Article

Secure Authentication Scheme Using Diffie–Hellman Key Agreement for Smart IoT Irrigation Systems

Shadi Nashwan

Computer Science Department, College of Computer and Information Sciences, Jouf University, Sakaka 42421, Saudi Arabia; shadi_nashwan@ju.edu.sa; Tel.: +966-56-450-6868

Abstract: Smart irrigation is considered one of the most significant agriculture management systems worldwide, considering the current context of water scarcity. There is a clear consensus that such smart systems will play an essential role in achieving the economic growth of other vital sectors. In general, the consequences of global warming and the unavailability of clean water sources for the agricultural sector are clear indications that the demand for these systems will increase in the near future, especially considering the recent expansions in the use of the Internet of Things (IoT) and Wireless Sensor Network (WSN) technologies, which have been employed in the development of such systems. An obvious result is that security challenges will be one of the main obstacles to attaining the widespread adoption of such systems. Therefore, this paper proposes a secure authentication scheme using Diffie–Hellman key agreement for smart IoT irrigation systems using WSNs. This scheme is based on Diffie–Hellman and one-way hash cryptographic functions in order to support the basic security services with a high data rate and ability to resist well-known attacks. The Burrows–Abadi–Needham (BAN) logic model is used to verify the proposed scheme formally. Based on various possible attack scenarios, a resistance analysis of the proposed scheme is discussed. Further analyses are performed in terms of the storage size, intercommunication, and running time costs. Therefore, the proposed scheme not only can be considered a secure authentication scheme but is also practical for smart IoT irrigation systems due to its reasonable efficiency factors.

Keywords: smart irrigation system; mutual authentication; anonymity; perfect forward secrecy; BAN logic model

Citation: Nashwan, S. Secure Authentication Scheme Using Diffie–Hellman Key Agreement for Smart IoT Irrigation Systems. *Electronics* **2022**, *11*, 188. <https://doi.org/10.3390/electronics11020188>

Academic Editors: Khaled Elleithy and Dimitra I. Kaklamani

Received: 4 December 2021

Accepted: 6 January 2022

Published: 8 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Increased demand for clean water in various vital sectors and the risks of climate change in particular geographic areas are considered the main factors that are directly affecting water availability [1]. Particularly, the consumption of fresh water in the agricultural sector is a rising concern, especially in poor and developing countries with reduced water quality and increased salinity of water [1,2]. These reasons have led to the usage of smart irrigation systems, such as smart fogging and fertigation systems [3].

The main benefit of smart internet of things (IoT) based irrigation systems consists of the remote monitoring of water usage in order to reduce water waste according to the actual needs of various crops. This applies regardless of whether the irrigation system depends on flood irrigation, spray irrigation, drip irrigation, or nebulizer irrigation. Moreover, smart IoT irrigation systems can improve agricultural productivity, provide the ability to monitor agriculture in a continuous manner, and allow for human interventions in real-time for remediation actions [4].

In general, smart IoT irrigation systems utilize a set of agricultural sensors to collect the most relevant environmental and climatic information in order to decide the amount of water that should be pumped to the crop. Different types of sensors are used for this purpose, such as humidity, rain, soil moisture, soil pH, illumination, and temperature sensors [5–8]. This environmental and climatic information is transmitted and analyzed by

the service provider of the system (i.e., the system analyzer server), which forwards the processed results to the responsible agriculture professional, who can provide immediate orders to the water pumping actuators. The system analyzer server in such a system can analyze the arriving environmental and climatic information in order to predict the level of required water by utilizing different machine learning algorithms, such as decision trees, k-nearest neighbors (KNN), random forest, and support vector machine [9]. Furthermore, with recent advances, most existing smart irrigation systems have employed the Internet of Things (IoT) and Wireless Sensor Network (WSN) technologies in their development in order to achieve communication lines between the system elements [1–8].

1.1. Architecture of Smart IoT Irrigation Systems

Figure 1 shows the proposed smart IoT irrigation system architecture, where each cluster head node is responsible for a specific type of crop and requires specific environmental and climatic data. These data are collected using specific sensors, which are then analyzed to determine the required quantity of water by the service provider for such a crop. Thus, the cluster head collects the required environmental and climatic data and forwards them to the service provider periodically using an internet connection. The system analyzer server then analyzes the received data and predicts the required water quantity. According to the irrigation scheduling of the crop, the responsible agriculture professional sends a request to receive the predicted required water quantity from the service provider. The irrigation orders are then sent by the responsible agriculture professional to the assigned water pumping actuators in order to irrigate the crop by a specific cluster head node. Therefore, each agriculture professional is responsible for specific types of crops, operating through specific cluster heads.

Figure 1. Proposed smart IoT irrigation system architecture.

In the same context, each cluster head should be managed and maintained by a specific operator. Finally, the authentication center server (AuC) provides authentication services to validate the identity and authority of the communication elements. Therefore, based on the characteristics of such system environments, which are proposed using IoT and WSN

technologies, addressing security issues are considered an essential mission to protect the system from different attack scenarios [10].

1.2. Vulnerabilities of Smart IoT Irrigation Systems

In terms of crop growth, smart irrigation plays a vital role in increasing crops yield but can also cause crop damage. Excessive pumping or a lack of water can lead to malnutrition, eventually causing agricultural production to become unusable [1–9].

An unauthorized party could also modify exchanged messages throughout insecure communication channels to transmit a wrong order by impersonating agriculture professionals. In the same context, an unauthorized party could also access the environmental and climatic information collected by sensors and cluster head nodes, leading to consequences such as the loss of private business data.

The limited capabilities of the sensor nodes themselves are considered another source of weakness. Different types of attacks can exploit these weaknesses, such as cloning of pump water data; malicious substitution of the sensor nodes, cluster head nodes, and agriculture professionals; and man-in-the-middle (MiTM) and denial-of-service (DoS) attacks [9–11].

The agriculture professional, cluster head nodes, and sensor nodes must be authenticated by the irrigation system to ensure their legality before transmitting the actual environmental and climatic data in order to ensure and maintain the quantity of pumped water. Therefore, authentication and a key agreement scheme provide an optimal solution to protect and achieve a high level of security in smart IoT irrigation systems [12–18].

1.3. Related Work

Numerous authentication schemes were proposed for IoT and WSN environments for different critical systems such as healthcare, smart homes, retail and supply chain management, and industrial IoT solutions systems [10,12]. Unfortunately, few researchers have specifically proposed authentication schemes to protect the irrigation systems and address the associated security concerns.

The healthcare IoT system is considered a closely similar system to IoT irrigation systems in terms of the used system elements and connectivity. In 2018, Amin et al. [13] proposed a new two-factor authentication scheme for WMSNs. They claimed that their authentication scheme could resist existing well-known attacks. In 2018, Ali et al. [14] demonstrated that the authentication schemes that were proposed by [13] could not withstand password guessing attacks, identity guessing, and user impersonation attacks, and proposed an enhanced three factor-based authentication protocol using wireless medical sensor networks for healthcare monitoring.

In 2019, Shuai et al. [15] proposed a three-factor authentication scheme to observe the patient remotely using WSNs. Shuai et al. [15] showed that the authentication schemes that were proposed in [13,14] cannot support the forward security and cannot resist the desynchronization attack. In 2020, Fotouhi et al. [16] showed that the authentication schemes that were proposed in [13,14] could not support sensor anonymity or perfect forward secrecy services. Therefore, they proposed a lightweight, secure two-factor authentication scheme for healthcare monitoring systems in order to prevent the mentioned attacks.

In 2021, Nashwan [17] demonstrated that the authentication schemes that were proposed by [15,16] could not withstand fully mutual authentication, sensor node anonymity, and resist the sensor node impersonation attack, and suggested an authentication scheme to support a high level of security for healthcare IoT systems using WMSNs.

According to the literature review, the only authentication scheme for irrigation systems was in 2018 and was proposed by Ali et al. [18]. It is called secure user authentication and a key-agreement scheme using wireless sensor networks for agricultural monitoring. Authors claimed that their scheme could withstand well-known attack types, providing user anonymity and other attractive security features. However, this scheme cannot fully

support mutual authentication between all system elements and is susceptible to various types of attacks, such as desynchronization and smart card attacks.

It can be noticed that none of the above-proposed schemes can achieve full authentication services between all communication elements of a system (i.e., sensor nodes, authentication center, and agriculture professionals) and establish secure session keys between them to protect the exchanged data streaming. Therefore, this research paper proposes a secure authentication scheme using Diffie–Hellman key agreement for a smart IoT irrigation system in order to achieve a high level of full authentication services with plausible performance and efficiency.

1.4. Motivations and Contributions

The smart IoT irrigation systems are important in developing the agricultural sector to achieve economic growth at the global level, as they can reduce water waste according to the actual needs of various crops, improve agricultural productivity, provide the ability to monitor agriculture in a continuous manner, and allow for human interventions in real-time for suitable remediation actions. An authentication scheme by integrating the smart IoT irrigation systems with WSN features can make such systems more secure and widely used. The main contributions of this research can be summarized as follows.

- An architecture for the smart IoT irrigation systems using WSN is proposed, including the main authentication components and the communication flow;
- An authentication scheme for smart IoT irrigation is proposed to achieve simultaneous full authentication services for all elements in the system;
- The proposed scheme used Diffie–Hellman cryptographic method to achieve the key agreement service in an innovative way wherein all cryptographic parameters were hidden using a set of one-way-hash functions;
- The Burrows–Abadi–Needham (BAN) logic model was used to verify mutual authentication between the components of the system;
- Informally, common attack scenarios were assumed and discussed in order to demonstrate that the proposed scheme can resist all related well-known attacks. Comparisons with other existing schemes in terms of security and efficiency features are presented;
- Finally, the results obviously show that the proposed scheme is feasible to use, with reasonable computation and communication outcomes.

