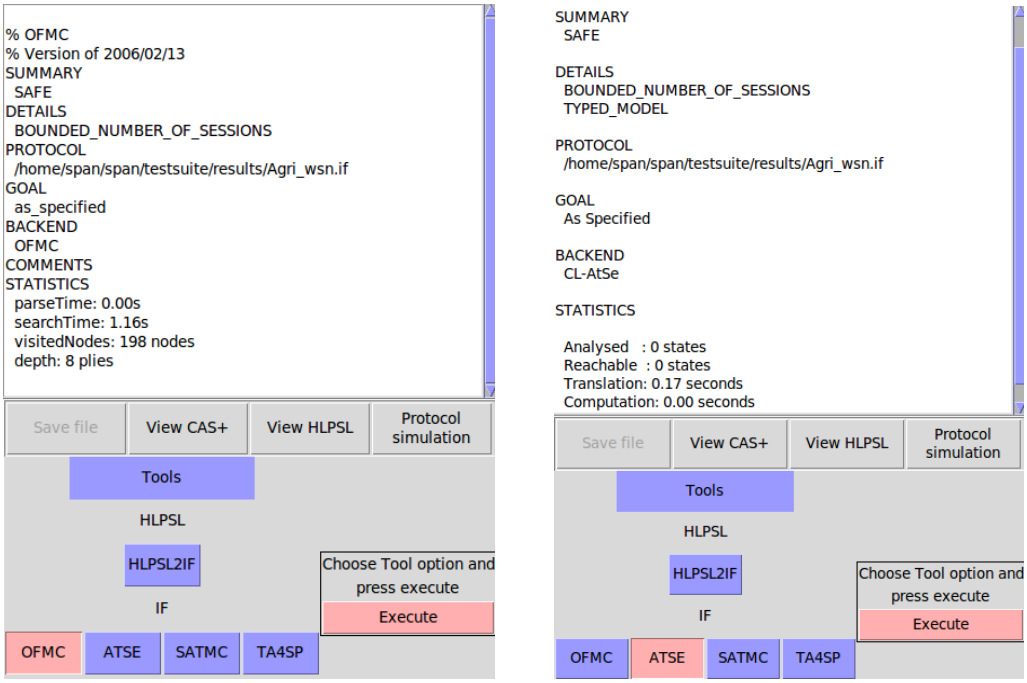

Figure 16. The simulation results using OFMC and CL-AtS back-ends in Case 1.

3.2. Security Features

This section details the proposed security analysis of security properties and resistance to various attacks against existing agriculture professional authentication schemes. It shows that the proposed scheme can resist a variety of security attacks and withstand multiple security features. Table 3 shows the comparison of the proposed scheme against other selected works in terms of security features. For example, it indicates that D. Rangwani et al. [41] and A.Vangala et al. [45] schemes are vulnerable to identity guessing, gateway, and sensor impersonation attacks. Furthermore, the A.Vangala et al. [45] scheme is vulnerable to sensor capture attack and does not guaranteed forward secrecy. However, the D. Rangwani et al. [41] and A.Vangala et al. [45] schemes have not considered the multi-gateway environment. Likewise, the work of Dhillon and Kalra [38] is vulnerable to insider attacks, user identity guessing attacks, session key attacks, sensor capture attacks, and offline guessing attacks. Furthermore, Dhillon and Kalra [38] did not consider security

features such as forward secrecy, untraceability, and multi-gateway supports. Furthermore, work of J. Lee et al. [28] is vulnerable to insider attacks, gateway impersonation attacks, DoS attacks, and sensor capture attacks.

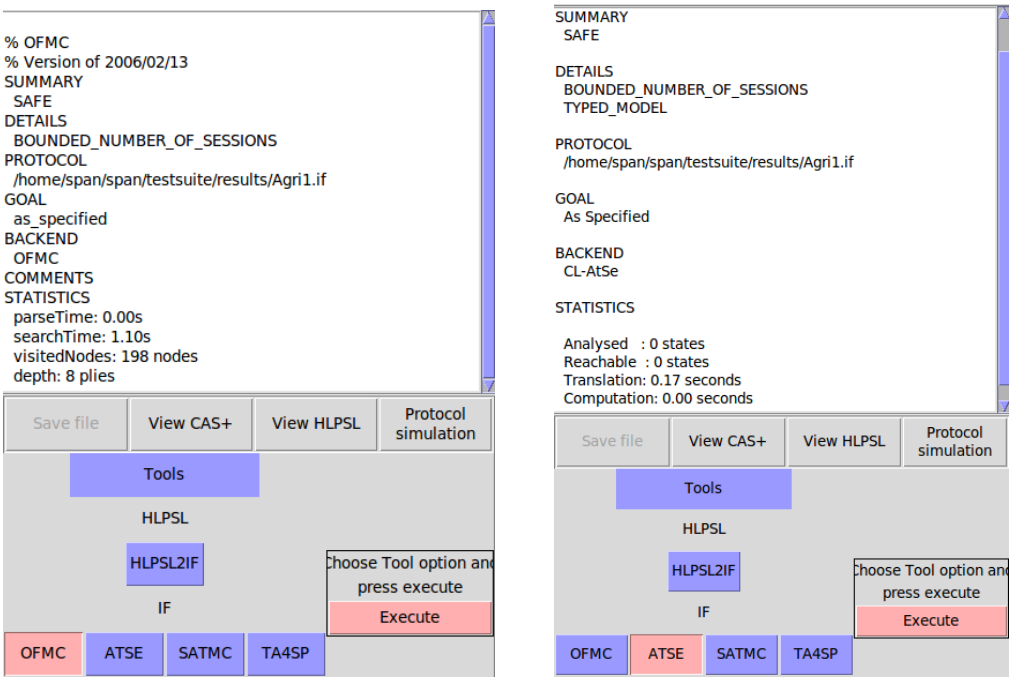

Figure 17. The simulation results using OFMC and CL-AtS back-ends in Case 2.

