New Efficient PUF-Based Authentication Protocol for IoT-Driven Smart Agriculture

Souhayla Dargaoui [1], Mourade Azrour [1,*], Ahmad El Allaoui [1], Azidine Guezzaz [2], and Mohammad Ali A. Hammoudeh [3]

[1] IMIA Laboratory, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Errachidia, Morocco
[2] Technology Higher School Essaouira, Cadi Ayyad University, Essaouira, Morocco
[3] Department of Information Technology, College of Computer, Qassim University, Buraydah, Saudi Arabia
Email: s.dargaoui@edu.umi.ac.ma (S.D.); mo.azrour@umi.ac.ma (M.A.); a.elallaoui@umi.ac.ma (A.E.A.); a.guezzaz@uca.ma (A.G.); maah37@qu.edu.sa (M.A.A.H.)
*Corresponding author

***Abstract*—Smart agriculture is one of the most important applications of Internet of Things (IoT) technology. However, IoT devices are used outdoors, which increases their vulnerability to cyberattacks. To secure data collected and exchanged in IoT-based smart agriculture, this paper provides a new enhanced mutual authentication protocol based on the Elliptic Curve Cryptography (ECC) and Physical Unclonable Function (PUF). The proposed protocol uses a one-way hash function and random numbers to support basic security features. Moreover, to secure the user's data against brute-force attacks and password guessing, we used the Honey Encryption algorithm. Formal and informal security analyses of our scheme highlight its performance and prove that the proposed scheme resists well-known attacks, such as the denning-ssaco attack, smart card loss, Gateway (GWN) bypassing, and Man-In-The-Middle (MITM) attack. Further, our scheme comparison with the state-of-the-art schemes shows that it requires acceptable computation and communication costs, less than 5 ms, regarding its advantages over existing authentication schemes and the security level it ensures.**

***Keywords*—Internet of Things (IoT), smart agriculture, security, mutual authentication, Elliptic Curve Cryptography (ECC)**

I. Introduction

Agriculture is one of the extremely crucial backbones of human society, it is the principal economic sector that ensures food security. According to Ref. [1], agriculture industries account for 6.4% of the economic production of the world with a total of $5,084,800 million. In 2021, the statistics show that in developing countries such as India, the contribution of agriculture toward the economy was around 20.2%. However, most of the agricultural customs are based on premonition and the experiences of previous farmers. As a result, the control on agricultural productivity quantitatively and qualitatively has been very minimal; in turn, the economic gain has been limited, even with the big efforts of farmers. Present approximations of the United Nations (UN) exhibit that almost 690 million or 8.9% of the world's population are hungry. Additionally, by 2030, the number of people impacted by hunger could exceed 840 million. The global population is increasing continuously, it is expected to reach 10 billion by the end of 2050. To ensure the needs of this increasing population, the Food and Agriculture Organization (FAO) declares that food production must be increased by nearly 70% in 2050 [2, 3]. On the other hand, natural resources are progressively declining, agricultural lands are depleting, and the rise of some natural phenomena, including flooding, global warming, salinization, and some diseases such as COVID-19, exacerbate the food security problems in the world. Considering all these reasons, the usual agriculture sector based on traditional mechanics has become obsolete and unable to secure the world's population's needs [4].

To enhance farming production, the agriculture field has incorporated several emerging technologies, including data analysis, cloud computing, and the Internet of Things (IoT) [5–7]. Thanks to Wireless Sensor Networks (WSN), which collect and upload data to the cloud, the monitoring and control of the environment characteristics have been enabled. Accordingly, IoT has completely changed the face of traditional agriculture by providing various IoT-driven smart farming applications. Precision farming is a technique that may help forecast the most productive crop for a specific land and the farming operations by analyzing the collected data. Soil management determines several soil measures, including pH and soil moisture, which are highly useful in soil fertilizer requirements identification. Moreover, IoT provides automated irrigation systems guided using some soil parameter thresholds to prevent water waste and enhance the yield quality by timely irrigation [4–13].

Since the farmer's decisions are based on the collected data, this last performs a crucial task in IoT-driven smart agriculture. However, the nature of the public canals used to exchange data makes smart farming susceptible to enormous attacks, including denial of services, man-in-the-middle, impersonation attacks, and node capture

Manuscript received October 22, 2024; revised November 16, 2024; accepted December 23, 2024; published April 27, 2025.

attacks [14, 15]. As a result, adversaries may save and change prominent information upon transmission and use the IoT devices to destroy crop fields entirely. To secure the communication between the users and IoT devices, authentication and key agreement are the most promising solutions. Moreover, authentication schemes must ensure some security features, such as anonymity and unlinkability [16–18].

The literature review shows that most existing authentication schemes may not provide unlinkability, key secrecy, and perfect forward secrecy, nor resist node capture, Denial of Service (DoS) attack, stolen verifier, denning-ssaco attack, and Gateway bypassing [19–22]. More than that, the IoT devices used outdoors in smart agriculture environments increase the network's vulnerability to physical attacks.

To improve security in such environments, we exploited the Elliptic Curve Cryptography (ECC), Physical Unclonable Function (PUF), and usage of biometric identity to provide an enhanced authentication scheme for IoT-based smart agriculture. Fingerprint scanning is a biometric technology that begins with the capture of images using a variety of methods, such as optical, capacitive, or ultrasonic sensors, then followed by image enhancement to improve quality and the extraction of features to identify unique patterns such as minutiae. The extracted features are converted into a secure biometric template, which can be stored locally or on a centralized server with encryption to protect sensitive data. To withstand spoofing attacks (where unauthorized users seek to mimic fingerprints using materials such as silicone), modern systems use presence detection techniques, such as pulse or temperature detection, as well as multi-factor authentication and sophisticated algorithms that analyze complex details. This combination of processes and security measures guarantees efficient, reliable identity verification in a wide range of applications.

The main contributions of our paper are detailed as follows:

- We propose a robust authentication scheme for IoT authentication. The proposed scheme is based on ECC, PUF, Fuzzy Extractor, and Honey-list.
- Informal and formal security analyses, using the ProVerif tool, are performed to demonstrate the security of the proposed scheme.
- We compare our scheme with state-of-the-art schemes based on security features and computation and communication costs to prove its efficiency.

II. Related Works

Considering the importance of the data exchanged in IoT smart agriculture networks, it is critical to execute a robust authentication scheme between communicating devices and users to ensure the security and confidentiality of the transmitted data. To strengthen authentication in such networks, several authentication schemes have been proposed based on different cryptography methods [23–32]. Shadi Nashwan presents an IoT authentication scheme based on hash functions and Diffie-Hellman for irrigation systems [33]. The security examination shows that the proposed scheme can resist various well-known threats. The formal analysis of the provided protocol has been offered using The Burrows–Abadi–Needham (BAN) logic. Compared with other related schemes, the scheme provided by Shadi seems to be secure and practical, considering its computation cost, communication cost, and storage cost. Mirsaraei *et al*. [34] provided an ECC authentication protocol for blockchain-based IoT environments using three authentication factors called defense-in-depth. The AVISPA tool was used to perform the formal analysis. Finally, they prove that their scheme is the best regarding the security ensured and the cost of computation and communication. Another ECC-based authentication protocol has been proposed by Abduljabbar *et al*. [3] to secure precision agriculture. They prove the resistance of their scheme against well-known attacks such as node capture attacks, password guessing, replay attacks, impersonation attacks, and denial of service attacks. Additionally, the proposed scheme ensures numerous security features, including anonymity and untraceability. The formal analysis of the scheme was carried out using BAN logic. The comparison of the presented scheme and some other schemes shows that it reduces the computation cost by 14.67% and the communication cost by 18%. Moreover, the provided scheme has a 35.29% enhancement in ensured security services.