1.5. Paper Organization

The remainder of this paper is organized as follows: The proposed scheme is presented in Section 2. In the first part of Section 3, the formal analysis using BAN logic is presented, while an informal analysis of the security achievements, resistance against well-known attacks, and comparisons with recently proposed schemes are provided in the second and third parts of Section 3. Section 4 details the performance evaluation of the proposed scheme. Finally, the conclusion is given in Section 5.

2. Proposed Scheme

This section proposes a secure authentication scheme using Diffie–Hellman key agreement for a smart IoT irrigation system using WSNs. The smart IoT irrigation system contains four main communication elements, namely, the agriculture professional (AP), authentication server (AuC), cluster head (CH), and agriculture sensor (AS).

Furthermore, the proposed scheme includes nine phases, namely, the registration phase for AP, registration phase for CH, the registration phase for AS, login phase for AP, login phase for CH, authentication and session key agreement phase, AS authentication phase, password change phase for AP, and password change phase for CH. The main cryptographic techniques that are used to achieve the desired security services are the discrete logarithm problem (Diffie–Hellman technique) and one-way hash functions [12]. Finally, the basic notation used to present the proposed scheme in the following is listed in Table 1.

Table 1. Basic notation and meanings.

Notation	Description
AuC	Authentication center server.
APi	Agriculture professional.
CHj	Cluster head.
ASk	Agriculture Sensor.
SCi	Smartcard of APi.
SCj	Smartcard of APi.
APiDi	Identity of APi.
PWi	Password of APi.
Si	Security code of APi.
R0	Random number generated by APi.
T0, T4	Timestamps of APi.
$\delta 1, \delta 2$	Selected Integer numbers by APi over P.
CHiDj	Identity of CHj.
PWj	Password of CHj.
Sj	Security code of CHj.
R2, R4	Random numbers generated by CHj.
T2	Timestamp of CHj.
$\delta 3, \delta 4$	Selected Integer numbers by CHj over P.
R1, R3, R5, R6	Random numbers generated by AuC.
T3, T4	Timestamp of AuC.
Ki	Secret Key between APi and AuC.
Kj	Secret Key between CHj and AuC.
Kij	Session Key between CHj and APi.
SSnk0, SSnk1	Sensor sequence numbers initiated by AuC.
P	Big prime number.
$\alpha, \beta i, \beta j$	Generator numbers over P selected by AuC.
T7	Random numbers generated by ASk.
$\parallel, \oplus, \text{mod}$	Concatenation, XOR, and Modulus operations.
$h(\cdot)$	One-way hash function.
$E_k(\cdot), D_k(\cdot)$	Encryption and Decryption functions.

2.1. Registration Phase of Agriculture Professional

In this phase, the agriculture professional (APi) is registered by the authentication center (AuC). This phase is accomplished over a secure channel, according to the following steps:

Step1: A new (APi) chooses their identity (APiDi), password (PWi), and secret code (Si). Then, the APi generates a random number (R0) and computes $C_i = h(\text{APiDi} \parallel \text{PWi} \parallel \text{R0})$. The APi then sends the registration request message [APiDi, Ci, Si] to the AuC.

Step2: In response to the registration request, the AuC verifies the presence of the identity (APiDi) in the agricultural professional database. If present, the AuC refuses the registration request and prompts the APi to choose another identity number. Otherwise, the AuC chooses a big prime number (P), selects a generator number α , where α is an integer number ($2 \leq \alpha \leq P - 2$). Then, the AuC selects an integer number βi , where βi is a private integer number ($2 \leq \beta i \leq P - 2$).

Next, the AuC generates the Ai0 value, where $Ai0 = ((\alpha \wedge \beta i) \text{ mod } P)$. Then, the AuC generates a random number (R1), initiates the session number $\text{SeNi} = h(R1)$, and computes $\text{APFi} = (Ai0 \oplus Si)$ and $\text{APVi} = h((\text{SeNi} \parallel Si) \oplus (Ci \parallel Ai0))$. Next, the AuC initiates the pseudonym identities $\text{APiDip} = h(\text{APiDi} \parallel \text{SeNi})$ and $\text{APiDis} = \Phi$, where Φ is an empty value. Then, the AuC inserts the APi's record (APiDi, APiDip, APiDis, SeNi) into the agriculture professional database. The AuC then personalizes the smart card (SCi) with authentication parameters ($\alpha, P, \text{SeNi}, \text{APFi}, \text{APVi}$), initiates the counter ($C0ij = 0$), create the cluster head list to connect the APi with their cluster heads as [CHiDj, C0ij], and provides the SCi to the APi.

Step3: The APi receives the SCi and then inserts R0 and the cluster head list into the memory of their device securely.

2.2. Registration Phase of Cluster Head

In this phase, the cluster head (CH_j) is registered by the AuC, and the operator of the CH_j becomes an authorized user. This phase is accomplished over a secure channel, according to the following steps:

Step1: The operator of a new CH_j selects their identity (CHID_j), password (PW_j), and secret code (S_j). Then, CH_j generates a random number R₂ and computes $C_j = h(CHID_j || PW_j || R_2)$. CH_j then sends the registration request message (CHID_j, C_j, S_j) to the AuC.

Step2: In response to the registration request, the AuC verifies the presence of the identity (CHID_j) in the cluster head database. If present, the AuC refuses the registration request and prompts the CH_j to choose another identity number. Otherwise, the AuC selects an integer number β_j , where β_j is a private integer number ($2 \leq \beta_j \leq P - 2$). Then, the AuC generates the value A_{j0} , where $A_{j0} = ((\alpha^{\beta_j}) \bmod P)$, generates a random number R₃, initiates $SeN_j = h(R_3)$, and computes $SeN = (S_j \oplus SeN_j)$, $CHF_j = (A_{j0} \oplus S_j)$, and $CHV_j = h((SeN_j || S_j) \oplus (C_j || A_{j0}))$. Then, the AuC initiates the pseudonym identities $CHID_{jp} = CHID_{js} = \Phi$, assigns a specific AP_i to the CH_j, and updates the cluster head list for AP_i securely.

Next, the AuC inserts CH_j's record (CHID_j, CHID_{jp}, CHID_{js}, SeN_j) into the cluster head database, personalize the authentication parameters (α , P, SeN, CHF_j, and CHV_j) into a new SC_j, and provides the SC_j to the CH_j.

Step3: The operator of CH_j receives the SC_j, saves R₂ into their device, and initiates CH_{1j} = 0.

2.3. Registration Phase of Agriculture Sensor

This phase prevents the agriculture sensor (AS_k) from being used by someone other than the actual cluster head CH_j. This phase is accomplished over a secure channel, according to the following steps:

Step1: A new AS_k sends the registration request message (ASID_k) to the CH_j, where the identity value (ASID_k) of AS_k was determined when it had been created by the service provider.

Step2: In response to the registration request, CH_j initiates a session number, $Snk_0 = R_7$, randomly and the sequence numbers $Sn_0 = Sn_1 = 0$. Next, CH_j inserts the AS_k's record into its sensors table [ASID_k, Sn₀, Sn_{k0}] and sends the response message (Sn₁, SN_{k0}) to the SN_k.

Step3: A new AS_k saves [Sn₁, Sn_{k0}] into its memory securely.

2.4. Login Phase of Agriculture Professional

To receive the predicted required water quantity from the service provider and send the irrigation order to water pumping actuators by a specific cluster head (CH_j), the AP_i authenticates himself to their smart card, SC_i, according to the following steps:

Step1: The AP_i enters their credential parameters (APID_i, PW_i, and S_i) as a login request into their SC_i.

Step2: In the response to the login request, the SC_i fetches R₀, computes $C_i = h(APID_i || PW_i || R_0)$, $Ai_0 = (APF_i \oplus S_i)$, and $XAPV_i = h((SeN_i || S_i) \oplus (C_i || Ai_0))$. Then, the SC_i examines whether XAPV_i matches with the APV_i that was stored in its memory by the AuC. If $XAPV_i \neq APV_i$, the SC_i refuses the request and closes the authentication session. Otherwise, the login request will be valid and the AP_i is considered an authorized user.

Step3: The SC_i initiates ID_i = $h(APID_i || SeN_i)$ and selects integer numbers δ_1 and δ_2 , where δ_1 and δ_2 are private integer numbers ($2 \leq \delta_1, \delta_2 \leq P - 2$). Next, SC_i computes the value Ai₁, where $Ai_1 = ((\alpha^{\delta_1}) \bmod P)$, computes Ai₂, where $Ai_2 = ((\alpha^{\delta_2}) \bmod P)$, and generates the private key value Ki, where $K_i = ((Ai_0^{\delta_1}) \bmod P)$.

2.5. Login Phase of Cluster Head

To send the environmental and climatic information collected by the agricultural sensors to the service provider and receive the irrigation orders from the APi, CHj authenticates himself to their smart card SCj, according to the following steps:

Step1: The operator of the CHj enters the credential parameters (CHIDj, PWj, and Sj) as a login request to the SCj.

Step2: In the response to the login request, the SCj fetches R2 and computes $SeNj = (Sj \oplus SeN)$, $Cj = h(CHIDj || PWj || R2)$, $Aj0 = (CHFj \oplus Sj)$, and $XCHVj = h((SeNj || Sj) \oplus (Cj || Aj0))$. Then, SCj examines whether XCHVj matches with the CHVj that was stored in its memory by the AuC. If $XCHVj \neq CHVj$, the SCj refuses the request and closes the authentication session. Otherwise, the login request will be valid, and the CHj will be used by an authorized operator.

Step3: SCj selects an integer number $\delta3$, where $\delta3$ is a private integer number ($2 \leq \delta3 \leq P - 2$). Next, SCj computes the value $Aj1$, where $Aj1 = ((\alpha \wedge \delta3) \bmod P)$, stores $\delta3$, and sends $Aj1$ to the AuC.

Step4: The AuC receives and enters the value of $Aj1$ into the CHj's record.