Table 3. Comparison of security features.

	Rangwani et al. [41]	Vangala et al. [45]	Dhillon et al. [38]	Lee et al. [28]	Proposed Scheme
Insider attack	✓	✓	×	×	✓
Agriculture professional identity-guessing attack	×	×	×	×	✓
Gateway impersonation attack	×	×	✓	×	✓
IoT smart device impersonation attack	×	×	×	✓	✓
Agriculture professional impersonation attack	✓	✓	×	✓	✓
Denial of service attack	✓	✓	✓	×	✓
Session Key attack	✓	×	×	✓	✓
Offline guessing attack	✓	✓	×	✓	✓
Replay attack	✓	✓	✓	✓	✓
Man-in-the-middle attack	✓	✓	✓	✓	✓
Smart card stolen attack	✓	✓	✓	✓	✓
Sensor Capture attack	✓	×	×	×	✓
Untraceability	✓	✓	×	✓	✓
Anonymity	✓	✓	✓	✓	✓
Forward secrecy	✓	×	×	×	✓
Mutual Authentication	✓	✓	✓	✓	✓
Multi-gateway supports	×	×	×	✓	✓

- Insider attack: The adversary gets the user's lost/stolen card and obtains the information $(HGW_ID, A^*_{th}), (FGW_{ID}, A^*_{if}), TID_i, Gen(.), Rep(.), h(.), t, \tau, T_i, S_i$ that is stored in the smart card. Even if the SA is trusted, information can be obtained PID_i and PWR_i by a malicious insider. Nevertheless, if the value $T_i = h(U_ID)R$ is calculated with 1024-bit large secret number R; the attacker needs R to guess the user information U_{ID}, and U_{PW}, which only the user Ui knows about it. Additionally, the attacker must know the biometric key data, if he/she wants to derive R, which is computationally infeasible to guess when compared to low-entropy passwords. Since the attacker cannot correctly guess U_{ID}, and U_{PW},, therefore, the proposed scheme is secure against insider attacks.

- Agriculture professional identity-guessing attack: As mentioned above, the SA knows the user information U_{ID}, and U_{PW} during the registration phase and in case of the adversary with malicious insider attack, the SA knows about it while sending requests for registration. To obtain the identity of the user U_{ID} from $PID_i = h(U_{ID} \parallel R)$, the attacker is required to know R. Furthermore, if the attackers intercept the messages $M_1 = [SN_{ID}, TID_i, CT_i]$ in the login phase,$M_2 = SN_{ID}, CT_j, M_3 = SN_{ID}, B_i, Z_i, TS_3$, during Case 1, and $M_5 = SN_{ID}, CT_f, M_6 = SN_{ID}, F_i, Q_i, TS_5, M_7 = SN_{ID}, CT_{if}$ and $M_8 = SN_{ID}, D_i, J_i, TS_7$ during the authentication procedure of Case 2. The attacker cannot correctly infer the user's identity since the TID_i is safeguarded using a one-way hash function. As a result, the proposed scheme is resistant to identity-guessing attacks.
- Gateway impersonation attack: If the adversary attempts to drop the message $M_2 = SN_{ID}, CT_j$ from the public channel during Case 1, where $P_i = h(PID_i \parallel N_j \parallel N_i^* \parallel TS_2)$, and $CT_j = E_{SK_j}[HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2]$, and tries to calculate the message $M_2 = SN_{ID}, CT_j$ to send to the SNi. If the sensor accepts the message, the attacker will impersonate the news as a legitimate gateway. However, this is not possible in our proposed scheme since the letter is attached with a fresh timestamp TS_2 and cannot pass the verification even if the adversary successfully generates a nonce N_j. Further, the attacker needs to compute the CT_j through the use of the secret key to encrypt additional parameters $SK_j = h(HGW_{ID} \parallel SN_{ID} \parallel HGW_{MSK}) = h(SN_{ID} \parallel SN_{MSK})$ that are shared between gateway and sensor. The SK_j is unknown to the attacker with N_j, and PID_i to compute P_i. As a result, even if the attacker successfully captures a sensor, he/she will be unable to impersonate a valid HGWN. As a result, the proposed scheme is resistant to a gateway impersonation attack.
- IoT smart device impersonation attack: The adversary must construct a valid message to impersonate the sensor node SN and deceive the HGWN, say $M_3 = SN_{ID}, B_i, Z_i, TS_3$ throughout the authentication phase, and make additional efforts to create a message M_3' via the public channel. The attacker needs PID_i, and N_j. As a result, the adversary cannot pose as a valid sensor node SN in the proposed system, preventing sensor node impersonation attacks.
- Agriculture professional impersonation attack: To impersonate the user U_i as a valid user, assume that the adversary eavesdrops on the message $M_1 = [SN_{ID}, TID_i, CT_i]$, where $W_i = h(PID_i^* \parallel TID_i \parallel N_i), CT_i = E_{SK_i}[HGW_{ID}, SN_{ID}, W_i, N_i, TS_1]$, and $PID_i^* = h(U_{ID} \parallel R^*)$. Assume the attacker attempts to construct another valid log-in request message, compelling the adversary to authenticate to the HGWN. To accomplish this, the adversary must know PID_i^*, which is impossible without the secret R^*. Assume the adversary gets the N_i, and TS_1, but cannot generate $CT_i = E_{SK_i}[HGW_{ID}, SN_{ID}, W_i, N_i, TS_1]$ because he/she does not have access to the shared User/HGWN Secret Key SK_i. As a result, a user impersonation attack can be used against the proposed scheme.
- Denial of service attack: Assume the attacker has the lost/stolen smart card of the user U_i; he/she cannot have the user information username U_{ID}, password U_{PW} and imprints of their biometric U_{BIO}. Furthermore, the smart card compute $\sigma_i^* = Rep(U_{BIO}, \tau)$ using the error tolerance thresholds value $\tau, R^* = T_i \oplus h(U_{ID} \parallel \sigma_i^*)$, $PID_i^* = h(U_{ID} \parallel R^*), PWR_i^* = h(U_{PW} \parallel R^*)$, and $R_i^* = h(PID_i^* \parallel PWR_i^* \parallel \sigma_i^*)$. After that, the smart card checks the validity of $R_i^* \neq R_i$. Therefore, without having valid user information, the validation will fail. Similarly, the adversary cannot update the smart card's stored secret credentials without access to user information. As a result, the proposed scheme protects against denial of service attacks.
- Session Key attack: The shared session key is established during the authentication step by the user Ui and the sensor node $SK_{U \to SN} = h(HGW_{ID} \parallel SN_{ID} \parallel PID_i \parallel N_j)$, which includes $PID_i = h(U_{ID} \parallel R)$, and random nonce N_j. In both cases, these parameters are protected using a one-way hash function, which means that an attacker