To secure data integrity and authenticity, Saleh *et al*. [35] designed an authentication protocol for IoT devices in smart agriculture environments exploiting Constrained Application Protocol (CoAP). The proposed scheme proves its robustness and resistance to tampering. Using the Contiki NG simulation tool, the authors illustrate the efficiency of their scheme in terms of computation and communication costs and resistance to well-known attacks such as replay attacks and DoS attacks. To remedy the holes in recent authentication approaches, Hassan *et al*. [24] offered an authentication scheme based on identity and hyperelliptic curve cryptography for precision agriculture built on IoT networks. The proposed scheme reduced the computational and communication costs. To perform a formal security analysis of the proposed approach, the authors use the Scyther tool. In addition, they proved the resistance of their scheme against security threats and its efficiency compared to the state-of-the-art approaches. Ali *et al*. [36] offered a symmetric encryption-based authentication protocol for smart agriculture built on WSNs. To validate and simulate the proposed approach, the authors used BAN logic and the AVISPA tool. Then, they used Random Oracle to perform the formal analysis of the proposed approach. Finally, they prove the resilience of their scheme against well-known attacks and its utility in agriculture monitoring.

Khalid *et al*. [37] offered a new authentication approach for WSN-based agriculture using symmetric encryption and a fuzzy extractor algorithm. Using the AVISPA tool, they validate the formal security

examination of the proposed scheme. Additionally, they ensure the resilience of their scheme against capture node, replay attacks, and enormous well-known threats thanks to informal security analysis. In the end, they offered the examination results of the proposed approach and some other related ones. The examination result illustrates the efficiency of the proposed scheme in terms of computational and communication costs of their scheme compared to existing approaches. Vangala *et al*. [25] offered a robust authentication and key agreement scheme based on blockchain technology and ECC to secure exchanged data in IoT-driven smart farming, AgroMobi Block. The proposed scheme optimizes the high cost of the state-of-the-art authentication schemes built on Blockchain. The authors used the AVISPA tool to perform the formal security analysis. Additionally, the informal security analysis demonstrates the resilience of AgroMobi Block contrary to man-in-the-middle attacks, impersonation attacks, and other security threats. In addition, using three authentication factors, Vangala *et al*. [38] presented an ECC-based authentication approach. The security of the proposed approach was tested using the Real or Random (ROR) model. Then, the informal security analysis and the validation using the AVISPA tool prove the resistance of the offered approach against several security threats. Moreover, the authors demonstrate that their approach ensures various security features, including anonymity and untraceability. The examination of the proposed approach and some other related ones enables the authors to prove that the proposed scheme offers more security, requiring less computational and communication costs.

All in all, we may classify IoT authentication schemes into three classes. Hash function-based authentication does not require high computational power, but it cannot ensure high security. Blockchain-based authentication offers high security; nevertheless, it needs a lot of computation power and expensive storage costs. ECC-based authentication ensures high security, reducing the need for computation power and memory storage [39].

III. Preliminaries

A. Hash Function

A hash function is a mathematical function that gives a fixed length output for each input. It is irreversible and Collision-resistant, which means that it is impossible to recover the original input from a hash value, and very similar inputs cannot have correlated hash values. These two features make the hash function the best solution to secure long-term keys in our scheme.

B. Physical Unclonable Function

PUF is a one-way function that extracts some unique information by analyzing the nano-scale level of the IoT device. Using an input called a challenge, the PUF generates an output called a response, which could be unpredictable and repeatable, R = PUF (C). It enables enhanced security compared to other hardware security mechanisms by the fact that the root keys of devices are never stored in persistent memory, making them unclonable and invisible to attackers. Additionally, it does not require dedicated security hardware or expensive components, so it comes at a very low cost and is compatible with limited resources IoT-driven agriculture sensors [40, 41]. Considering their resiliency to environmental variations, ring oscillator PUFs may be the best choice for smart agriculture applications. Appendix A presents a simple comparison between some PUFs.

Environmental aging may significantly impact the reliability of PUF. Over time, environmental factors can alter the physical properties of the materials or increase error rates in the challenge-response pairs, leading to degradation in the unique characteristics that PUFs rely on for security and identification. Further, PUFs may become more susceptible to side-channel attacks. Choosing materials that are more resistant to environmental aging can improve the long-term reliability of PUFs. Implementing error correction codes can help correct errors in the challenge-response pairs generated by aging PUFs. Regular Recalibration and redundant PUF Structures may also be potential mitigation strategies. Table I shows some PUF variants.

TABLE I. PUF Variants Comparison

PUF	Advantages	Disadvantages
SRAM PUFs	Ease of Integration High Uniqueness Low Overhead	Environmental Sensitivity Limited Challenge-Response Space
Ring Oscillator PUFs	High Uniqueness Robustness Flexible Challenge-Response Generation	Complexity Power Consumption
Delay PUFs	High Security Versatility	Environmental Sensitivity Complexity
Optical PUFs	High Security Unique Physical Characteristics	Implementation Complexity Environmental Sensitivity

C. Honey-List

Generally, honey encryption is used to secure IoT devices against brute-force attacks and offline-guessing attacks. If an attacker tries to access a network using the wrong password, HE generates fake responses that look correct, “Honey words”, to feed the attacker and send an alert to the user [42, 43]. In our scheme, during the login and authentication phase, the honey list enables the Gateway (GWN) to detect intrusion by monitoring the attacker’s login source. Moreover, when the number of elements in the honey list exceeds the predefined threshold, the GWN stops the session and alerts the user to change the password.

D. Fuzzy Extractor

Considering that physical unclonable functions are susceptible to environmental factors such as temperature variations and circuit noise, they may respond differently to the same challenge [44–46]. A fuzzy extractor produces the same output for nearly identical noisy input sources. To do this, there are two procedures: Generation procedure (Gen), which may extract an equably diffused value R (key), and a public reconstruction data P from a noisy signal w. Reproduction procedure (Rep): which can

reproduce R from a value w0 and P if w0 is close enough to w. As a result, we can have the same result for noisy PUF output.

IV. Materials and Methods

A. Research Methodology

Several steps have been taken to construct our authentication scheme. Firstly, we conducted a state-of-the-art study that explored the latest published authentication protocols for IoT environments. Several classes of authentication schemes have been defined based on the cryptography techniques and authentication factors. This study offered different cryptography methods that can be used to build an authentication scheme, such as elliptic curve cryptography, symmetric and asymmetric encryption, Blockchain, hash functions, and research directions that need more improvement.

Secondly, we analyzed the features of each authentication class based on the security level offered and the requirements in terms of energy, computational power, and storage. By the end of this step, we chose ECC to build our authentication scheme because it provides a high-security level and low latency, and does not require either big computation power or significant memory storage.

Thirdly, we constructed our authentication scheme, as presented in Section IV.C. Then, we examined our scheme using the ProVerif tool. Finally, we discussed the requirements and the security features provided by our scheme compared to other schemes.

B. System Model

Generally, IoT-based smart agriculture systems consist of three kinds of entities: consumers, IoT sensors, and a trusted Gateway Node. To securely communicate, the GWN authenticates the consumers and IoT sensors and helps both to authenticate mutually. In the setup and registration phases, a secure channel is used to perform the registration of all users and sensors. By the end of these phases, the GWN stores the registration information in the IoT device's memory and transfers a smart card to each user. After that, the IoT devices in the field collect the environmental data. When a user wants to access the data collected, it requests the GWN to perform authentication and session key agreement between the user and the IoT sensor. Then, the user sends requests and receives data from the sensors using the established session key. Fig. 1 illustrates the architecture of the IoT-driven smart agriculture systems.