2.6. Authentication and Session Key Agreement Phase

In this phase, the APi needs to achieve mutual authentication with the AuC and CHj, as well as establish a shared session key with the CHj through insecure communication channels. Figure 2 summarizes the authentication and session key agreement phase between the APi, CHj, and AuC. The following steps are executed at this phase after the login phases of the APi and CHj have been executed.

Step1: The APi initiates the authentication request via their SCi by inserting the identity of the CHj (CHIDj). Then, the APi computes $CTi0 = EK_i(T0 || Ai2 || CHIDj)$ and $Vi0 = h(T0 || Ai2 || SeNi)$, where T0 is the current timestamp of the APi. Then, the APi sends an authentication request [$M1 = IDi, Ai1, CTi0, Vi0$] to the AuC.

Step2: Upon receiving the APi's request, the AuC searches in the agriculture professional database to find APIDip and APIDis values based on the IDi received from the APi. One of the following four events will be triggered by the AuC, which were previously employed in [10,17,19] to achieve user anonymity services.

In the first event, wherein $IDi \neq APIDip$ and $IDi \neq APIDis$, the AuC refuses the authentication request and closes the authentication session.

In the second event, wherein $IDi = APIDip$ and $APIDis \neq \Phi$, the AuC computes $SeNi = h(SeNi)$, generates $Ki = ((Ai1 \wedge \beta i) \bmod P)$, and decrypts $DKi(CTi0) = (T0 || Ai2 || CHIDj)$. The AuC then examines whether the APi can access CHj with identity CHIDj. If it does not satisfy the conditions, the AuC refuses the authentication request and closes the authentication session. Otherwise, the AuC examines the value of T0. If it does not hold, the AuC refuses the authentication request and closes the authentication session. Otherwise, AuC calculates $XVi0 = h(T0 || Ai2 || SeNi)$ in order to examine whether $XVi0$ matches with the $Vi0$ that was received. If $XVi0 = Vi0$, the AuC updates the following identity values $APIDis = APIDip$ and $APIDip = h(IDi || Ai2)$, respectively. Otherwise, the AuC refuses the authentication request and closes the authentication session.

In the third event, wherein $IDi = APIDip$ and $APIDis = \Phi$, the AuC generates $Ki = ((Ai1 \wedge \beta i) \bmod P)$ and decrypts $DKi(CTi0) = (T0 || Ai2 || CHIDj)$. Then, the AuC examines whether the APi can access CHj with identity CHIDj. If not, the AuC refuses the authentication request and closes the authentication session. Otherwise, the AuC examines the value of T0. If it does not hold, the AuC refuses the authentication request and closes the authentication session. Otherwise, the AuC calculates $XVi0 = h(T0 || Ai2 || SeNi)$ in order to examine whether $XVi0$ matches with the $Vi0$ that was received. If $XVi0 = Vi0$, the AuC updates the following identity values $APIDis = APIDip$ and $APIDip = h(IDi || Ai2)$, respectively. Otherwise, the AuC refuses the authentication request and closes the authentication session.

Figure 2. Authentication and session key agreement phase.

In the fourth event, wherein IDi = APIDis, the AuC generates Ki = $((Ai1 \wedge \beta i) \bmod P)$ and decrypts DKi (CTi0) = $(T0 \parallel Ai2 \parallel CHIDj)$. Then, the AuC examines whether the APi can access CHj with identity CHIDj. If it does not satisfy, AuC refuses the authentication request and closes the authentication session. Otherwise, the AuC examines the value of T0. If it does not hold, the AuC refuses the authentication request and closes the authentication session. Otherwise, the AuC calculates XVi0 = $h(T0 \parallel Ai2 \parallel SeNi)$ to examine whether the XVi0 matches with the Vi0 that was received. If XVi0 = Vi0, the AuC updates the identity value of the APIDip = $h(IDi \parallel Ai2)$. Otherwise, the AuC refuses the authentication request and closes the authentication session.

Step3: Using the values of $APID_i$ and $CHID_j$, the AuC fetches CH_j 's record from the cluster head database. Then, the AuC calculates $K_j = ((A_j1 \wedge \beta_j) \bmod P)$, initiates the session counter $CH0_j = (CH0_j + 1)$, computes the pseudonym identity of the cluster head $CHID_{jp} = h(CHID_j || CHID_{jp})$, and calculates $SeN_j = h(SeN_j)$. Next, the AuC generates a random number $R4$ and computes $CT_{j0} = EK_j(T1 || R4 || Ai2)$ and $V_{j0} = h(T1 || Ai2 || CHID_{jp} || R4)$, where $T1$ represents the current timestamp of the AuC. The AuC then sends an authentication request $[M2 = CH0_j, CT_{j0}, \text{ and } V_{j0}]$ to CH_j .

Step4: In response to the authentication request, CH_j , by its SC_j , computes $\Delta CH0_{1j} = (CH0_j - CH1_j)$ and checks whether the value of $\Delta CH0_{1j}$ satisfies $(1 \leq \Delta CH0_{1j} \leq \mu_2)$, where the value of μ_2 is assigned according to the irrigation system specifications. If it does not hold, CH_j refuses the authentication request and closes the authentication session. Otherwise, it retrieves the value of the $SeN_j = (S_j \oplus SeN)$ and computes $SeN_j = h(SeN_j) \Delta CH0_{1j} - 1$ times, until $(\Delta CH0_{1j} - 1) = 1$. Then, CH_j updates $SeN = (S_j \oplus SeN_j)$, generates K_j , where $K_j = ((A_{j0} \wedge \delta_3) \bmod P)$, and computes $DK_j(CT_{j0}) = (T1 || R4 || Ai2)$. Next, CH_j examines the value of $T1$. If it does not hold, the CH_j refuses the authentication request and closes the authentication session. Otherwise, CH_j sets $CHID_{js} = CHID_{jp}$ and computes $CHID_{js} = h(CHID_j || CHID_{js}) \Delta CH0_{1j} - 1$ times, until $(\Delta CH0_{1j} - 1) = 1$. Then, CH_j computes $XV_{j0} = h(T1 || Ai2 || CHID_{js} || R4)$ in order to verify whether XV_{j0} matches the V_{j0} that was received. If $XV_{j0} \neq V_{j0}$, CH_j refuses the authentication request and closes the authentication session. Otherwise, the AuC is considered an authorized server. CH_j then chooses an integer number δ_4 , where δ_4 is a private integer number $(2 \leq \delta_4 \leq P - 2)$, computes A_{j2} , where $A_{j2} = ((\alpha \wedge \delta_4) \bmod P)$, generates a random number $R5$ and computes $CT_{j1} = EK_j(T2 || R5 || A_{j2})$ and $V_{j1} = h(T2 || A_{j2} || CHID_{js} || R5)$, where $T2$ is the current timestamp of CH_j . Next, CH_j sets $CHC1_j = CHC0_j$ and computes the session key K_{ij} with AP_i , where $K_{ij} = (Ai2 \wedge \delta_4) \bmod P$. Finally, CH_j sends the response authentication $[M3 = CHID_{js}, CT_{j1}, \text{ and } V_{j1}]$ to the AuC.

Step5: Upon receiving the response message, the AuC computes $DK_j(CT_{j1}) = (T2 || R5 || A_{j2})$, where the pseudonym identity $CHID_{js} = CHID_{jp}$. Then, the AuC examines the value of $T2$. If it does not satisfy, the AuC refuses the response message and closes the authentication session. Otherwise, AuC computes $XV_{j1} = h(T2 || A_{j2} || CHID_{js} || R5)$ in order to examine whether XV_{j1} matches V_{j1} . If $XV_{j1} \neq V_{j1}$, the AuC refuses the response message and closes the authentication session. Otherwise, the CH_j is considered a legitimate node and can be used by an authorized operator. The AuC then generates a random number $R6$ and computes $CT_{i1} = EK_i(R6 || A_{j2} || T3)$, where $T3$ is the current timestamp of the AuC. Then, the AuC computes $Vi1 = h(APID_i || A_{j2} || R6 || SeNi || T3)$ and sends the response authentication $[M4 = CT_{i1}, Vi1]$ to the AP_i .

Step6: When the response message is received, the AP_i computes $DK_i(CT_{i1}) = (R6 || A_{j2} || T3)$. Then, the AP_i examines the value of $T3$. If it does not hold, AP_i refuses the response message and closes the authentication session. Otherwise, the AP_i computes $XVi1 = h(APID_i || A_{j2} || R6 || SeNi || T3)$, in order to examine whether $XVi1$ matches with $Vi1$. If $XVi1 \neq Vi1$, the AP_i refuses the response message and closes the authentication session. Otherwise, AuC is considered an authorized server. AP_i then computes $Vi2 = h(APID_i || R6 || SeNi || (T4 - T3))$ and $Vix = ((T4 \oplus T3) || Vi2)$, where $T4$ is the current timestamp of the AP_i . Then, they update $SeNi = h(SeNi)$ and set $ID_i = h(ID_i || Ai2)$, sending an acknowledgment $[M5 = ID_i, Vix]$ to the AuC. Finally, the AP_i computes the session key K_{ij} with CH_j , as $K_{ij} = (A_{j2} \wedge \delta_2) \bmod P$.

Step7: Upon receiving $M5$, the AuC computes $T4 = ((T4 \oplus T3) \oplus T3)$ and $\Delta TP = (T4 - T3)$. Then, the AuC examines whether the value of ΔTP holds, according to the value of μ_3 , where μ_3 is determined based on the irrigation system requirements. If it does not hold, the AuC sends the response authentication $[M4 \text{ with a fresh } T3]$ to the AP_i . Otherwise, the AuC computes $XVi2 = h(APID_i || R6 || SeNi || \Delta TP)$, in order to examine whether $XVi2$ matches with the received $Vi2$. If $XVi2 \neq Vi2$, AuC refuses the acknowledgment message and closes the authentication session. Otherwise, the AP_i is considered an authorized user. Then, the AuC updates $SeNi = h(SeNi)$ and $APID_i = \Phi$.