cannot obtain the session key without knowing the secret parameters of the session key. Therefore, the session key attack is resisted in the proposed scheme.

- Offline guessing attack: Assume that the user password U_{PW} is guessed by the adversary, he/she will not be able to generate a valid authentication request $CT_j = E_{SK_j}[HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2]$, where $P_i = h(PID_i \parallel N_j \parallel N_i^* \parallel TS_2)$. Because the adversary does not have the PID_i, and N_j and cannot forge the user biometric U_{BIO}. Even if the adversary generates N_j, they still will not be able to compute CT_j, because he/she does not know the secret key SK_j. Therefore, the proposed scheme is resilient against offline guessing attacks.
- Replay attack: Assume that the adversary intercepts the messages $M_1 = [SN_{ID}, TID_i, CT_i]$, $M_2 = SN_{ID}, CT_j$, $M_3 = SN_{ID}, B_i, Z_i, TS_3$ in both cases during authentication. The adversary will be unable to replay the message, as each message contains timestamps and a random nonce, both of which are verified by the recipient before any message processing. Thus, the receiver can determine an older message by comparing the timestamp to the timestamp of the current system. As a result, the proposed scheme prevents replay attacks.
- Man-in-the-middle attack: Assume that the adversary intercepts the messages $M_1 = [SN_{ID}, TID_i, CT_i]$, $M_2 = SN_{ID}, CT_j$, $M_3 = SN_{ID}, B_i, Z_i, TS_3$, and tries to tamper with the content before passing it to the receiver so that the receiver will not be aware of the modified messages. In the proposed scheme, the messages are encrypted, say $CT_j = E_{SK_j}[HGW_{ID}, SN_{ID}, PID_i, N_i^*, N_j, P_i, TS_2]$, which involves random nonce, timestamp, and PID_i. The receiver checks the condition of the timestamp and random nonce before any processing of the received message. Furthermore, the parameters are encrypted using the shared key SK_j, which is computationally infeasible for the attacker to generate and obtain the parameters. If the attacker generates the secret key, he/she does not know $PID_i^* = h(U_{ID} \parallel R^*)$ because it is protected using a one-way hash function and involves a secret value R^*. Therefore, the proposed scheme withstands a man-in-the-middle attack.
- Smart card stolen attack: Assume that an attacker steals the user's smart-card SC and extracts the value TID_i, σ, T_i and S_i. The attacker will not be able to compute $T_i = h(U_{ID} \parallel \sigma) \oplus R, S_i = h(PID_i \parallel PWR_i \parallel \sigma)$ since they are computed using the biometric key data. Furthermore, the adversary cannot compute $PID_i = h(U_{ID} \parallel R)$ because it is protected using a one-way hash function. Thus, without knowing the user information, the adversary cannot generate the login message. Therefore. the proposed scheme protects against smart card stolen attacks.
- Sensor Capture attack: In a harsh environment, the attackers quickly capture the sensor nodes. If the attacker captures the node SN, he/she will extract the secret information (SN_{ID}, A_j), where $A_j = h(SN_{ID} \parallel SN_{MSK})$ is computed using the SN_{MSK}, which is a secret value not known to other participants. Therefore, identifying the sensor secured with the one-hash function cannot negatively affect the sensor node nor can it disrupt the authentication process between the agricultural professional and the sensor node. Therefore, the proposed scheme protects against sensor capture attacks.
- Agriculture professional/sensor node untraceability: Assume that the attacker eavesdrops on the authentication messages from different sessions and checks whether the messages are the same. If they are the same, both messages are sent by identical identities, e.g., agriculture professional or sensor node. However, despite recording the authentication message and stealing $M_1 = [SN_{ID}, TID_i, CT_i]$, $M_2 = SN_{ID}, CT_j$, $M_3 = SN_{ID}, B_i, Z_i, TS_3$, the adversary cannot trace the agriculture professional or the sensor node because these messages are comprised of the random nonces N_i^*, N_j, and timestamps TS_1, TS_4, which are generated freshly in every session separately, leading to a new formation of the messages. Therefore, the user anonymity and sensor node cannot be traced.
- User anonymity: The adversary in this attack tries to obtain the user information when the messages are transmitted via a public channel in their original form. The user sends