C. Threat Model

Referring to the Dolev-Yao threat model [47], we present an adversary's capabilities as follows:

- The attacker could spy on all messages transmitted via an open channel.
- The attacker can modify, insert, replay, and forward spy messages.
- If the attacker gets a user's smart card, HE can get all the data stored in the smart card.
- If the attacker captures a sensor node, it can get all the data stored in that sensor.
- The attacker may be a legitimate user.

Nevertheless, we assume that the GWN in the proposed scheme is a trusted node and is not compromised under any circumstances.

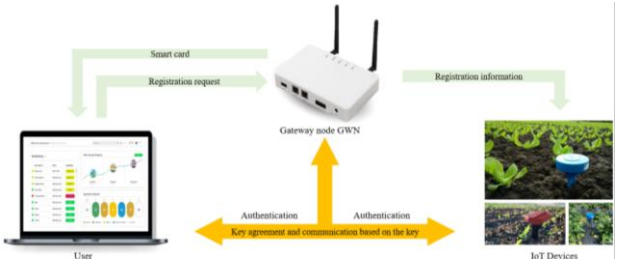

Fig. 1. Network model.

D. The Proposed Approach

1) Initialization phase

In this phase, the gateway selects a random number K_c as the private key, selects an elliptical curve generator point P, and chooses a one-way hashing function h(.). Then, the GWN computes its public key $PK_c = K_c.P$. Finally, it publishes the generation and reproduction algorithms of the fuzzy extractors Gen(.) and Rep(.), h(.), PK_c, and P. Table II shows the notation used in this paper.

TABLE II. Notations

Notation	Description
ID_i	User identity
HID	User pseudo-identity
$THID$	Temporary user identity
ID_j	Sensor identity
HID_j	Sensor pseudo-identity
PW_i	User password
GWN	Gateway
ID_c	Gateway identity
K_u	User secret key
PK_u	User public key
K_C	Gateway secret key
PKc	Gateway public key
K_s	Sensor secret key
PKs	Sensor public key
$h(.)$	One-way hashing function
SK	Session key
Gen, Rep	Generation and reproduction algorithm of fuzzy extractor
$R_u, R_s, R_{c1}, R_{c2}, r_i, r_j$	Random nonces
T_i	Timestamp
B_{ioi}	User biometric model
C_j, B_j	The challenge/response pair
PUF	Physical Unclonable Function
$\oplus$	Xor procedure
$\|\|$	String concatenation procedure
P	The generator point on the curve

2) Device registration phase

This phase requires three steps. In the first step, the IoT device picks an identity ID_j, challenges C_j, and a random number K_s. Then it computes B_j as in Eq. (1), generates (R_j, P_j) as in Eq. (2), and computes the pseudo identifier HID_j, as in Eq. (3). Then, the device sent a registration request {HID_j, C_j} to the GWN.

In the second step, after revising the registration request the GWN selects a random number r_j and computes K_{sc} as in Eq. (4), T_s as in Eq. (5), and D as in Eq. (6). Finally, the GWN stores {D, C_j, K_{sc}, T_s} in its memory, and sends the response {K_{sc}, T_s} to the sensor.

In the third step, the sensor computes K_s* as in Eq. (7), and PK_s as in Eq. (8), then it stores {ID_j, K_s*, Pj, T_s, K_{sc}} in its memory and publishes PK_s.

$$B_j = PUF(C_j) \quad (1)$$

$$(R_j, P_j) = Gen(B_j) \quad (2)$$

$$HID_j = h(IDj||CR_j) \quad (3)$$

$$Ksc = h(HID_j||Kc) \quad (4)$$

$$Ts = h(Ksc||r_j) \quad (5)$$

$$D = HID_j \oplus h(Kc||r_j) \quad (6)$$

$$Ks *= Ks \oplus h(HID_j||R_j) \quad (7)$$

$$PKs = Ks.P \quad (8)$$

3) User registration phase

This phase requires three steps:

First step: The user U selects identity IDi and password PW_i, scans his fingerprints, and selects randomly two numbers r1 and K_u to compute the pseudo identity HID as in Eq. (9). Then, he computes H as in Eq. (10) to secure HID. Subsequently, he generates *Ri* and *Pi*, as in Eq. (11), exploiting a fuzzy extractor. Afterward, he computes HPW and W as in Eq. (12) and Eq. (13). Ultimately, the user sends the registration request {*HID*, *HPW*} to the GWN.

Second step: the entrance selects a random number r_i, and *THID*. Then, it computes *A* in Eq. (14) and *I* in Eq. (15). Finally, the GWN stores {r_i, *THID*, *I*} in its database and sends the response {*A*, *THID*} to the user.

Third step: the user computes *B* as in Eq. (16), PK_u as in Eq. (17), and K_u* as in Eq. (18). Thereafter, the user publishes PK_u and stores {K_u*, *H*, P_i, *W*, *B*, *Rep*(.), *h*(.), *P*, *THID*} in a smart card.

$$HID = h(IDi||r1) \quad (9)$$

$$H = HID \oplus h(IDi||PWi) \quad (10)$$

$$(Ri, Pi) = Gen(Bioi) \quad (11)$$

$$HPW = h(PWi||Ri) \quad (12)$$

$$W = HPW \oplus h(IDi||PWi) \quad (13)$$

$$A = h(HID||Kc||HPW) \quad (14)$$

$$I = HID \oplus h(Kc||ri) \quad (15)$$

$$B = A \oplus h(IDi||PWi) \quad (16)$$

$$PKu = Ku.P \quad (17)$$

$$Ku *= Ku \oplus h(IDi||Pwi||Ri) \quad (18)$$

4) Login and authentication phase

In this phase, the communication is made through a public channel. Firstly, the user logs in using the smart card and its credentials, then it sends an authentication request to the *GWN*, which verifies its authenticity and sends an authentication request to the IoT device. After receiving and authenticating the IoT device response, the *GWN* sends an authentication request to the user. At the end of this phase, the user and the *GWN* update the *THID* to *THID* new, and a session key is established between the user and the IoT device. Later, using the session key, secure data exchanges can be ensured through public channels. Fig. 2 illustrates the login and authentication process.

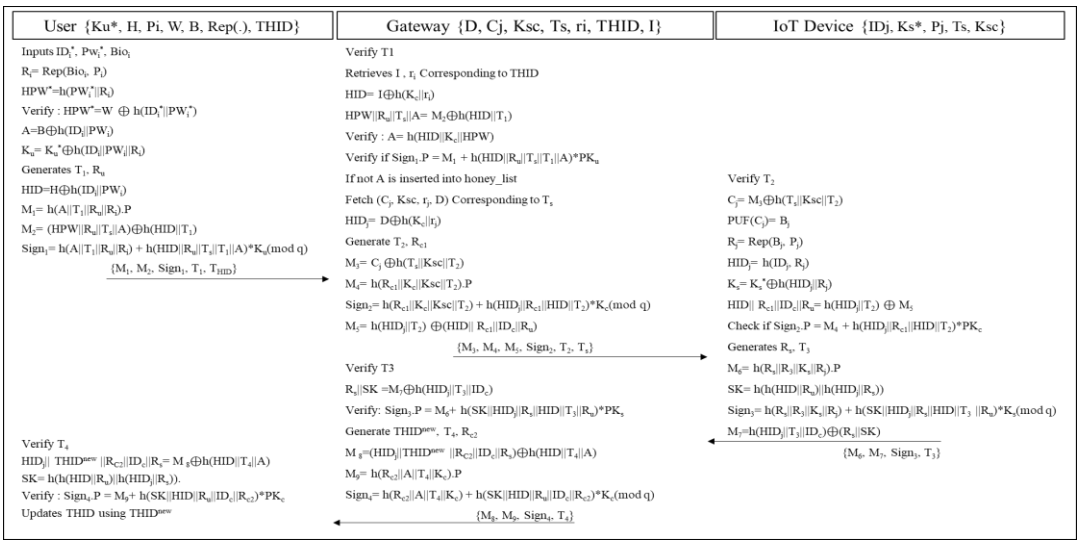

Fig. 2. Login and authentication phase.