2.7. Authentication Phase of Agriculture Sensor

In order to exchange the environmental/climatic information and irrigation orders between the CHj and connected sensors (ASk), mutual authentication must be achieved for all sessions between both of them through insecure channels, as shown in Figure 3. The main steps of this phase are achieved as follows:

Figure 3. Authentication phase of agriculture sensor.

Step1: To connect with a specific sensor ASk, CHj determines the identity ASIDk of the ASk that wants to connect with it. Then, CHj generates a private key (ASkK) randomly, updates $Snk0 = h(Snk0 || ASIDk)$, and calculates $CTk = ((ASkK || ST) \oplus h(Snk0 || ASIDk || Sn0))$, where the value of ST is used to determine whether the CHj needs to receive the data or forward the irrigation orders. Next, CHj calculates the $IDk = h(ASkK || ASIDk)$, wherein Idk is a pseudonym identity for the ASk. Then, CHj calculates $Vk0 = h(ST || ASIDk || ASkK || Snk0 || Sn0)$ and renews $Sn0 = Sn0 + 1$. Finally, CHj transmits the authentication request $[M6 = CTk, Vk0, Sn0]$ to the ASk.

Step2: In response to the request message, the ASk calculates $\Delta Sn = (Sn0 - Sn1)$ and examines whether $(1 \leq \Delta Sn \leq \mu_0)$, where the value of μ_0 is determined according to the system specifications. If not, the ASk refuses the request message and closes the authentication session. Otherwise, the ASk initiates the $Snk1 = Snk0$, and executes $Snk1 = h(Snk1 || ASIDk)$ for ΔSn times, until $(\Delta Sn - 1) = 1$.

Next, the ASk computes $(ASkK || ST) = CTk \oplus h(Snk0 || ASIDk || Sn1)$ and computes $XVk0 = h(ST || ASIDk || Snk1 || ASkK || Sn0 - 1)$. Then, the ASk checks whether XVk0 matches Vk0. If $XVk0 \neq Vk0$, the ASk refuses the request message and closes the authentication session. Otherwise, the CHj is considered legitimate and can be used by an authorized operator.

The ASk calculates $Snk0 = h(Snk1 || ASIDk)$, $Vk1 = h(ST || ASIDk || ASkK || Snk0 || Sn0)$, and $IDk = h(ASkK || ASIDk)$. Then, the ASk updates $Sn1 = Sn0$ and calculates $Snk0 = h(Snk1 || ASIDk)$. Finally, the ASk transmits the response message $[M7 = IDk, Vk1]$ to the CHj.

Step3: Upon receiving the response message, CHj computes $Snk0 = h(Snk0 || ASIDk)$, $XVk1 = h(ST || ASIDk || ASkK || Snk0 || Sn0)$, and checks whether XVk2 matches with Vk1. If $Vk2 = Vk1$, the ASk is an authorized sensor. Otherwise, the CHj refuses the response message and closes the authentication session.

2.8. Password Change Phase of Agriculture Professional

This phase is accomplished between the AP_i and their SC_i when the AP_i wants to change their password without increasing the network overhead and back to the AuC. The AP_i needs to execute the following steps:

Step1: The AP_i enters their current credential parameters ($APID_i$, PW_i , Si) and a new password ($*PW_i$), as the request to change their password;

Step2: In response to the password change request, the SC_i calculates $C_i = h(APID_i || PW_i || R_0)$, $Ai_0 = (APF_i \oplus Si)$, and $XAPV_i = h((SeNi || Si) \oplus (C_i || Ai_0))$. Then, the SC_i examines whether $XAPV_i$ matches the APV_i . If $XAPV_i \neq APV_i$, SC_i refuses the password change request. Otherwise, SC_i computes $*C_i = h(APID_i || *PW_i || R_0)$ and a new verification code $*APV_i = h((SeNi || Si) \oplus (C_i || Ai_0))$. Finally, the SC_i replaces the computed verification code with the previous one ($APV_i = *APV_i$).

2.9. Password Change Phase of Cluster Head

This stage is accomplished between the CH_j and SC_j , when the operator of CH_j wants to change their password without increasing the network overhead and back to the AuC. The operator of the CH_j needs to execute the following steps:

Step1: The operator enters their current credential parameters ($CHID_j$, PW_j , and S_j) and a new password ($*PW_j$) as the request to change their password;

Step2: In response to the password change request, SC_j calculates $C_j = h(CHID_j || PW_j || R_2)$, $Aj_0 = (CHF_j \oplus S_j)$, and $XCHV_j = h((SeNj || S_j) \oplus (C_j || Aj_0))$. The SC_j then examines whether $XCHV_j$ matches with CHV_j . If $XCHV_j \neq CHV_j$, the SC_j refuses the password change request. Otherwise, the SC_j computes $*C_j = h(CHID_j || *PW_j || R_2)$ and $*CHV_j = h((SeNj || S_j) \oplus (*C_j || Aj_0))$. Finally, the SC_j replaces the computed verification code with the previous one ($CHV_j = *CHV_j$).

3. Security Analysis

This section examines the security features of the proposed scheme. In the first part, the proposed scheme is verified in order to prove that it can provide mutual authentication formally, using the BAN logic model. The second part presents the informal security analysis, demonstrating that the proposed scheme can achieve attractive security features with the ability to resist well-known attacks. Finally, comparisons based on different security features between the proposed scheme and other authentication schemes that were proposed recently are presented.

3.1. Formal Verification Analysis

In the literature review, the BAN logic model is considered the most popular formal verification model that is used to examine the source, freshness, and originality of the authentication messages exchanged during authentication scheme execution [20].

In the proposed scheme, the registration, login, and password change phases are executed through secure channels or are not used frequently. Therefore, this part concentrates on the soundness of the authentication phases between the main elements of the irrigation system. In this section, the BAN logic model is used to demonstrate that the authentication messages that have been exchanged throughout the authentication phases between system elements are original and fresh messages. The main used notation of the BAN logic model can be summarized as follows:

1. $X \equiv F$ means that the statement (X) is considered a true statement for a principle (F);
2. $F \triangleleft X$ means that the statement (X) can be received and forwarded by a principle (F);
3. $F \vdash \sim X$ means that the principle (F) can send a message containing statement (X);
4. $F \Rightarrow \bigcap X$ means that the principle (F) has an authority on statement (X);
5. $\#(X)$ means that the statement (X) is an original and fresh statement;
6. (X, Y) means that the statements (X) and/or (Y) are a portion of the formula (X, Y);
7. $\langle X \rangle Y$ means that the statement (X) has been merged with the statement (Y);
8. $\{X\}K$ means that the statement (X) has been encrypted by a shared key (K);

9. $(F \rightarrow K \leftarrow Q)$ means that principles (F) and (Q) can communicate with each other using the shared key (K);
10. SK denotes a session key for the current communication session.

The main rules, goals, idealized forms, and assumptions that are utilized in the verification process using the BAN logic model are presented in Tables 2–8.

Table 2. Verification rules of ban logic model.

Rule	Rule Formula
Meaning (Ru1)	If $(F \models F \rightarrow K \leftarrow Q, F \triangleleft \langle X \rangle_K)$ then $(F \models Q \mid \sim X)$
Freshness (Ru2)	If $(F \models \#(X))$ then $(F \models \#(X, Y))$
Belief (Ru3)	If $(F \models X, F \models Y)$ then $(F \models (X, Y))$
Verification (Ru4)	If $(F \models \#(X), F \models Q \mid \sim X)$ then $(F \models Q \mid \equiv X)$
Jurisdiction (Ru5)	If $(F \models Q \Rightarrow \bigcap X, F \models Q \mid \equiv X)$ then $(F \models X)$
Session key (R6)	If $(F \models \#(X), F \models Q \mid \equiv X)$ then $(F \models F \rightarrow K \leftarrow Q)$

Table 3. Goals of authentication and key agreement phase.

Goal	Description
Go1	$AuC \mid \equiv AuC \rightarrow SK \leftarrow APi$
Go2	$AuC \mid \equiv APi \mid \equiv AuC \rightarrow SK \leftarrow APi$
Go3	$CHj \mid \equiv CHj \rightarrow SK \leftarrow AuC$
Go4	$CHj \mid \equiv AuC \mid \equiv CHj \rightarrow SK \leftarrow AuC$
Go5	$AuC \mid \equiv AuC \rightarrow SK \leftarrow CHj$
Go6	$AuC \mid \equiv CHj \mid \equiv AuC \rightarrow SK \leftarrow CHj$
Go7	$APi \mid \equiv APi \rightarrow SK \leftarrow AuC$
Go8	$APi \mid \equiv AuC \mid \equiv APi \rightarrow SK \leftarrow AuC$

Table 4. Goals of the authentication phase of agriculture sensor.

Goal	Description
G9	$CHj \mid \equiv CHj \rightarrow SK \leftarrow ASk$
G10	$CHj \mid \equiv ASk \mid \equiv CHj \rightarrow SK \leftarrow ASk$
G11	$ASk \mid \equiv ASk \rightarrow SK \leftarrow CHj$
G12	$ASk \mid \equiv CHj \mid \equiv ASk \rightarrow SK \leftarrow CHj$

Table 5. Idealized form of the authentication and key agreement phase messages.

Message	Idealized Form
M1	$IDi, CTi0, Vi0: \langle (T0, SeNi) \rangle Ki.$
M2	$CH0j, CTj0, Vj0: \langle (T1, R4) \rangle Kj.$
M3	$CHIDjs, CTj1, Vj1: \langle (T2, R5) \rangle Kj.$
M4	$CTi1, Vi1: \langle (T3, SeNi, R6) \rangle Ki.$

Table 6. Idealized form of the authentication phase of agriculture sensor.

Message	Idealized Form
M6	$CTk, Vk0: \langle (Sn0, Snk0) \rangle h(ASkK)$
M7	$Vk2: \langle (Sn0, Snk0) \rangle h(ASkK)$

Table 7. Assumptions of the authentication and key agreement phase.