the messages, say $M_1 = [SN_{ID}, TID_i, CT_i]$, $M_2 = SN_{ID}, CT_j$, $M_3 = SN_{ID}, B_i, Z_i, TS_3$ to the gateway, and the transmitted messages do not contain any identity of the agriculture professionals. Additionally, the messages are sent in encrypted form where $CT_i = E_{SK_i}[HGW_{ID}, SN_{ID}, W_i, N_i, TS_1]$ using the freshly generated shared secret key. The messages are further formed using an irreversible hash operation. Thus, each message that comes from the same user is different from one session to another. Therefore, the scheme guarantees user anonymity.

- Forward secrecy: In the proposed scheme, the long-term key SK_i is disclosed to the user only, and the session key is also kept securely. The secret key is computed $SK_i = A^*_{th} \oplus h(PID^*_i \parallel PWR^*_i \parallel TID_i) \oplus h(\sigma^*_i \parallel R^*_i) = h(HGW_{ID} \parallel PID_i \parallel HGW_{MSK})$, and it needs PID_i, R^*_i, and σ^*_i only known to the user. If the adversary somehow reveals the secret key of both user and gateway, he/she also needs to know PID_i, which is protected using a one-way hash function, and the random number R^*_i. The complexity of guessing the secret key and the random number chosen by the user or sensor node in polynomial time using any powerful computer is amazingly massive and almost impossible. As a result, the proposed scheme preserves forward secrecy.
- Mutual authentication: The proposed scheme provides mutual authentication because the agriculture professional sends the login message, say $M_1 = [SN_{ID}, TID_i, CT_i]$, to the HGWN via a public channel. Upon receiving message, the HGWN verifies the TS_1 by selecting a new timestamp TS_2 to check the freshness $|TS_2 - TS_1| \leq \triangle T$, where $\triangle T$is allowed transmission delay. Furthermore, it decrypts $CT_i = D_{SK_i}[HGW^*_{ID}, SN^*_{ID}, W^*_i, N^*_i, TS^*_1]$, where D_{SK_i} depicts the decryption of a symmetric key using the key SK_i. After retrieving the information, HGWN verifies the timestamp $|TS^*_1 - TS^{'}_1| \leq \triangle T$, where $TS^{'}_1$ is the message receiving time. If it holds, it checks $HGW^*_{ID} \neq HGW_{ID}$, and $SN^*_{ID} \neq SN_{ID}$, and if these parameters are valid, it computes $W^{**}_i = h(PID_i \parallel TID_i \parallel N^*_i)$ based on the stored PID_i, and TID_i, then checks $W^{**}_i \neq W^*_i$. If it does not hold, it terminates the session. The verification will fail here since the validation depends on the one-way hash function. Therefore, mutual authentication is provided in the proposed scheme among all the participants.
- Multi-gateway supports: In the proposed scheme, multi-gateways (e.g., HGWN and FGWN) are registered with SA to enable agriculture professionals to authenticate to a sensor node with other fields. When the HGWN receives the login message, it checks if the HGWN database contains SN_{ID}, and performs HGWN authentication (Case 1); otherwise, it performs FGWN authentication (Case 2). Therefore, the proposed scheme supports multi-gateway authentication.

3.3. Computation Cost

In Table 4, this subsection compares the proposed scheme in terms of computation cost in the login and authentication phases with other related schemes, e.g., D. Rangwani et al. [41], Dhillon and Kalra [38], J. Lee et al. [28], and A.Vangala et al. [45]. In WSN, the sensor node has the most limited resources such as the user's smart card, sensor node, and GWN (base station). We used the hardware platform primarily on the previous studies to calculate execution time, including an Intel dual-core processor with a clock speed of 2.20 GHz, Ubuntu 12.04.1 LTS 32-bit Operating System, and 2 GB memory. The approximate execution time for various cryptographic operations by using the cryptography PBC library (version 0.5.12) is based on the GMP Library (version 5.0.5) reported in [41]. The execution time in ms is required for each primitive operation as noted in Table 4. During the login and authentication phase of the A.Vangala et al. [45] scheme, a user requires $13T_h + 4T_{ecm} + T_{eca} + T_{fe} = 15.473$ ms, an IoT smart device (sensor node) requires $9T_h + 4T_{ecm} + T_{eca} = 11.949$ ms, and a gateway node requires $12T_h + 6T_{ecm} + 2T_{eca} = 4.708$ ms. In the Dhillon and Kalra [38] scheme, the user needs to perform $10T_h + 1T_{E/D}$, and the sensor applies $6T_h + 2T_{(E/D)}$. In the gateway side, $7T_h$, and $2T_{E/D}$ are needed, so, the total computation cost in Dhillon and Kalra [38] is 20.382 ms.