First step: the user inserts the smart card and enters its credentials ID_i*, PW_i*, and biometric B_{ioi}, reconstructs the *Ri* using the Fuzzy extractor as in Eq. (19), calculates *HPW** as in Eq. (20) and verifies the validity of *HPW** as in Eq. (21). If so, the user recovers A as in Eq. (22), *HID* as in Eq. (23), and K_u as in Eq. (24) from the smart card. Then, he generates *T*1, R_u, and constructs the authentication request {*M*1, *M*2, *Sign*1, *T*1, *THID*} as in Eqs. (25)–(27).

Finally, the user sends the authentication request to the *GWN*.

$$Ri = Rep(Bioi, Pi) \tag{19}$$

$$HPW *= h(PWi * ||Ri) \tag{20}$$

$$HPW *= W \oplus h(IDi * ||PWi *) \tag{21}$$

$$A = B \oplus h(IDi||PWi) \tag{22}$$

$$HID = H \oplus h(IDi||PWi) \tag{23}$$

$$Ku = Ku * \oplus h(IDi||PWi||Ri) \tag{24}$$

$$M1 = h(A||T1||Ru||Ri).P \tag{25}$$

$$M2 = (HPW||Ru||Ts||A) \oplus h(HID||T1) \tag{26}$$

$$Sign1 = h(A||T1||Ru||Ri) + h(HID||Ru||Ts||T1||A) \times Ku(mod\ q) \tag{27}$$

Second step: as soon as receiving the user request, the *GWN* checks the freshness of the timestamp *T*1, retrieves *HID* corresponding to *THID*, recovers *HPW*, R_u, T_s, and A from *M*2, and verify the authenticity of the user using Eqs. (28) and (29).

If the equations are not the same the *GWN* inserts *A* into honey_list. Or else the *GWN* retrieves (C_j, K_{sc}) Corresponding to T_s, Generate *T*2, R_{c1}, and computes the authentication request {*M*3, *M*4, *M*5, *Sign*2, *T*2, T_s} as in Eqs. (30)–(33). Then, the GWN sends the request to the IoT device.

$$A = h(HID||Kc||HPW) \tag{28}$$

$$Sign1.P = M1 + h(HID||Ru||Ts||T1||A) \times PKu \tag{29}$$

$$M3 = Cj \oplus h(Ts||Ksc||T2) \tag{30}$$

$$M4 = h(Rc1||Kc||Ksc||T2).P \tag{31}$$

$$Sign2 = h(Rc1||Kc||Ksc||T2) + h(HIDj||Rc1||HID||T2) \times Kc(mod\ q) \tag{32}$$

$$M5 = h(HIDj||T2) \oplus (HID||\ Rc1||IDc||Ru) \tag{33}$$

Third step: after receiving the *GWN* request, the device verifies the freshness of *T*2, recovers the challenge C_j from *M*3, and recovers the *HID*, R_{c1}, ID_c, and R_u as in Eqs. (34)–(37).

Then, the device checks the authenticity of the *GWN* as in Eq. (38). If it is legitimate, the device generates R_s, *T*3 and computes the response message {*M*6, *M*7, *Sign*3, *T*3}, as in Eq. (39)–(42). Finally, the device sends the response to the *GWN*.

$$PUF(C_j) = B_j \tag{34}$$

$$R_j = Rep(B_j, P_j) \tag{35}$$

$$HID_j = h(ID_j, P_j) \tag{36}$$

$$HID||Rc1||IDc||Ru = h(HID_j||T2) \oplus M5 \tag{37}$$

$$Sign2.P = M4 + h(HID_j||Rc1||HID||T2) \times PKc \tag{38}$$

$$M6 = h(Rs||R3||Ks||R_j).P \tag{39}$$

$$SK = h(h(HID||Ru)||h(HID_j||R_s)) \tag{40}$$

$$Sign3 = h(Rs||R3||Ks||R_j) + h(SK||HID_j||Rs||HID||T3||Ru) \times Ks(mod\ q) \tag{41}$$

$$M7 = h(HID_j||T3||IDc) \oplus (R_s||SK) \tag{42}$$

Forth step: whenever the *GWN* acquires the device response, it verifies the validity of *T*3, recovers R_s and *SK* from *M*7, and verifies the authenticity of the IoT device as in Eq. (43). If so, the *GWN* Generates *THIDnew*, T4, R_{c2}, and constructs the response message {*M*8, *M*9, *Sign*4, *T*4} as in Eqs. (44)–(46). In the end, the GWN sends the message to the user.

$$Sign3.P = M6 + h(SK||HIDj||Rs||HID||T3||Ru) \times PKs \tag{43}$$

$$M8 = \left(HIDj\middle||THIDnew\middle||RC2\middle||IDc\middle||Rs\right) \oplus h\left(HID\middle||T4\middle||A\right) \tag{44}$$

$$M9 = h(Rc2||A||T4||Kc).P \tag{45}$$

$$Sign4 = h(Rc2||A||T4||Kc) + h(SK||HID||Ru||IDc||Rc2) \times Kc(mod\ q) \tag{46}$$

Fifth step: right after receiving the *GWN* message, the user checks the freshness of *T*4 and recovers HID_j, *THIDnew*, R_{c2}, ID_c, and R_s from *M*8. Then, the user computes the session key as in Eq. (40) and verify the authenticity of the *GWN* as in Eq. (47). Finally, the user updates *THID* to *THIDnew*.

$$Sign4.P = M9 + h\left(SK\middle||HID\middle||Ru\middle||IDc\middle||Rc2\right) \times PKc \tag{47}$$

5) Password update phase

To update the password, the user inserts the smart card in the card reader, inputs its credentials, and recovers Ri as in Eq. (19). Then, he computes *HPW** as in Eq. (20) and verifies the validity of *HPW** as in Eq. (21). If the condition is met, the user selects a new password and computes the new variables Wnew as in Eq. (48), Hnew

as in Eq. (49), *Bnew* as in Eq. (50), and K_u*new as in Eq. (51). Finally, the user updates the information stored in the smart card.

$$Wnew = h(PWinew||Ri) \oplus h(IDi||PWinew) \quad (48)$$

$$Hnew = H \oplus h(IDi||PWi) \oplus h(IDi||PWinew) \quad (49)$$

$$Bnew = B \oplus h(IDi||PWi) \oplus h(IDi||PWinew) \quad (50)$$

$$Ku * new = Ku *\oplus h\big(IDi\big||Pwi\big||Ri\big) \oplus h\big(IDi\big||PWinew\big||Ri\big) \quad (51)$$

V. Result and Discusion

A. Security Analysis

In this section, we examined the proposed scheme's security and privacy using informal and formal security analysis as described below.