Assumption	Description
A1	$APi \models \#(T3, R6)$
A2	$APi \models AuC \Rightarrow \cap (T3, R6)$
A3	$AuC \models \#(T0, T2, T4, R5)$
A4	$AuC \models APi \Rightarrow \cap (T0, T4, SeNi)$
A5	$AuC \models CHj \Rightarrow \cap (T2, R5)$
A6	$CHj \models \#(R4, T1)$
A7	$CHj \models AuC \Rightarrow \cap (R4, T1)$
A8	$APi \models APi \rightarrow Ki \leftarrow AuC.$
A9	$AuC \models AuC \rightarrow Ki \leftarrow APi$
A10	$AuC \models AuC \rightarrow Ki \leftarrow CHj$
A11	$CHj \models CHj \rightarrow Ki \leftarrow AuC$

Table 8. Assumptions of the authentication phase of agriculture sensor.

Assumption	Description
A12	$CHj \models \#(SnK0, Sn0)$
A13	$CHj \models Ask \Rightarrow \cap (SnK0, Sn0)$
A14	$ASK \models \#(SnK0, Sn0)$
A15	$ASK \models CHj \Rightarrow \cap (SnK0, Sn0)$
A16	$CHj \models CHj \rightarrow ASkK \leftarrow ASk$
A17	$ASK \models Ask \rightarrow ASkK \leftarrow CHj$

The authentication and key agreement phase use different parameters to support mutual authentication between APi, AuC, and CHj. Ki and Kj are symmetric secret keys that are used to encrypt authentication messages between the APi and AuC, and CHj and AuC, respectively. Kij is an agreed symmetric secret key between APi and CHj. The exchanged messages comprise fresh parameters, such as timestamps (T1, T2, T3, T4) and random numbers (R4, R5, and R6).

Meanwhile, to support mutual authentication in the AS authentication phase, both of CHj and ASk utilize a set of sequential numbers (Sn0, Sn1), session numbers (Snk0, Snk1), and a symmetric secret key (ASkK), which is used to encrypt the authentication messages between them. The validation and verification processes of the authentication and key agreement phase can be summarized as follows:

(1) Using M1, a1: $(AuC \triangleleft IDi, Vi1, CTi1: \langle (T0, SeNi) \rangle Ki)$ can be seen. From a1, A9, Ru3, and Ru1, a2: $(AuC \models APi \sim \langle (T0, SeNi) \rangle Ki)$ can be obtained. Using A3 and Ru2, a3: $(AuC \models \# \langle (T0, SeNi) \rangle Ki)$ can be obtained. Then, using a2, a3, and Ru4, a4: $(AuC \models APi \models \langle (T0, SeNi) \rangle Ki)$ can be obtained. Thus, from a3, a4, and Ru6, a5: $(AuC \models (AuC \rightarrow SK \leftarrow Ui))$ can be deduced, which represents Go1. Considering A4, a5, and Ru4, Q6: $(AuC \models APi \models AuC \rightarrow SK \leftarrow APi)$ can also be deduced, which leads to Go2.

(2) Using M2, b1: $(CHj \triangleleft CH0j, CTj0, Vj0: \langle (T1, R4) \rangle Kj)$ can be seen. Therefore, from b1, A11, Ru3, and Ru1, b2: $(CHj \models AuC \sim \langle (T1, R4) \rangle Kj)$ can be obtained. Next, using A6 and Ru2, b3: $(CHj \models \# \langle (T1, R4) \rangle Kj)$ can be obtained. Then, using b2, b3, and Ru4, b4: $(CHj \models AuC \models \langle (T1, R4) \rangle Kj)$ can be obtained. Thus, from b3, b4, and Ru6, b5: $(CHj \models CHj \rightarrow SK \leftarrow AuC)$ can be deduced, which represents Go3. Then, using A7, b5, and Ru4, b6: $(CHj \models AuC \models CHj \rightarrow SK \leftarrow AuC)$ can also be deduced, which leads to Go4.

(3) Using M3, c1: $(AuC \triangleleft CHIDjs, CTj1, Vj1: \langle (T2, R5) \rangle Kj)$ can be seen. Thus, from c1, A10, Ru3, and Ru1, c2: $(AuC \models CHj \sim \langle (T2, R5) \rangle Kj)$ can be obtained. Next, using A3 and Ru2, c3: $(AuC \models \# \langle (T2, R5) \rangle Kj)$ can be obtained. Then, using c2, c3, and Ru4, c4: $(AuC \models CHj \langle (T2, R5) \rangle Kj)$ can be obtained. Thus, from c3, c4, and Ru6, c5: $(AuC \models AuC \rightarrow SK \leftarrow CHj)$ can be deduced, which leads to Go5. Then, using A5, c5, and Ru4, c6: $(AuC \models CHj \models AuC \rightarrow SK \leftarrow CHj)$ can also be deduced, which leads to Go6.

(4) Using M4, d1: $(APi \triangleleft CTi1, Vi1: \langle (T3, SeNi, R6) \rangle Ki)$ can be seen. Thus, from d1, A8, Ru3, and Ru1, d2: $(APi \models AuC \sim \langle (T3, SeNi, R6) \rangle Ki)$ can be obtained. Next, using

A1 and Ru2, d3: $(APi \mid \equiv \# \langle (T3, SeNi, R6) \rangle Ki)$ can be obtained. Then, using d2, d3, and Ru4, d4: $(APi \mid \equiv AuC \mid \equiv \langle (T3, SeNi, R6) \rangle Ki)$ can be deduced. Thus, from d3, d4, and Ru6, d5: $(APi \mid \equiv APi \rightarrow SK \leftarrow AuC)$ can be deduced, which leads to Go7. Then, according to A2, d5, and Ru4, d6: $(APi \mid \equiv AuC \mid \equiv APi \rightarrow SK \leftarrow AuC)$ can also be deduced, which leads to Go8.

According to (1), (2), (3), and (4), the authentication and key agreement phase goals can be proven using the BAN logic model and, thus, mutual authentication can be achieved between APi, CHj, and AuC throughout this phase.

The validation and verification processes of the AS authentication phase can be summarized as follows:

(5) Using M6, e1: $(ASK \triangleleft CTK, Vk0: \langle Sn0, Snk0 \rangle h(ASKK))$ can be seen. From e1, A17, Ru3, and Ru1, e2: $(ASK \mid \equiv CHj \mid \sim \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Then, using A14 and Ru2, e3: $(ASK \mid \equiv \# \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Then, using e2, e3, and Ru4, e4: $(ASK \mid \equiv CHj \mid \equiv \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Thus, from e3, e4, and Ru6, e5: $(ASK \mid \equiv ASK \rightarrow SK \leftarrow CHj)$ can be deduced, which represents Go9. Considering A15, e5, and Ru4, e6: $(ASK \mid \equiv CHj \mid \equiv ASK \rightarrow SK \leftarrow CHj)$ can also be deduced, which leads to Go10.

(6) Using M7, f1: $(CHj \triangleleft Vk2: \langle (Sn0, Snk0) \rangle h(ASKK))$ can be seen. Therefore, from f1, A16, Ru3, and Ru1, f2: $(CHj \mid \equiv ASK \mid \sim \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Next, using A12 and Ru2, f3: $(CHj \mid \equiv \# \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Then, using f2, f3, and Ru4, f4: $(CHj \mid \equiv ASK \mid \equiv \langle (Sn0, Snk0) \rangle h(ASKK))$ can be obtained. Thus, from f3, f4, and Ru6, f5: $(CHj \mid \equiv CHj \rightarrow SK \leftarrow ASK)$ can be deduced, which represents Go11. Then, using A13, f5, and Ru4, f6: $(CHj \mid \equiv ASK \mid \equiv CHj \rightarrow SK \leftarrow ASK)$ can also be deduced, which leads to Go12.

According to (5) and (6), the goals of the AS authentication phase can be proven using the BAN logic model and, so, mutual authentication can be achieved between the CHj and ASK throughout this phase. Therefore, the proposed scheme can support mutual authentication among the APi, CHj, ASK, and AuC elements during the authentication phases.

3.2. Informal Verification Analysis

This section assures that the proposed scheme can achieve attractive security features and is fully protected against well-known types of attacks.

3.2.1. Security Features Achievement

Proposition 1. *The proposed scheme can achieve session key agreement.*

Proof. Based on the Diffie–Hellman key agreement technique, elements of the irrigation system generate a set of shared secret keys to execute the authentication and key agreement phase. Ki is used to achieving mutual authentication between the AuC and APi, Kj is used to achieve mutual authentication between the AuC and CHj, and Kij represents the agreed session key between the APi and CHj. During this authentication phase, the AuC generates P with two generator numbers (β_i and β_j) in order to produce the public keys (Ai0 and Aj0) for APi and CHj, respectively. The APi, according to the P-value, generates two generator numbers (δ_1 and δ_2) to produce the public keys (Ai1 and Ai2) for AuC and CHj. The CHj, according to the P-value, generates two generator numbers (δ_3 and δ_4) to produce the public keys (Aj1 and Aj2) for AuC and APi. It should be noted that all the public keys are exchanged between system elements using different one-way hash functions in order to prevent man-in-the-middle and replay attacks. $\square$

In the AS authentication phase, the CHj generates the ASKk key randomly and sends it to the ASK in a secure manner through the use of a one-way hash function in order to achieve the mutual authentication feature between them.

Thus, the proposed scheme is able to provide the key agreement feature between the elements of the irrigation system.

Proposition 2. *The proposed scheme can achieve mutual authentication.*

Proof. The authentication and session key agreement phase contains set verification checkpoints that are performed to validate a fully mutual authentication between the APi, AuC, and CHj elements. Mutual authentication between the AuC and APi can be achieved as follows: The AuC verifies the M1 message via the received challenge authentication parameters [CTi0 and Vi0] in order to verify the legality of APi. In the same manner, the APi verifies the legitimacy of the AuC by verifying the M4 message via the received challenge authentication parameters [CTi1 and Vi1]. □

Similarly, the mutual authentication between AuC and CHj can be achieved as follows: the CHj verifies the M2 message via the received challenge authentication parameters [CH0j, CTj0, and Vj0] in order to verify the legality of the AuC. Next, the AuC verifies the legitimacy of the CHj by verifying the M3 message via the received challenge authentication parameters [CHIDjs, CTj1, and Vj1].