In the D. Rangwani et al. [41] scheme, the user requires $4T_H + 2T_{ecm}$, and the gateway nodes require $7T_H + 1T_{ecm}$. Likewise, in the sensor side, the computation cost is $4T_H + 1T_{ecm}$, thus, the total computation cost in the D. Rangwani et al. [41] scheme can be represented as $15T_H + 4T_{ecm}$ and the estimation time is 8.9730 ms. In the J. Lee et al. [28] scheme, the user needs to perform hash function operation $14T_H$ and one-time fuzzy extractor operation $1T_{fe}$. At the sensor side, nine times hash operations $9T_H$ are required. Likewise, five times hash operations $5T_H$ are needed at the gateway side. Therefore, the total computation cost of the J. Lee et al. [28] scheme is $28T_H + 1T_{fe} \equiv 27.2743ms$. The proposed scheme, on the other hand, has a total computational cost of $17T_H + 1T_{fe} + 4T_{E/D}$ for Case 1, and $19T_H + 1T_{fe} + 5T_{E/D}$ for Case 2. The user is not required to employ symmetric encryption/decryption (e.g., AES) because of the computational efficiency of both the fuzzy extractor operation and symmetric encryption/decryption (e.g., AES). The computational cost of a resource-constrained sensor node is $3T_H + 1T_{E/D}$ in Case 1, whereas it is $4T_H + 1T_{E/D}$ in Case 2. Both the hash function and symmetric encryption/decryption are very efficient, which makes the proposed scheme very efficient for resource-constrained sensor nodes in WSNs. Table 4 compares the computation costs required in different schemes during the "login and authentication phase. For instance, it shows that the A.Vangala et al. [45], Dhillon and Kalra [38], and D. Rangwani et al. [41] require the computation costs 32.13ms, 32.13ms, and 20.4885ms, respectively. According to the comparison, the proposed scheme has less computation cost when compared to the existing schemes. Furthermore, due to the utilization of the fuzzy extractor technique, it provides "superior security and more functionality features when compared with all authentication schemes".

Table 4. Comparison of the computation and communication costs of schemes.

Scheme	Computation Cost (ms)				Communication Cost
	User	Sensor	Gateway	Total	
A.Vangala et al. [45]	$13T_h + 4T_{ecm} + T_{eca} + T_{fe}$	$9T_h + 4T_{ecm} + T_{eca}$	$12T_h + 6T_{ecm} + 2T_{eca}$	32.13 ms	5792 bits
Dhillon and Kalra [38]	$10T_h + 1T_{E/D}$	$6T_h + 2T_{E/D}$	$7T_h + 2T_{E/D}$	20.382 ms	4016 bits
D. Rangwani et al. [41]	$4T_H + 2T_{ecm}$	$4T_H + 1T_{ecm} + 1T_{E/D}$	$7T_H + 1T_{ecm} + 2T_{E/D}$	20.4885 ms	2752 bits
J. Lee et al. [28]	$14T_H + 1T_{fe}$	$9T_H$	$5T_H$	27.2743 ms	2400 bits
Proposed scheme	$9T_H + 1T_{fe} + 1T_{E/D}$	$3T_H + 1T_{E/D}$	$5T_H + 2T_{E/D}$	17.6651 ms	1696 bits

3.4. Communication Cost

The communication cost comparison of the proposed scheme and other existing schemes regarding the total number of bits required for that transmitted message is illustrated in Table 4. For computing the communication cost, we assume that the identities $HGW_{ID}, FGW_{ID}, PID_i$, and U_{ID} are all 160 bits in length; the hash output is 160 bits long; and the identities SN ID, random number/nonce, and timestamp are all 32 bits. For a 128-bit plaintext block, symmetric encryption/decryption (using AES-128) requires 128 bits. In the A.Vangala et al. [45] scheme, the messages $Msg_1 = <TID_U, M_1, Y_1, Sign_{y1}, T_1>$ with $|TID_U| = 160$ bits, $|M_1| = 320$ bits, $|Y_1| = 320$ bits, $|Sign_{y1}| = 256$ bits and $|T_1| = 32$ bits, requires 1088 bits; $Msg_2 = <TID_C, M_2, Y_1, Y_2, Sign_{y2}, T_2>$ with $|TIDC| = 160$ bits, $|M_2| = 256$ bits, $|Y_1| = 320$ bits, $|Y_2| = 320$ bits, $|Sign_{y2}| = 256$ bits and $|T_2| = 32$ bits, needs 1344 bits; $Msg_3 = <TID^*_S, M_3, M_4, Y_3, Sign_{y3}, T_3>$ with $|TID^*_S| = 256$ bits, $|M_3| = 256$ bits, $|M_4| = 256$ bits, $|Y_3| = 320$ bits, $|Sign_{y3}| = 256$ bits and $|T_3| = 32$ bits, demands 1376 bits; and $Msg4 = <TID^*_U, M_3, M_4, M_5, Y_3, Y_4, Sign_{y4}, T_3, T_4>$ with $|TID^*_U| = 256$ bits, $|M_3| = 256$ bits, $|M_4| = 256$ bits, $|M_5| = 256$ bits, $|Y_3| = 320$ bits, $|Y_4| = 320$ bits, $|Sign_{y4}| = 256$ bits, $|T_3| = 32$ bits and $|T_4| = 32$ bits, requires 1984 bits. The cumulative communication cost in this scheme amounts to 5792 bits. Furthermore, the Dhillon and Kalra [38] scheme has four messages, which are transmitted among the participant's entities, costing 4016 bits. In the D. Rangwani et al. [41] scheme, the user transmitting