1) Informal security analysis

- Mutual Authentication

Our proposed scheme ensures that the GWN authenticates the user and the IoT device and helps both to mutually authenticate. After receiving the request from the user, the GWN verifies its authenticity by checking the correctness of Eq. (29). Then, to authenticate the GWN, the IoT device checks Eq. (38). If the equation is met, the GWN is successfully authenticated. Right after receiving the device response, the GWN verifies the correctness of Eq. (43) to authenticate the device. Finally, the user checks the correctness of Eq. (47). If the equation is correct, the user successfully authenticates the GWN and the IoT device.

- Anonymity and Untraceability

When the attacker tries to extract the user identity by recovering the information exchanged through public channels, it should first extract the user's pseudo-identity HID from *M*2, *M*4, *M*5, *M*7, *Sign*1, *Sign*2, or *Sign*3. However, this is impractical because of the use of one-way hash function. In addition, if the opponent extracts the HID, it is still unable to recover the ID without knowing the parameter r1. Moreover, the adversary cannot track the user because this last has no long-term parameter, and the exchanged data of the same user changes from one session to another because of the random numbers and timestamps. For these reasons, our protocol not only secures the user's anonymity, instead it also ensures the untrace ability of users.

- Session Key Secrecy

In our authentication scheme, the adversary cannot extract the pseudo-identity of the user or the IoT device from the GWN because it is suggested trusted. In addition, the attacker may not recover random nonces R_u or R_s from the communicated information. As a result, no one can compute the session key except the GWN and the user. Hence, the suggested scheme ensures the session key secrecy.

- Impersonation Attack

To disguise itself as a legitimate user, GWN, or IoT device, the attacker must compute authentication requests of one of these entities: {*M*1, *M*2, *Sign*1, *T*1, *THID*}, {*M*3, *M*4, *M*5, *Sign*1, *T*2, T_s} and {*M*8, *M*9, *Sign*4, *T*4}, or {*M*6, *M*7, *Sign*3, *T*3}. To compute {*M*1, *M*2, *Sign*1, *T*1, *THID*}, the attacker requires Ri, A, K_u, HID, and HPW. Even if the attacker may steal the smart card, it could not recover the user's secret information without the login credentials. To construct {*M*3, *M*4, *M*5, *Sign*1, *T*2, T_s} and {*M*8, *M*9, *Sign*4, *T*4} and impersonate the GWN, the opponent needs to know the secret key K_c to recover the required parameters from the user request, which is impossible because of the trusted feature of the GWN. In addition, to compute the request {*M*6, *M*7, *Sign*3, *T*3}, the attacker needs to know R_j, *SK*, HID_j, *HID*, and K_s, which is not feasible unless knowing K_{sc}. Accordingly, the proposed scheme may withstand impersonation attacks.

- Replay Attack

In this kind of attack, the opponent tries to connect with the name of a previous entity, using authentication requests used in past sessions, so that the receiver cannot detect that the request is repeated. Our proposed scheme can resist this kind of attack by exploiting timestamps and random nonces. Once the entity receives the authentication request, it checks the freshness of the timestamp. If the attacker modifies the timestamp, it must change the request, which is not practical without knowing secret parameters.

- Node Capture

In a node capture attack, the opponent may recover the information stored in the device {ID_j, K_s*, P_j, T_s, K_{sc}}. Nevertheless, using these secret parameters, the attacker cannot compute the session key or the authentication request {*M*6, *M*7, *Sign*3, *T*3}, without the HID_j. The computation of HID_j is impractical because the attacker needs R_j, which is calculated using the unduplicated PUF function. Accordingly, the node capture attacks cannot be implied in our scheme.

- Denial of Service

Our proposed scheme ensures resistance against DoS attacks using timestamps, random nonces, and honey-list. In case an opponent tries to replay various fake authentication requests to the GWN, the requests will be rejected as soon as the GWN checks the freshness of the timestamp. Moreover, when the number of fake data in the honey list exceeds a certain threshold, the user and his smart card will be blocked until the user updates its password.

- Privileged Insider Attack

As a legitimate user, the attacker can only recover the data stored in the smart card or data exchanged via public channels. To start a new session as a legitimate user, the attacker needs to recover the credentials. However, in our scheme, we use only pseudo identity HID and a hidden form of password HPW. Suppose that the privileged insider may recover HID and HPW it is still impractical to extract the identity and the password because of the collision resilient feature of the one-way hash function. Therefore, our scheme is resilient against privileged insider attacks.

- Stolen Verifier

In case the attacker obtains the verification tables of the GWN {D, C_j, K_{sc}, T_s} {ri, THID, I}, it is still unable to compute the session key or impersonate the GWN. Without the secret key K_c, the attacker may not even recover HID or HID_j, which means that the information stored in these tables is not useful. As a result, our proposed scheme can resist stolen verifier attacks.

- Denning-Sacco

In our scheme, the session key doesn't contain any long-term key. It is computed as in Eq. (40), where R_u and R_s are random values that change every session. As a result, the attacker may not recover any long-term parameter from the previous session key. Therefore, our scheme is strongly resistant to Denning-sacco attacks.

- Smart-Card Loss

Suppose that the attacker may steal the user's smart card and recover the information stored in it {K_u*, H, P_i, W, B, Rep(.), h(.), P, THID}. In practice, it is impossible to Guess the user's identity and password simultaneously in polynomial time, and all secret parameters in the smart card are hidden using a one-way hash function, the user credentials, and private biometric parameter Ri. For all these reasons, the attacker won't be able to go through authentication.

- Password Guessing

The use of the honey list in our scheme makes it safe against online-guessing attacks since the session is blocked, and the user changes the password when the threshold of the honey list is exceeded. In addition, in case the attacker may recover the information communicated through public channels, the extract of HPW requires HID, which is also not accessible to the attacker, moreover, the PW_i is hidden in HPW using a one-way hash function. If the attacker attempts an offline-guessing attack using the data stored in the smart card, it may not extract HPW from W or PW_i from HPW. Accordingly, our scheme is resilient against password-guessing attacks.

- GWN Bypassing

To successfully perform a GWN bypassing attack, the opponent should complete the authentication process and secure the GWN tasks without informing this trusted entity. To send an authentication request to the IoT device the attacker could computes M3, M4, Sign2, and M5. However, the attacker may not compute these messages because it has no access to secret parameters such as K_c, HID, HID_j, and R_u. As a result, this kind of attack is impossible in our scheme.

- The Man in the Middle

To manipulate, eavesdrop, or intercept the information exchanged through public channels, the attacker needs some secret parameters such as R_u, K_u, IDi, and PW_i. However, none of these parameters is accessible to the attacker. Accordingly, the man-in-the-middle attacks are not applicable in our scheme.

- Side-channel

Side-channel attacks use indirect information coming from a system to get unauthorized access, usually by measuring elements such as time, energy consumption, electromagnetic leakage or even sound. In order to protect authentication protocols against these attacks, execution time needs to be reduced.

2) Formal security analysis

To perform a security analysis of our scheme, we used ProVerif as a security verification tool. ProVerif is an automatic verification tool known for security protocols defined in the Dolev-Yao model that enables verification of security features such as authentication. It may reconstruct attacks and run the protocol only for an unlimited number of sessions. To check the protocol using the process calculation syntax of Blanchet *et al*. [48], ProVerif may accept as inputs Horn clauses or Pi calculus codes.