In the AS authentication phase, both ASk and CHj verify each other through two verification checkpoints, as follows: ASk verifies the authenticity of the CHj by checking the received authentication parameters through the M6 message that contains [CTk, Vk0, and Sn0], while CHj verifies the authenticity of the ASk by checking the received authentication parameters through the M7 message that contains [IDk, and Vk2].

Thus, the proposed scheme is able to support fully mutual authentication between the elements of the irrigation system.

Proposition 3. *The proposed scheme can achieve anonymity and untraceability.*

Proof. In the authentication and key agreement phase, in order to preserve the anonymity and untraceability for the users of the APi and CHj, all authentication messages exchanged within the authentication process do not include the actual identities, nor that of the APi (APiDi) or the CHj (CHIDj). Instead, the authentication process uses pseudonym identities (IDi and CHIDjs), which are created by one-way hash functions after completing each authentication session for the APi and CHj. Therefore, it is considered impossible for an unauthorized party to extract the actual identities of either the APi or CHj from the messages exchanged. □

Similarly, in the AS authentication phase, the exchanged messages between CHj and ASk do not include the actual identity of the ASk (ASIDk). Instead, the authentication process uses pseudonym identities (IDk).

Thus, the proposed scheme can achieve full anonymity and untraceability for the elements of the irrigation system.

Proposition 4. *The proposed scheme can achieve perfect forward secrecy.*

Proof. In the proposed scheme, if an adversary obtains one or more keys of the system elements, such as Ki, Kj, or ASkK, in some way, they still cannot acquire the secret session key (Kij) that is generated by the APi and CHj during the next authentication session. The reason for this is that, after completing the authentication and key agreement phase successfully, the generator numbers (δ_1 , δ_2 , δ_3 , and δ_4) are changed for each authentication session. □

Similarly, if an adversary obtains the secret key ASkK, they still cannot acquire the secret session key for the next authentication session between the ASk and CHj, wherein

this key will be changed in each authentication session between them using a one-way hash function.

Therefore, the proposed scheme can achieve fully perfect forward secrecy in future authentication sessions.

3.2.2. Resistance against Well-Known Attacks

Despite the fact that the exchanged messages between the elements of the system can be blocked, modified, replaced, tracked, and retransmitted over insecure communication channels by the opponent, this section demonstrates that the proposed scheme can withstand well-known attacks.

Proposition 5. *The proposed scheme can resist desynchronization attacks.*

Proof. In order to support anonymity and untraceability, the proposed scheme uses a set of pseudonym identities, such as $APiDi$, $CHiDj$, and $ASiDk$, along with a set of the one-way hash functions to renew such identities for each authentication session. Furthermore, to support the perfect forward secrecy feature, the proposed scheme uses set verification codes, such as $Vi0$, $Vj0$, $Vj1$, $Vi1$, Vix , $Vk0$, and $Vk1$, with a set of the one-way hash functions, random numbers, and timestamps to renew such codes. Therefore, the synchronization of the computed hash values must be preserved in the proposed scheme between the APi , CHj , ASk , and AuC . In the authentication and key agreement phase, the synchronization of the $APiDi$ and hash chain values of $SeNi$ will be guaranteed by $APiDip$ and $APiDis$ in each authentication session between the APi and AuC . Similarly, in each authentication session between the CHj and AuC , the values of $CHiDjp$ and $CHiDjs$ are used to ensure that the $CHiDj$ and the hash chain value of $SeNj$ are synchronized. In the AS authentication phase, the synchronization of the $ASiDk$ and hash chain values of $Snk0$ will be guaranteed by IDk in each authentication session between the ASk and CHj . As the one-way hash functions used in the proposed scheme are collision hash functions, even if the opponent can block authentication messages, the APi , CHj , ASk , and AuC can resynchronize the hash values between them. To make the discussion more precise regarding the ability of the proposed scheme to withstand desynchronization attacks, consider the following possible desynchronization attack scenarios:

S1: Suppose the opponent has blocked M1. Obviously, this will not affect the synchronization between APi and AuC during the next authentication session, wherein the elements of the system have not yet begun renewing the values of the IDi nor $h(SeNi)$. Thus, this scenario will be ignored.

S2: Suppose the opponent has blocked M2. Obviously, this will not affect the synchronization between the APi , CHj , and AuC during the following sessions. In this case, the hash values of $SeNj$ and $CHiDjs$ have only been updated on the AuC . However, M3 has not been received by the AuC . Thus, the AuC can resend M2 again, using a new $CH0j$ with distinct values of T1 and R4. Alternatively, the CHj using the value of $\Delta CH01j$ can resynchronize and update the hash values of $SeNj$ and $CHiDjs$, as in the AuC . Thus, the AuC can resynchronize the next session with the CHj .

S3: Suppose the opponent has blocked M3. Obviously, this will not affect the synchronization between the APi , CHj , and AuC during the next session. This scenario is similar to S2. In this case, the hash values of the $SeNj$ and $CHiDjs$ have been renewed in AuC and CHj . However, M3 has not been received by the AuC . Thus, the AuC will send M2 again, using distinct values for T1 and R4. Then, the CHj , using the value of $\Delta CH01j$, can resynchronize and renew the hash values of $SeNj$ and $CHiDjs$, as in the AuC . Next, the CHj can send M3 again, using new values for T2 and R5. Thus, the AuC can resynchronize in the next session with CHj .

S4: Suppose the opponent has blocked M4. Obviously, this will not affect the synchronization between the APi , CHj , and AuC during the next session. In this case, the hash values of $SeNi$ and $APiDis$ have not yet been renewed in the APi nor AuC , and only

synchronization is needed for the hash value of IDi in the APi. However, M4 has not been received by the APi. Thus, the APi can send M1 again, using a distinct value of T0. As the value of IDi for the AuC is equal to APIDis, the AuC can send M4 using the new values for T3 and R6. Thus, the APi can resynchronize in the next session with AuC.

S5: Suppose the opponent has blocked M5. Obviously, it will not affect the synchronization between the APi, CHj, and AuC during the next session. In this case, the hash values IDi and APIDip have been renewed in the APi nor AuC, wherein IDi = APIDip, and the (SeNi) has been renewed in the APi, but has not yet been updated on the AuC. However, the APi will not be able to receive processed data from the AuC. Therefore, APi can send M1 again, using a distinct value of T0. As the value of IDi for the AuC will be equal to APIDip and APIDis $\neq \Phi$, the AuC will send M4 again to the APi, using distinct values of T3 and R6. Therefore, the APi can send M5 and resynchronize in the next session with AuC.

S6: Suppose the opponent has blocked M6. Obviously, this will not affect the synchronization between the ASk and CHj during the next session. In this case, the values of CTk, Snk0, and IDk have only been renewed in the CHj. However, M7 has not been received by the CHj. Therefore, the CHj can send M6 using distinct values of ASkK and Sn0. Then, the ASk, according to the Δ Sn value, can renew the IDk, S0k1, and Snk1 values, as in the CHj. Thus, the CHj can resynchronize in the next session with ASk.

S7: Suppose the opponent has blocked M7. Obviously, this will not affect the synchronization between the ASk and CHj during the next session. In this case, the values of CTk, IDk, and Snk0 have been updated on both ASk and CHj. However, M7 has not been received by the CHj. Therefore, the CHj can send M6, using distinct values for ASkK and (Snk0). The ASk, using the new value of Δ Sn, can renew the hash values of IDk, SSk1, and SNk1, as in the CHj. Next, the ASk can send M7 again. Thus, the CHj can resynchronize in the next session with ASk. $\square$

Thus, based on the analysis of the above-discussed attack scenarios (S1–S7), the proposed scheme can resist desynchronization attacks.

Proposition 6. *The proposed scheme can resist impersonation attacks.*

Proof. To make the discussion more precise regarding the ability of the proposed scheme to withstand impersonation attacks, consider the following possible impersonation attack scenarios:

S8: Suppose the opponent has intercepted M1, which is sent to the AuC to impersonate the APi, wherein $CTi0 = EK_i(T0 \parallel Ai2 \parallel CHIDj)$ and $Vi0 = h(T0 \parallel Ai2 \parallel SeNi)$. The decryption of CTi1 is not an obtainable value, wherein the opponent cannot know Ki. Besides, the actual SeNi and Ai2 values are not available. Thus, the opponent would be unable to impersonate APi by computing Vi0 with completely separate T0 and SeN1 values.

S9: Suppose the opponent has intercepted M2, which is sent to the CHj to impersonate the AuC, wherein $CTj0 = EK_j(T1 \parallel R4 \parallel Ai2)$ and $Vj0 = h(T1 \parallel Ai2 \parallel CHIDjp \parallel R4)$. The decryption of CTj0 is not an obtainable value, wherein the opponent cannot know Kj, which is considered an invisible value. Besides, the actual CHIDjp and Aj2 values are not available. Thus, the opponent would be unable to impersonate the AuC by computing Vj0 with completely separate T1 and R4 values.

S10: Suppose the opponent has intercepted M3, which is sent to the AuC to impersonate the CHj, wherein $CTj1 = EK_j(T2 \parallel R5 \parallel Aj2)$ and $Vj1 = h(T2 \parallel Aj2 \parallel CHIDjs \parallel R5)$. The decryption of CTj1 is not an obtainable value, wherein the opponent cannot know Kj (an invisible value). Besides, the actual CHIDjp and Aj2 values are not available. Thus, the opponent would be unable to impersonate the AuC by computing Vj0 with completely separate T2 and R5 values.