the message $M_1 = h(storedDG||R_{U1}||T1||K)$ that requires 256 bits, and receives $M_6 = h(receivedID_N||ID_G||storedDG||T_4), M_7 = h(M_6||RG_1||receivedRN||K)$ with 352 bits size length. Likewise, the gateway transmits the message $M_1^{'} = h(storedDG||receivedRU_1||receivedT_1||K)$ and receives $M_3 = h(M_2||RG_1||receivedRU_1||L)$ with a size of 1376 bits. The sensor node transmits and receives the messages $M_4 = h(ID_N||receivedRG_1||storedAN||T_3)$ and $M_5 = h(M_4||RN||L)$ that cost approximately 832 bits. Therefore, the total communication cost of D. Rangwani et al. [41] is 2752 bits. In J. Lee et al. [28], four messages are exchanged between entities, the user transmits the login request message $\{HID_i, C_i, VU_i\}$ and receives an authentication message $\{H_i, Mc_u\}$ from GWNs. While the gateway sends the message $\{HID_i, C_i, PID_j, D_i, VS_j\}$ to control server and receives $\{Mc_g, Mc_u, E_i, F_i\}$ from the GWN. Therefore, the total communication cost of J. Lee et al. [28] is 2400 bits.

The communication cost required in the proposed scheme, on the other hand, is 1696 bits (for case 1) and 3168 bits (for instance 2) when the FGWN is involved in the authentication phase. In our scheme, during the log-in phase, the log-in request message, $M_1 = [SN_{ID}, TID_i, CT_i]$ requires 704 bits, where 32 and 160 bits are required for SN_{ID} and TID_i,, respectively, and (160 + 32 + 160 + 32 + 32)/128e = 512 bits for $CT_i = E_{SK_i}[HGW_{ID}, SN_{ID}, W_i, N_i, TS_1]$. For the authentication and key agreement phase, consider case 1, the messages of $M_2 = SN_{ID}, CT_j$ and $M_3 = SN_{ID}, B_i, Z_i, TS_3$ 640 bits and 384 bits are required, respectively. As a result, the proposed scheme's communication cost for case 1 becomes (704 + 640 + 384) = 1696 bits. Similarly, the communication cost for case 2 in our scheme is 3168 bits due to the messages' involvement $M_1[SN_{ID}, TID_i, CT_i], M_5 = SN_{ID}, CT_f, M_6 = SN_{ID}, F_i, Q_i, TS_5, M_7 = SN_{ID}, CT_{if}, and M_8 = SN_{ID}, D_i, J_i, TS_7$. Table 4 compares the costs of communication in terms of the number of bits required for message exchange; it illustrates that D. Rangwani et al. [41], Dhillon and Kalra [38], J. Lee et al. [28], and A.Vangala et al. [45] require 5792 bits, 4016 bits, 2400 bits and 2752 bits, independently. This comparison proves that the proposed scheme has less communication cost, which required 1696 bits.

4. Conclusions

As security breaches become more prevalent, new authentication techniques must incorporate agriculture professionals' biometrics to improve the system's security. To cater for this need, a robust multi-gateway authentication scheme for agriculture WSN is proposed in this paper. The proposed scheme exploits the advantages of the fuzzy extractor to design a secure authentication system. The study proposed a multi-gateway model to overcome single point failure that exists in a single gateway communication model. This paper pointed out that multi-gateway WSN can allow users to access data from multiple sensor areas (a typical IoT deployment). Furthermore, the study added a new joined phase to enable new sensors to join the agriculture field. The proposed scheme is resistant to various known attacks, including the sensor node capture attack, as proved by formal and informal security research. The proposed scheme is secure against replay and man-in-the-middle attacks, as demonstrated by the extensive formal security verification performed using the AVISPA tool. Furthermore, we demonstrated that our scheme is efficient and provides additional functionality as compared to previous schemes through performance results. In terms of performance, the proposed scheme is better suitable for IoT deployment, where devices deployed in agriculture are generally resource constrained. The future works of this research can be summarized as follows: First, this paper provides secure communication for a multi-gateway environment with efficient results. This solution can be extended to enable the environment with different communication methods used in the same area (e.g., Bluetooth, ZigBee, and WiFi) to guard against interference. Second, due to the efficiency of the proposed scheme, it can further be extended to provide secure user authentication to monitor the field progress. Third, we plan to extend the proposed scheme to protect against distributed denial of service attacks (DDoS) that mainly target sensor nodes.