The result of our analysis is provided in Fig. 3. the analysis results confirm that the mutual authentication process in our scheme was executed in order. Moreover, the certainty of the user credentials, the private key of the user, the private key of the GWN, the private key of the IoT device, and the session key are provided by our scheme.

```
Verification summary:

Query not attacker (SK[]) is true.

Query not attacker (IDi[]) is true.

Query not attacker (PWi[]) is true.

Query not attacker (Ku[]) is true.

Query not attacker (Ke[]) is true.

Query not attacker (Ks[]) is true.

Query inj-event (UAuthentication Phase(user[])) ==> inj-event (ULoginPhase(user[])) is true.

Query inj-event (GWNAuthentication (server[])) ==> inj-event (UAuthenticationPhase(user[])) is true.

Query inj-event (SNSessionKey(sensor [])) ==> inj-event (GWNAuthentication (server[])) is true.

Query inj-event (UserSessionKey(user[])) ==> inj-event (SNSessionKey(sensor[])) is true.

--------------------------------------------------------------------------------
```

Fig. 3. Obtained results.

B. Performance Examination

1) Analysis security performance

Table III shows the comparative analysis of our scheme with state-of-the-art schemes in terms of security features and resistance against attacks. Our scheme proved its resilience against well-known attacks and its support of all the security features. Nonetheless, the scheme proposed in reference [27] doesn't demonstrate that it supports key secrecy, or perfect forward secrecy, in addition, it proves the resistance only against impersonation attacks, reply attacks, insider attacks password guessing, and smart card loss attacks. Authors of reference [28] did not prove that their scheme supports perfect forward secrecy. What is more, they only justify the resistance of the scheme against impersonation attacks and smart card loss attacks. Authors of reference [29] did not prove the support of key secrecy, likewise, they did not confirm the resistance of their scheme against impersonation attacks, DoS attacks, insider attacks, stolen verifier, and Denning-ssaco attacks. The scheme in reference [30] did not prove that it offers unlinkability, key agreement, key secrecy, and perfect forward secrecy. As well it proves the resilience only against impersonation attacks, replay attacks, stolen verifier, and man-in-the-middle attacks. Authors of reference [31] did not show its resistance against Denning-ssaco attacks, smart card loss, GWN bypassing, and man-in-the-middle attacks.

TABLE III. SECURITY FEATURES AND RESISTANCE AGAINST ATTACKS

Protocol	F_1	F_2	F_3	F_4	F_5	F_6	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
[3]	√	√	√	√	√	-	√	√	√	√	√	√	-	√	√	-	√
[24]	√	√	-	√	√	√	√	√	√	√	-	-	-	-	-	-	-
[38]	√	√	-	√	√	-	√	√	√	√	√	-	-	√	√	-	√
[27]	√	√	√	√	-	-	√	√	-	-	√	-	-	√	√	-	-
[28]	√	√	√	√	√	-	√	-	-	-	-	-	-	-	√	-	-
[29]	√	√	√	√	-	√	-	√	√	-	√	-	-	-	√	√	√
[30]	√	√	-	√	-	-	√	√	-	×	-	√	-	-	-	-	√
[31]	√	√	√	√	√	√	√	√	√	√	√	√	-	√	-	-	-
Our protocol	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

Note: F_1: mutual authentication, F_2: Anonymity, F_3: unlinkability, F_4: key agreement, F_5: key secrecy, F_6: perfect forward secrecy, A_1: Impersonation attack, A_2: reply attack, A_3: node capture, A_4: DoS attack, A_5: Insider attack, A_6: Stolen verifier, A_7: Denning-ssaco attack, A_8: password guessing, A_9: smart card loss, A_{10}: GWN bypassing, A_{11}: man in the middle.

2) Computation overheads

This section provides login and authentication computation overheads of our scheme and other related schemes. Table IV summarizes our comparison results. The notations T_h, T_e, T_a, T_s, T_f, T_g, T_{puf}, T_{ps}, T_{mtp}, and T_d describe, respectively the temporal requirements of one-way hashing, ECC multiplication, ECC addition, symmetric encryption, fuzzy extractor, HECDSA signature generation/ verification, PUF execution, Elliptic curve point substruction, Map-to-point hashing, and hyperelliptic curve divisor multiplication. According to references [30] and [32], Th ≈ 0.0001 ms, Tf ≈ Te ≈ 0.442 ms, Ta ≈ 0.002 ms, T_s ≈ 0.0026 ms, Tg ≈ 3.1920, and Tpuf = 0.12 ms, Tps = 0.0115 ms, Tmtp = 5.264 ms, and Td = 0.48 ms.

In our scheme, the login and authentication phase require 36Th + 2Tf + Tpuf + 4Ta + 8Te. To compute the authentication request, verify the response of the GWN, and compute the session key, the user needs 11Th + Tf + Ta + 2Te. In addition, The GWN demands 14Th+2Ta+4Te to construct the authentication requests and authenticate the user and the IoT device. The sensor requires 11Th + Tf + Tpuf + Ta + 2Te to verify the GWN authenticity and compute the response message.

TABLE IV. COMPUTATIONAL COST COMPARISON

Scheme	User	Gateway	Sensor	total
[3]	-	-	-	Tmtp + 12 Th + 3 Te + Tps
[24]	-	-	-	2Th + 5Td
[38]	-	-	-	34 Th + 14 Te + 4Ta + Tf
[27]	8Th	11^{Th} + 1Ts	5Th + 1Ts	24Th + 2Ts
[28]	4Th + 2Tf	11Th	3Th	18Th + 2Tf
[29]	12Th + Tpuf + 3Te	12Th + Te	7Th +Tpuf + 2Te	30Th + 2Tpuf + 6Te
[30]	-	-	-	15Th + 2Tf + 4Ts + 2Tg + 6Te
[31]	11Th	15Th	4Th	30Th
Our protocol	11Th + Tf + Ta + 2Te	14Th + 2Ta + 4Te	11Th + Tf + Tpuf + Ta + 2Te	36Th + 2Tf + Tpuf + 4Ta + 8Te

Fig. 4 presents the execution time of the login and authentication phase of our schemes compared to other schemes. Our scheme demands more computational costs than most part of other schemes because of the use of signatures, ECC, fuzzy extractor, and PUF. These requirements are understandable since our scheme supports more security features and resists more attacks compared to other schemes. Moreover, our scheme requires less than 5 ms, it would show no appreciable retardation in the eyes of any human being.

To evaluate scalability, we simulated authentication latency with increasing network size. Fig. 5 illustrates the latency of our protocol for different network sizes [100, 300, 500, 1000, 3000, 5000, 10,000]. Considering that the network latency increase equals 0.1 ms per user, our protocol is scalable for 3000 users.

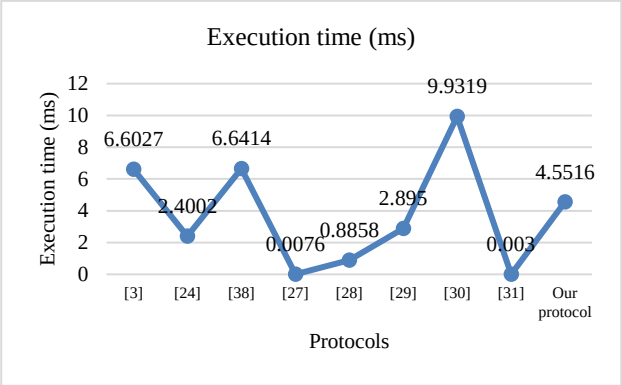

Fig. 4. Execution time of the login and authentication phases.