S11: Suppose the opponent has intercepted (M6), which is sent to the ASk to impersonate the CHj, wherein the $CTk = ((ASKK \parallel ST) \oplus h(Snk0 \parallel ASIDk \parallel Sn1))$, $IDk = h$

(ASKK||ASIDk), and $Vk0 = h(ST||ASIDk||ASKK||Snk0||Sn0)$. Neither of the CTk nor IDK is obtainable values, wherein the opponent cannot know ASKk, which is considered an invisible value. Besides, the actual ASIDk and Snk0 values are not available. Thus, the opponent would be unable to impersonate the CHj by computing Vk0 with completely separate ASKk and ASIDk values. $\square$

Therefore, according to the above-discussed attack scenarios (S8–S11), the proposed scheme can resist impersonation attacks.

Proposition 7. *The proposed scheme can resist replay attacks.*

Proof. The proposed scheme uses a set of timestamps, random numbers, and serial numbers, which are incorporated into the exchange authentication messages. In order to make the discussion more precise regarding the ability of the proposed scheme to withstand replay attacks, consider the following possible replay attack scenarios:

S12: Suppose the opponent has retransmitted M1 that was sent by the APi. The AuC will refuse the authentication request and close the authentication session, as T0 has become out of range.

S13: Suppose the opponent has retransmitted M2 that was sent by the AuC. The CHj will refuse the authentication request and close the authentication session, as $\Delta CH01j$ will be out of the irrigation system specifications and T1 has become out of range.

S14: Suppose the opponent has retransmitted M6 that was sent by CHj. The ASK will refuse the authentication request and close the authentication session, as ΔSn will be out of the irrigation system specifications. $\square$

Thus, based on the analysis of the above-discussed attack scenarios (S12–S14), the proposed scheme can resist replay attacks.

Proposition 8. *The proposed scheme can resist stolen password table attacks.*

Proof. The proposed scheme does not have an AP password table or a CHj password table stored in the AuC at all. $\square$

Thus, the proposed scheme will not be exposed to stolen verifier table attacks and can resist such attacks.

Proposition 9. *The proposed scheme can resist incorrect password login attacks.*

Proof. Suppose the opponent tries to use an incorrect password. In the proposed scheme, the password verification code $APVi = h((SeNi||Si) \oplus (Ci||Ai0))$ is stored in the smart card of the APi, which is used to validate the APi's password. If the APi enters an incorrect password (PWi) or security code (Si), the verification codes ($XAPVi \neq APVi$) are not equal, and the SCi will refuse the login request. Similarly, the CHj has the password verification code $CHVj = h((SeNj||Sj) \oplus (Cj||Aj0))$ stored in the smart card of the CHj, which is used to validate the operator password of the CHj. If the operator enters an incorrect password (PWj) or security code (Sj), the verification codes ($XCHVj \neq CHVj$) will not be equal, and the SCj will refuse the login request. $\square$

Therefore, the proposed scheme can resist an incorrect password login attack without increasing the network overhead and going back to the AuC.

Proposition 10. *The proposed scheme can resist smartcard loss attacks.*

Proof. Suppose the opponent attempts to steal the concealed data from a smartcard, wherein the proposed authentication scheme utilizes three factors (3F) during the login

phases (i.e., identity, password, and security code). Even if an opponent is able to steal such concealed data from a smartcard, they will be incapable of logging in, whether as an APi user or CHj operator. This is as the opponent also requires knowledge of the actual user's identities (APiDi, CHiDj), (PWi, PWj), and (Si, Sj) in order to create a login message. $\square$

Thus, the proposed scheme can resist smartcard loss attacks without increasing the network overhead and going back to the AuC.

Proposition 11. *The proposed scheme can resist man-in-the-middle attacks.*

Proof. Suppose the opponent tries to forge either legitimate challenge or response messages. As the proposed scheme uses the Diffie–Hellman key agreement technique, the values of public keys ($A0i$, $Aj0$, $A1i$, $A1j$, $A2i$, and $Aj2$) are exchanged as concealed values between the elements of the irrigation system within hash values. Besides, challenge or response messages that have been exchanged are protected by (Ki and $SeNi$) and (Kj and $SeNj$), and an opponent without these pairs of keys cannot forge such legitimate messages. $\square$

Thus, the proposed scheme can resist man-in-the-middle attacks.

Proposition 12. *The proposed scheme can resist insider privileged attacks.*

Proof. Suppose the inside workers within the service provider try to carry out privileged insider attacks. In the proposed scheme, when the APi executes the registration phase, the (PWi , Si) values are sent as hidden values using a hash value $Ci = h(APiDi || PWi || Si)$. Similarly, when the CHj executes the registration phase, the (PWj , Sj) values are sent as hidden values using a hash value $Cj = h(CHiDj || PWj || Sj)$. The collision feature of the one-way hash function, which is used to compute (Ci , Cj), can stop the insider from revealing the actual values. $\square$

Thus, the proposed scheme can resist insider privileged attacks.

3.3. Security Comparisons

In this section, a comparison of security features between the proposed authentication scheme and other recently proposed schemes [13–16,18] is presented. The comparisons, in terms of the security features achievements and resistance against well-known attacks, are detailed in Table 9.

It should be noted here that the elements of irrigation systems (agriculture professionals, AuC, cluster heads, and agriculture sensors) are somewhat analogous to the elements of health care systems (healthcare professional; gateway node, GWN; patient's smart device; and physiological sensors).

The outputs of the comparison in Table 9 illustrate that the proposed scheme can achieve all of the attractive security features, while the other schemes cannot support some of the security features, such as fully mutual authentication among the elements of the system, agriculture sensor node anonymity, and perfect forward secrecy (SF1–SF11). This comparison was proved in Section 3.2.1 from proposition 1 to proposition 4.

Furthermore, the proposed scheme can resist all well-known attacks, while other schemes cannot resist all listed attack types (SF12–SF22). The resistance of the proposed scheme against all well-known attacks according to all attacks scenarios was proved in Section 3.2.2 from Proposition 5 to Proposition 12.

Table 9. Security achievements and resistance to well-known attacks.

Security Features	[13]	[14]	[15]	[16]	[18]	Proposed Scheme
SF1	No	Yes	Yes	Yes	Yes	Yes
SF2	Yes	Yes	Yes	Yes	Yes	Yes
SF3	NA	NA	NA	NA	NA	Yes
SF4	Yes	Yes	Yes	Yes	Yes	Yes
SF5	NA	NA	NA	NA	NA	Yes
SF6	Yes	Yes	Yes	Yes	Yes	Yes
SF7	No	No	Yes	Yes	Yes	Yes
SF8	NA	NA	NA	NA	NA	Yes
SF9	No	No	Yes	Yes	No	Yes
SF10	No	Yes	Yes	No	No	Yes
SF11	No	No	Yes	Yes	No	Yes
SF12	No	No	Yes	Yes	No	Yes
SF13	Yes	Yes	Yes	Yes	Yes	Yes
SF14	NA	NA	NA	NA	NA	Yes
SF15	Yes	Yes	Yes	Yes	Yes	Yes
SF16	No	Yes	Yes	Yes	Yes	Yes
SF17	No	Yes	Yes	Yes	Yes	Yes
SF18	NA	NA	NA	NA	NA	Yes
SF19	Yes	Yes	Yes	Yes	Yes	Yes
SF20	Yes	Yes	Yes	Yes	Yes	Yes
SF21	No	Yes	Yes	Yes	No	Yes
SF22	No	Yes	Yes	Yes	No	Yes

SF1 = Secure session key agreement achievement. SF2 = Correct password change phase for agriculture professional. SF3 = Correct password change phase for cluster head. SF4 = Mutual authentication between Agriculture professional and AuC. SF5 = Mutual authentication between cluster head and AuC. SF6 = Mutual authentication between agriculture sensor and system. SF7 = Agriculture professional anonymity. SF8 = Cluster head anonymity. SF9 = Agriculture sensor anonymity. SF10 = Three authentication factors (3F). SF11 = Perfect forward secrecy feature. SF12 = Desynchronization attack resistance. SF13 = Smartcard loss attack resistance for agriculture professional. SF14 = Smartcard loss attack resistance for cluster head. SF15 = Replay attack resistance. SF16 = AuC impersonation attack resistance. SF17 = Agriculture professional impersonation attack resistance. SF18 = Cluster head impersonation attack resistance. SF19 = Man-in-the-middle attack resistance. SF20 = Incorrect login attack resistance. SF21 = Insider privileged attack resistance. SF22 = Stolen password verifier table attack resistance. a footer.

Thus, the proposed scheme displayed more security features and a higher level of resistance than the other recently proposed schemes.

4. Performance Analysis

This section provides estimates of the performance of the proposed scheme, as well as comparing its costs, in terms of the required storage size, running time, and intercommunication, with other authentication schemes that have recently been proposed [13–16,18]. The estimated costs of the running time and intercommunication were computed for the authentication and key agreement phase, whereas the cost of the storage size needed was computed for the agriculture professional, cluster head, and sensor registration phases. In order to carry out accurate and reasonable comparisons, the following values were assumed [10,17,19,21]:

(1) The size of sequential numbers, security codes, random numbers, passwords, and identities were set to be 128 bits; (2) the output of the used hash functions was equal to 160 bits; (3) the input/output of the encryption/decryption functions were multiples of 128 bits; (4) the running times to execute the fuzzy extractor generating function (Tfe) was set as 0.0171 s; (5) SHA-1 was used as the hash function used, wherein the running time (Th) was 0.00032 s; and (6) the AES cryptographic function was used, with running time (TE/D) of 0.0056 s.

4.1. Storage Size Cost Analysis

Reducing the storage size that is needed, whether for the smart cards of the users or sensor nodes memories, is considered an important issue in such systems. Thus, the storage

size costs for the smartcards (i.e., SCi and SCj) and agriculture sensor were calculated for the proposed scheme, as well as for other recently proposed schemes [13–16,18], as illustrated in Table 10.

Table 10. Storage size cost analysis.