Author Contributions: Conceptualization, H.K. and S.J.H.; methodology, H.K., S.J.H.; software, H.K.; validation, H.K.; results interception, H.K. and S.J.H.; formal analysis, H.K.; writing—original draft preparation, H.K.; writing—review and editing, H.K. and S.J.H.; supervision, S.J.H., S.M.S.A., F.H. and M.A.C.; and project administration, S.J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: We would like to thank the reviewers for their careful, constructive and insightful comments in relation to this work.

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

References

1. (CAO), C.O. Society 5.0: What is Society 5.0? 2021. Available online: https://www8.cao.go.jp/cstp/english/society5_0/index.html (accessed on 15 July 2021).
2. Team, S.P.H.S.I. Realization of Society 5.0 by utilizing precision agriculture into smart agriculture in NARO, Japan. In *International Workshop on Icts For Precision Agriculture*; National Agricultural Research Organisation: Tokyo, Japan, 2019; p. 58.
3. Yu, X.; Wu, P.; Han, W.; Zhang, Z. A survey on wireless sensor network infrastructure for agriculture. *Comput. Stand. Interfaces* **2013**, *35*, 59–64. [CrossRef]
4. Olariu, S. Smart Communities: From Sensors to Internet of Things and to a Marketplace of Services. In Proceedings of the 9th International Conference on Sensor Networks—SENSORNETS, Valletta, Malta, 28–29 February 2020; pp. 7–18.
5. Yang, X.; Shu, L.; Chen, J.; Ferrag, M.A.; Wu, J.; Nurellari, E.; Huang, K. A survey on smart agriculture: Development modes, technologies, and security and privacy challenges. *IEEE CAA J. Autom. Sin.* **2020**, *8*, 273–302. [CrossRef]
6. Iqbal, A.; Olariu, S. A survey of enabling technologies for smart communities. *Smart Cities* **2021**, *4*, 54–77. [CrossRef]
7. Pujari, M.S.; Bogiri, M.N. A survey on wireless sensor network for agriculture. *Int. J. Recent Innov. Trends Comput. Commun.* **2017**, *5*, 269–272.
8. Ferrag, M.A.; Shu, L.; Yang, X.; Derhab, A.; Maglaras, L. Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges. *IEEE Access* **2020**, *8*, 32031–32053. [CrossRef]
9. Klein, L.J.; Hamann, H.F.; Hinds, N.; Guha, S.; Sanchez, L.; Sams, B.; Dokoozlian, N. Closed loop controlled precision irrigation sensor network. *IEEE Internet Things J.* **2018**, *5*, 4580–4588. [CrossRef]
10. Diedrichs, A.L.; Bromberg, F.; Dujovne, D.; Brun-Laguna, K.; Watteyne, T. Prediction of frost events using machine learning and IoT sensing devices. *IEEE Internet Things J.* **2018**, *5*, 4589–4597. [CrossRef]
11. Chen, W.L.; Lin, Y.B.; Lin, Y.W.; Chen, R.; Liao, J.K.; Ng, F.L.; Chan, Y.Y.; Liu, Y.C.; Wang, C.C.; Chiu, C.H.; others. AgriTalk: IoT for precision soil farming of turmeric cultivation. *IEEE Internet Things J.* **2019**, *6*, 5209–5223. [CrossRef]
12. Mukherjee, A.; Misra, S.; Raghuwanshi, N.S.; Mitra, S. Blind entity identification for agricultural IoT deployments. *IEEE Internet Things J.* **2018**, *6*, 3156–3163. [CrossRef]
13. Zamora-Izquierdo, M.A.; Santa, J.; Martínez, J.A.; Martínez, V.; Skarmeta, A.F. Smart farming IoT platform based on edge and cloud computing. *Biosyst. Eng.* **2019**, *177*, 4–17. [CrossRef]
14. Abouzar, P.; Michelson, D.G.; Hamdi, M. RSSI-based distributed self-localization for wireless sensor networks used in precision agriculture. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 6638–6650. [CrossRef]
15. Vuran, M.C.; Akyildiz, I.F. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 226–230.
16. Silva, A.R.; Vuran, M.C. Communication with aboveground devices in wireless underground sensor networks: An empirical study. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–6.
17. Ma, J.; Zhou, X.; Li, S.; Li, Z. Connecting agriculture to the internet of things through sensor networks. In Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Dalian, China, 19–22 October 2011; pp. 184–187.
18. Gutiérrez, J.; Villa-Medina, J.F.; Nieto-Garibay, A.; Porta-Gándara, M.Á. Automated irrigation system using a wireless sensor network and GPRS module. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 166–176. [CrossRef]
19. Nikiforova, A. Smarter Open Government Data for Society 5.0: Are Your Open Data Smart Enough? *Sensors* **2021**, *21*, 5204. [CrossRef] [PubMed]
20. Grgić, K.; Žagar, D.; Balen, J.; Vlaović, J. Internet of Things in Smart Agriculture—Possibilities and Challenges. In Proceedings of the 2020 International Conference on Smart Systems and Technologies (SST), Osijek, Croatia, 14–16 October 2020 ; pp. 239–244.