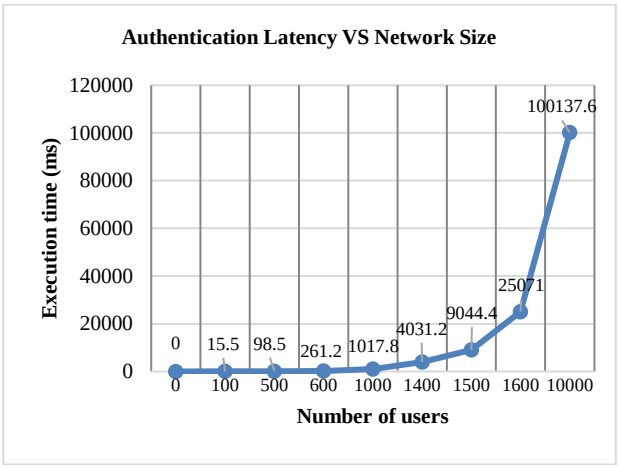

Fig. 5. Authentication latency with increasing network size.

Considering IoT devices are often battery-powered, using PUF and ECC for authentication may require high energy consumption. Several techniques, such as Genetic Algorithm (GA), Artificial Neural Network (ANN), and fuzzy logic, can be useful for optimizing energy consumption.

3) Communication overheads

The communication overhead presents the number of bits communicated during the login and authentication phase. Table V shows the number of exchanged messages and the number of communicated bits in each scheme.

TABLE V. COMMUNICATION COST COMPARISON

Protocol	Total messages	Communication cost
[3]	2	1600
[24]	1	240
[38]	4	5792
[27]	6	2000
[28]	4	2688
[29]	4	2248
[30]	3	-
[31]	4	3680
Our protocol	4	5760

Assuming the size of the random nonce, timestamp, identity, hashing function output, elliptic curve point, and symmetric encryption block, are respectively 128 bits, 32 bits, 128 bits, 256 bits, 320 bits, and 160 bits.

VI. CONCLUSION AND FUTURE WORKS

The deployment of IoT has encountered a real issue with security, which has prompted researchers to design various authentication schemes to secure exchanged data. However, most of those schemes are vulnerable to customer anonymity issues. Our paper presents a robust three-factor mutual authentication scheme based on ECC, PUF, and Fuzzy extractor. The presented scheme supports the basic security features, including authentication, user anonymity, unlinkability, key secrecy, and perfect forward secrecy. Furthermore, as mentioned in the security analysis, the proposed scheme may resist well-known attacks, including the denning-ssaco attack, smart card loss, GWN bypassing, and Man-In-The-Middle (MITM). Additionally, our scheme's comparison with the state-of-the-art schemes shows that it requires acceptable computation and communication costs regarding the security level that it ensures. As a result, our protocol may be useful for other IoT domains, such as healthcare or smart cities. To adapt it to such applications, we can integrate a cloud server to store data. To reduce the computation requirement, we can eliminate the use of PUF for domains that do not require outdoor IoT devices. However, the number of connected devices in an IoT environment is increasing continuously, which may increase the network latency and create a scalability issue.

We also believe that real experiences are important to validate any authentication protocol. However, in our study, we have just validated it under ProVerif. Hence, as a perspective for our future work, the deployment of the proposed protocol in the real world should be carried out using network technologies such as LoRa and LTE-M, both of which are well adapted to industrial applications thanks to their capability to handle long-range communications with minimal energy consumption. On the other hand, the rise of quantum computing poses significant challenges for IoT security, fundamentally transforming the protection, communication, and processing of data among interconnected devices, with the potential to undermine existing cryptographic methods and expose sensitive information to unprecedented risks. As this technology advances, IoT developers, manufacturers, and users must grasp its implications and proactively address potential vulnerabilities.

In the future, based on the proposed protocol, we will analyze the network delay and consider the quantum attacks. Further, we may integrate artificial intelligence and Quantum key distribution combined with blockchain to enhance the security of key exchanges and data transmission, ensuring confidentiality and integrity. On the other side, we will study the impact of edge computing on the IoT authentication. Edge computing enhances IoT authentication scalability and reduces latency by processing requests locally, which speeds up responses and minimizes data travel to central servers. This approach distributes the load across multiple nodes, enabling faster authentication and allowing devices to

cache tokens for offline use, while preprocessing data improves security.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

SD and MA conducted the research and collected the data; AG analyzed the data and done simulation; AEA has revised the paper; SD wrote the paper; MAAH has analyzed the obtained results; all authors had approved the final version.

REFERENCES

[1] S. Sontowski, M. Gupta, S. S. L. Chukkapalli *et al*., "Cyber-attacks on smart farming infrastructure," in *Proc. the 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC) Atlanta*, GA, USA, 2020, pp. 135–143.

[2] K. Demestichas, N. Peppes, and T. Alexakis, "Survey on security threats in agricultural IoT and smart farming," *Sensors*, vol. 20, 2020. doi: 10.3390/s20226458

[3] Z. A. Abduljabbar, V. O. Nyangaresi, H. M. Jasim *et al*., "Elliptic curve cryptography-based scheme for secure signaling and data exchanges in precision agriculture," *Sustainability*, vol. 15, no. 13, 10264, 2023.

[4] M. Dhanaraju, P. Chenniappan, K. Ramalingam *et al*., "Smart farming: Internet of Things (IoT)-based sustainable agriculture," *Agriculture*, vol. 12, no. 10, 1745, 2022.

[5] M. Azrour, J. Mabrouki, G. Fattah *et al*., "Machine learning algorithms for efficient water quality prediction," *Modeling Earth Systems and Environment*, vol. 8, no. 2, pp. 2793–2801, 2022.

[6] J. Mabrouki, K. Azoulay, S. Elfanssi *et al*., "Smart system for monitoring and controlling of agricultural production by the IoT," in *IoT and Smart Devices for Sustainable Environment*, Springer, 2022, pp. 103–115.

[7] J. Mabrouki, G. Fattah, S. Kherraf *et al*., "Artificial intelligence system for intelligent monitoring and management of water treatment plants," *Emerging Real-World Applications of Internet of Things*, pp. 69–87, 2022.

[8] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of internet of things in smart agriculture: A survey," *Futur. Gener. Comput. Syst.*, vol. 126, pp. 423–454, 2022. doi: 10.1016/j.future.2021.08.006

[9] J. Contreras-Castillo, J. A. Guerrero-Ibañez, P. C. Santana-Mancilla *et al*., "SAgric-IoT: An IoT-based platform and deep learning for greenhouse monitoring," *Applied Sciences*, vol. 13, no. 3, 1961, 2023.

[10] A. B. Kathole, K. N. Vhatkar, S. Kumbhare *et al.*, "IoT-based smart agriculture for onion plant disease management: A comprehensive approach," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 13s, pp. 472–476, 2024.

[11] A. Gupta and P. Nahar, "Classification and yield prediction in smart agriculture system using IoT," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 10235–10244, 2023.

[12] A. D. Mini, M. A. S. Anuradha, G. S. R. Asha *et al.*, "IoT based smart agriculture monitoring system," *International Research Journal of Engineering and Technology*, vol. 10, no. 4, pp. 1442–1448, 2023.

[13] M. Azrour, S. Dargaoui, J. Mabrouki *et al.*, "A survey of machine and deep learning applications in the assessment of water quality," in *Technical and Technological Solutions Towards a Sustainable Society and Circular Economy*. *World Sustainability Series*, J. Mabrouki and A. Mourade, Eds. Cham: Springer Nature Switzerland, 2024, pp. 471–483. doi: 10.1007/978-3-031-56292-1_38

[14] S. Dargaoui, M. Azrour, A. El Allaoui *et al.*, "An overview of the security challenges in IoT environment," in *Advanced Technology for Smart Environment and Energy*. *Environmental Science and Engineering*, J. Mabrouki, A. Mourade, A. Irshad, and S. A. Chaudhry, Eds., Cham: Springer International Publishing, 2023, pp. 151–160. doi: 10.1007/978-3-031-25662-2_13

[15] S. Dargaoui, M. Azrour, J. Mabrouki *et al.*, "Security issues in internet of medical things," in *Blockchain and Machine Learning for IoT Security*, Chapman and Hall/CRC, 2023, pp. 77–91.