Schemes	Smartcard/(SCi)	Smartcard/(SCj)	Sensor
[13]	608 bits	NA	288 bits
[14]	768 bits	NA	288 bits
[15]	864 bits	NA	256 bits
[16]	736 bits	NA	800 bits
[18]	768 bits	NA	288 bits
Proposed scheme	736 bits	736 bits	256 bits

It should be noted that the size of the cryptographic functions was omitted from the calculations in order to simplify the analysis. In the proposed scheme, the storage size cost for both SCi and SCj is (736) bits, while the storage size cost for the agriculture sensor is (256) bits. Table 10 shows that the proposed scheme required the least storage size cost for agriculture sensors. Furthermore, the storage size cost that is required for the smart cards in the proposed scheme is only greater than the scheme presented in [13] but was equal to or less than those of other schemes.

4.2. Intercommunication Cost Analysis

The intercommunication cost can be estimated using the total intercommunication size of the exchanged messages between the elements of the system throughout the authentication and key agreement phase. The total intercommunication costs of the proposed scheme and other recently proposed schemes [13–16,18] are illustrated in Table 11.

Table 11. Intercommunication cost analysis.

Schemes	M1	M2	M3	M4	M5	Total
[13]	896	608	320	800	NA	2624 bits
[14]	1088	1248	578	800	NA	3714 bits
[15]	704	800	448	320	288	2560 bits
[16]	768	960	608	800	NA	3136 bits
[18]	1088	1248	578	800	NA	3714 bits
Proposed scheme	672	672	672	544	288	2848 bits

The total intercommunication cost of the proposed scheme was (2848) bits. Wherein Table 11 shows that the needed intercommunication cost for proposed authentication was only greater than that of the authentication schemes in [13,15] but less than other authentication schemes.

4.3. Running Time Cost Analysis

In this section, the estimated running time costs of the authentication phases are calculated in order to conduct a comparison between the proposed scheme and other recently proposed authentication schemes [13–16,18]. For each element of the system, the number of cryptographic functions was determined, as shown in Table 12. Furthermore, the total running time costs were calculated for the sensor node, as shown in Table 13. Finally, the total running time costs were calculated for all elements of the system, as shown in Table 14.

Table 12. Cryptographic functions in each element of the system.

Schemes	Agriculture/Healthcare Professional	AuC/GWN	Cluster Head/Access Point	Sensor
[13]	12Th	16Th	N/A	6Th
[14]	2TED + 11Th	3TED + 16Th	N/A	TED + 6Th
[15]	1Tfe + 15Th	12Th	N/A	17Th
[16]	10Th	17Th	N/A	7Th
[18]	2TED + 6Th	3TED + 8Th	2TED + 5Th	1TED + 5Th
Proposed scheme	2TED + 4Th	4TED + 7Th	2TED + 8Th	7Th

Table 13. Total running time in sensor node.

Schemes	Sensor	Cost (s)
[13]	6Th	0.00192 s
[14]	TED + 6Th	0.00752 s
[15]	17Th	0.00544 s
[16]	7Th	0.00224 s
[18]	1TED + 5Th	0.00720 s
Proposed scheme	7Th	0.00224 s

Table 14. Total running time costs.

Schemes	Total Functions	Cost (s)
[13]	34Th	0.01088 s
[14]	6TED + 33Th	0.04416 s
[15]	1Tfe + 30Th	0.02670 s
[16]	34Th	0.01088 s
[18]	8TED + 24Th	0.05248 s
Proposed scheme	8TED + 26Th	0.05712 s

The sensor node is considered a device with poor computational capabilities. Therefore, the authentication schemes that contain cryptographic functions, except the hash functions such as those in [14,18], cannot be employed in such systems. The results in Table 13 illustrate that the proposed scheme had a greater total running time cost on the sensor side than the authentication scheme proposed in [13] but was equal to or better than the other schemes.

It should be pointed out here that the proposed scheme contains two authentication phases (i.e., the authentication and key agreement phase and the authentication phase of the agriculture sensor), while the other authentication schemes [13–16,18] used just one authentication phase. This is why the total running time cost of the proposed scheme was greater than that of other authentication schemes, as can be seen from Table 14.

Besides, authentication schemes should not use only a one-way hash function to achieve a high level of security for such unattended systems. Table 14 illustrates that the proposed scheme can be applied in irrigation systems, wherein each element in the system has a reasonable execution time.

5. Conclusions

In order to overcome the existing security concerns of smart IoT irrigation systems, this paper proposed a secure authentication scheme based on Diffie–Hellman key agreement techniques over WSNs. The proposed scheme possesses attractive security features that allow for its use in such systems, wherein fully mutual authentication, full anonymity, and fully perfect forward secrecy were achieved. The BAN logic model was applied to validate the mutual authentication between the elements of the system throughout the authentication phases. According to the possible attack types scenarios that were assumed

and discussed, the demonstrated security level of the proposed scheme shows that it not only provides attractive security features but also can resist well-known types of attacks. Furthermore, compared with other recently proposed authentication schemes, the proposed scheme had the highest level of security in terms of the security feature achievements and resistance to the well-known attacks. A performance analysis demonstrated that the proposed scheme has a minimal cost in terms of storage size and also has a suitable level of intercommunication and running time costs, compared with other recently proposed authentication schemes. Finally, the proposed scheme is applicable for use in smart IoT irrigation systems in order to remotely monitor the actual amount of water required to irrigate the crops.

Funding: The author received no specific funding for this study.

Acknowledgments: The author expresses his gratitude to all members of the Computer and Information Sciences College at Jouf University for their support.

Conflicts of Interest: The author declares no conflict of interest to report regarding the present study.

References

1. Ahmed, I.; Akash, M.S. IoT based smart irrigation system at university of Chittagong, Bangladesh. *Int. J. Comput. Appl.* **2020**, *176*, 39–45. [\[CrossRef\]](#)
2. Munir, M.S.; Bajwa, I.S.; Ashraf, A.; Anwar, W.; Rashid, R. Intelligent and smart irrigation system using edge computing and IoT. *Complexity* **2021**, *2021*, 6691571. [\[CrossRef\]](#)
3. Manimegalai, V.; Judy, A.L.; Gayathri, A.; Ashadevi, S.; Mohanapriya, V. Smart irrigation system with monitoring and controlling using IoT. *Int. J. Eng. Adv. Technol. (IJEAT)* **2020**, *9*, 1373–1376. [\[CrossRef\]](#)
4. Naik, P.; Kumbi, A.; Katti, K.; Telkar, N. Automation of irrigation system using IoT. *Int. J. Eng. Manuf. Sci.* **2018**, *8*, 77–88.
5. Rawal, S. IoT based smart irrigation system. *Int. J. Comput. Appl.* **2017**, *159*, 7–11. [\[CrossRef\]](#)
6. Goap, A.; Sharma, D.; Shukla, A.K.; Krishna, C.R. An IoT based smart irrigation management system using Machine learning and open source technologies. *Comput. Electron. Agric.* **2018**, *155*, 41–49. [\[CrossRef\]](#)
7. García, L.; Parra, L.; Jimenez, J.M.; Lloret, J.; Lorenz, P. IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. *Sensors* **2020**, *20*, 1042. [\[CrossRef\]](#)
8. Hsu, W.-L.; Wang, W.-K.; Fan, W.-H.; Shiao, Y.-C.; Yang, M.-L.; Lopez, D.J. Application of internet of things in smart farm watering system. *Sens. Mater.* **2021**, *33*, 269–283. [\[CrossRef\]](#)
9. Gupta, M.; Abdelsalam, M.; Khorsandroo, S.; Mittal, S. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access* **2020**, *8*, 34564–34584. [\[CrossRef\]](#)
10. Nashwan, S. AAA-WSN: Anonymous access authentication scheme for wireless sensor networks in big data environment. *Egypt. Inform.* **2021**, *22*, 15–26. [\[CrossRef\]](#)
11. Nashwan, S. SAK-AKA: A secure anonymity key of authentication and key agreement protocol for LTE network. *Int. Arab. J. Inf. Technol. (IAJIT)* **2017**, *14*, 790–801.
12. Yu, B.; Li, H. Anonymous authentication key agreement scheme with pairing-based cryptography for home-based multi-sensor Internet of Things. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719879379. [\[CrossRef\]](#)
13. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [\[CrossRef\]](#)
14. Ali, R.; Pal, A.K.; Kumari, S.; Sangaiah, A.K.; Li, X.; Wu, F. An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring. *J. Ambient. Intell. Humaniz. Comput.* **2018**, *2018*, 1–22. [\[CrossRef\]](#)
15. Shuai, M.; Liu, B.; Yu, N.; Xiong, L. Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks. *Secur. Commun. Netw.* **2019**, *2019*, 8145087. [\[CrossRef\]](#)
16. Fotouhi, M.; Bayat, M.; Das, A.K.; Far, H.A.; Pournaghi, S.M.; Doostari, M.A. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Comput. Netw.* **2020**, *177*, 107333. [\[CrossRef\]](#)
17. Nashwan, S. An End-to-End authentication scheme for healthcare IoT systems using WMSN. *CMC-Comput. Mater. Contin.* **2021**, *68*, 607–642. [\[CrossRef\]](#)
18. Ali, A.; Pal, A.K.; Kumari, S.; Karuppiyah, 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. [\[CrossRef\]](#)
19. Xiong, L.; Peng, D.; Peng, T.; Liang, H.; Liu, Z. A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks. *Sensors* **2017**, *17*, 2681. [\[CrossRef\]](#)
20. Nashwan, S.; Alshammari, B.M. Formal analysis of MCAP protocol against replay attack. *Br. J. Math. Comput. Sci.* **2017**, *22*, 1–14. [\[CrossRef\]](#)
21. Nashwan, S.; Nashwan, I.I.H. Reducing the overhead messages cost of the SAK-AKA authentication Scheme for 4G/5G mobile networks. *IEEE Access* **2021**, *9*, 97539–97545. [\[CrossRef\]](#)