21. Pavithra, L.; Abdullah, M.; Prakash, S.; Karthick, S.; Ragavi, B.; Nandhini, V. Wireless Sensor Networks: A Review on Sensor Deployment and Routing Protocols for Different Application. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Sathyamangalam, India, 11–12 December 2021; Volume 1084, p. 012052.
22. Khalid, H.; Hashim, S.J.; Ahmad, S.M.S.; Hashim, F.; Chaudhary, M.A. Security and Safety of Industrial Cyber-Physical System: Systematic Literature Review. *PalArch's J. Archaeol. Egypt Egyptol.* **2020**, *17*, 1592–1620.
23. Khalid, H.; Hashim, S.J.; Ahmad, S.; Hashim, F.; Chaudary, M.A. Cybersecurity in Industry 4.0 context: Background, issues, and future directions. In *The Nine Pillars of Technologies for Industry 4.0*; 2020; pp. 263–307.
24. Prodanović, R.; Rančić, D.; Vulić, I.; Zorić, N.; Bogićević, D.; Ostojić, G.; Sarang, S.; Stankovski, S. Wireless sensor network in agriculture: Model of cyber security. *Sensors* **2020**, *20*, 6747. [CrossRef] [PubMed]
25. Saini, R.K.; Prakash, C. Internet of Things (IoT) for Agriculture growth using Wireless Sensor Networks. *Glob. J. Comput. Sci. Technol.* **2020**, 20, 4584.
26. Jawad, H.M.; Nordin, R.; Gharghan, S.K.; Jawad, A.M.; Ismail, M. Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors* **2017**, *17*, 1781. [CrossRef]
27. Xu, L.; Wu, F. A lightweight authentication scheme for multi-gateway wireless sensor networks under IoT conception. *Arab. J. Sci. Eng.* **2019**, *44*, 3977–3993. [CrossRef]
28. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
29. Yuan, J.J. An enhanced two-factor user authentication in wireless sensor networks. *Telecommun. Syst.* **2014**, *55*, 105–113. [CrossRef]
30. Turkanović, M.; Brumen, B.; Ḧolbl, 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]
31. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Inf. Sci.* **2015**, *321*, 263–277. [CrossRef]
32. He, D.; Zhang, Y.; Chen, J. Robust Biometric-Based User Authentication Scheme for Wireless Sensor Networks. *IACR Cryptol. EPrint Arch.* **2012**, *2012*, 203.
33. Chen, M.; Lee, T.F.; Pan, J.I. An enhanced lightweight dynamic pseudonym identity based authentication and key agreement scheme using wireless sensor networks for agriculture monitoring. *Sensors* **2019**, *19*, 1146. [CrossRef]
34. Wu, H.T.; Tsai, C.W. An intelligent agriculture network security system based on private blockchains. *J. Commun. Netw.* **2019**, *21*, 503–508. [CrossRef]
35. Wu, F.; Xu, L.; Kumari, S.; Li, X. A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security. *J. Ambient Intell. Humaniz. Comput.* **2017**, *8*, 101–116. [CrossRef]
36. 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]
37. Sadhukhan, D.; Ray, S.; Biswas, G.; Khan, M.K.; Dasgupta, M. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. *J. Supercomput.* **2021**, *77*, 1114–1151. [CrossRef]
38. 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]
39. 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]
40. Ojha, T.; Misra, S.; Raghuwanshi, N.S. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* **2015**, *118*, 66–84. [CrossRef]
41. 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.
42. 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]
43. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [CrossRef]
44. Haseeb, K.; Ud Din, I.; Almogren, A.; Islam, N. An energy efficient and secure IoT-based WSN framework: An application to smart agriculture. *Sensors* **2020**, *20*, 2081. [CrossRef]
45. 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. Pract. Exp.* **2021**, e6187. [CrossRef]
46. Almadani, B.; Mostafa, S.M. IIoT based multimodal communication model for agriculture and agro-industries. *IEEE Access* **2021**, *9*, 10070–10088. [CrossRef]
47. Saiz-Rubio, V.; Rovira-Más, F. From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy* **2020**, *10*, 207. [CrossRef]
48. Wang, C.; Wang, D.; Tu, Y.; Xu, G.; Wang, H. Understanding node capture attacks in user authentication schemes for wireless sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2020**. [CrossRef]
49. Guo, H.; Gao, Y.; Xu, T.; Zhang, X.; Ye, J. A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks. *Ad Hoc Netw.* **2019**, *95*, 101965. [CrossRef]

50. Syverson, P.; Cervesato, I. The logic of authentication protocols. In *International School on Foundations of Security Analysis and Design*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 63–137.
51. Syverson, P.F.; Van Oorschot, P.C. *A Unified Cryptographic Protocol Logic*; Technical Report; Naval Research Lab.: Washington, DC, USA, 1996.
52. Khalid, H.; Hashim, S.J.; Syed Ahmad, S.M.; Hashim, F.; Chaudhary, M.A. Cross-SN: A Lightweight Authentication Scheme for a Multi-Server Platform Using IoT-Based Wireless Medical Sensor Network. *Electronics* **2021**, *10*, 790. [CrossRef]
53. Khalid, H.; Hashim, S.J.; Ahmad, S.M.S.; Hashim, F.; Chaudhary, M.A. SELAMAT: A New Secure and Lightweight Multi-Factor Authentication Scheme for Cross-Platform Industrial IoT Systems. *Sensors* **2021**, *21*, 1428. [CrossRef] [PubMed]