[16] S. Dargaoui, M. Azrour, A. El Allaoui *et al.*, "Authentication in internet of things: State of art," in *Proc. the 6th International Conference on Networking, Intelligent Systems and Security*, Larache, Morocco, 2023, pp. 1–6.

[17] M. Azrour, J. Mabrouki, A. Guezzaz *et al*., "New enhanced authentication protocol for Internet of Things," *Big Data Min. Analytics*, vol. 4, 2021. doi: 10.26599/BDMA.2020.9020010

[18] S. Dargaoui, M. Azrour, A. El Allaoui *et al*., "Applications of blockchain in healthcare: Review study," in *IoT, Machine Learning and Data Analytics for Smart Healthca*re, CRC Press, 2024, pp. 1–12.

[19] S. Dargaoui, M. Azrour, A. El Allaoui *et al.*, "An exhaustive survey on authentication classes in the IoT environments," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 12, no. 1, 2024.

[20] S. Dargaoui, M. Azrour, A. El Allaoui *et al*., "Internet-of-Things-Enabled smart agriculture: security enhancement approaches," in *Proc. 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Morocco, 2024, pp. 1–5. doi: 10.1109/IRASET60544.2024.10548705

[21] S. Dargaoui, M. Azrour, A. El Allaoui *et al.*, "Internet of things authentication protocols: Comparative study," *Computers, Materials and Continua*, pp. 1–10, 2024. doi: 10.32604/cmc.2024.047625

[22] S. Dargaoui, M. Azrour, A. El Allaoui *et al*., "IoT-driven smart agriculture: security issues and authentication schemes classification," in *Proc. the International Conference on Connected Objects and Artificial Intelligence (COCIA2024)*, Y. Mejdoub and A. Elamri, Eds. Cham: Springer Nature Switzerland, 2024, pp. 61–66. doi: 10.1007/978-3-031-70411-6_10

[23] S. Itoo, A. A. Khan, M. Ahmad *et al.*, "A secure and privacy-preserving lightweight authentication and key exchange algorithm for smart agriculture monitoring system," *IEEE Access*, vol. 11, pp. 56875–56890, 2023.

[24] B. Hassan, A. A. AlSanad, I. Ullah *et al.*, "A cost effective identity-based authentication scheme for internet of things-enabled agriculture," *Wireless Communications and Mobile Computing*, vol. 2022, 4275243, 2022.

[25] A. Vangala, A. K. Das, A. Mitra *et al.*, "Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 904–919, 2022.

[26] S. Jegadeesan, S. K. Dharshini, M. Jeevitha *et al*., "Physically secure and privacy preserving blockchain based authentication scheme for IoT enabled farms," in *Proc. 2022 International Virtual Conference on Power Engineering Computing and Control: Developments in Electric Vehicles and Energy Sector for Sustainable Future (PECCON)*, IEEE, 2022, pp. 1–7.

[27] R. Kumar, S. Singh, and P. K. Singh, "A secure and efficient computation based multifactor authentication scheme for Intelligent IoT-enabled WSNs," *Computers and Electrical Engineering*, vol. 105, 108495, 2023.

[28] B. Khalid, K. N. Qureshi, K. Z. Ghafoor *et al.*, "An improved biometric based user authentication and key agreement scheme for intelligent sensor based wireless communication," *Microprocessors and Microsystems*, vol. 96, 104722, 2023.

[29] P. Guo, W. Liang, and S. Xu, "A privacy preserving four-factor authentication protocol for internet of medical things," *Computers and Security*, vol. 137, 103632, 2023.

[30] J. Pirayesh, A. Giaretta, M. Conti *et al.*, "A PLS-HECC-based device authentication and key agreement scheme for smart home networks," *Computer Networks*, vol. 216, 109077, 2022.

[31] S. U. Jan, A. Ghani, A. Alzahrani *et al.*, "Bandwidth and power efficient lightweight authentication scheme for healthcare system," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, 101601, 2023.

[32] J. Cui, F. Cheng, H. Zhong *et al.*, "Multi-factor based session secret key agreement for the industrial internet of things," *Ad Hoc Networks*, vol. 138, 102997, 2023.

[33] S. Nashwan, "Secure authentication scheme using Diffie–Hellman key agreement for smart IoT irrigation systems," *Electronics*, vol. 11, no. 2, 188, 2022.
[34] A. G. Mirsaraei, A. Barati, and H. Barati, "A secure three-factor authentication scheme for IoT environments," *Journal of Parallel and Distributed Computing*, vol. 169, pp. 87–105, 2022.
[35] S. Alyahya, W. U. Khan, S. Ahmed *et al.*, "Cyber secure framework for smart agriculture: Robust and tamper-resistant authentication scheme for IoT devices," *Electronics*, vol. 11, no. 6, 963, 2022.
[36] R. Ali, A. K. Pal, S. Kumari *et al.*, "A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring," *Future Generation Computer Systems*, vol. 84, pp. 200–215, 2018.
[37] H. Khalid, S. J. Hashim, S. M. S. Ahmad *et al.*, "Robust multi-gateway authentication scheme for agriculture wireless sensor network in society 5.0 smart communities," *Agriculture*, vol. 11, no. 10, 1020, 2021.
[38] A. Vangala, A. K. Das, and J.-H. Lee, "Provably secure signature-based anonymous user authentication protocol in an internet of things-enabled intelligent precision agricultural environment," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 16, e6187, 2023.
[39] S. Dargaoui, M. Azrour, A. El Allaoui *et al.*, "ECC-Based anonymous and multi-factor authentication scheme for IoT environment", *International Journal of Online and Biomedical Engineering (IJOE)*, vol. 21, no. 1, pp. 56–75, 2025.
[40] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2004, pp. 523–540.
[41] O. Günlü and R. F. Schaefer, "An optimality summary: Secret key agreement with physical unclonable functions," *Entropy*, vol. 23, no. 1, 16, 2020.
[42] A. A. M. Gharbi and A. S. Nori, "Honey encryption security techniques: A review paper," *AL-Rafidain Journal of Computer Sciences and Mathematics*, vol. 16, no. 1, pp. 1–14, 2022.
[43] S. Sawant, P. Saptal, K. Lokhande *et al.*, "Honeywords: Making password cracking detectable," *Journal Impact Factor*, vol. 2, 2018.
[44] K. Chen, P. Shen, K. Lv *et al.*, "Dynamic group fuzzy extractor," *Cybersecurity*, vol. 7, no. 1, 25, 2024. doi: 10.1186/s42400-024-00210-2
[45] N. Singh and A. K. Das, "TFAS: Two factor authentication scheme for blockchain enabled IoMT using PUF and fuzzy extractor," *J. Supercomput*., vol. 80, no. 1, pp. 865–914, 2024. doi: 10.1007/s11227-023-05507-6
[46] A. Kuznetsov, A. Kiyan, A. Uvarova *et al.*, "New code based fuzzy extractor for biometric cryptography," in *Proc. 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, 2018, pp. 119–124. doi: 10.1109/INFOCOMMST.2018.8632040
[47] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[48] B. Blanchet, B. Smyth, V. Cheval *et al.*, "Automatic cryptographic protocol verifier, user manual and tutorial," CNRS, Departement dInformatique, Ecole Normale Superieure, Paris, 2015.