Honey-List Based Authentication Protocol for Industrial IoT Swarms

Mohamed A. El-Zawawy[a], Pallavi Kaliyar[b], Mauro Conti[c], Sokratis Katsikas[b]

[a]*Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt*
[b]*Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik, Norway*
[c]*Department of Mathematics and HIT Research Center, University of Padova, 35121, Padua, Italy*

Abstract

Industrial Internet of Things (IIoT) systems are advanced IoT systems composed of sensor devices supported with dynamic objects such as smart vehicles and drones. The collaboration among static and heterogeneous mobile objects makes the topologies of IIoT systems dynamic and complex. This dynamic topology is also partially due to that fact that the static devices are typically partitioned into categories of collaborating sensors (called swarms) managed by side servers. However, existing authentication techniques for IIoT systems do not consider realistic system models simultaneously hosting different types of dynamics objects. For such scenarios, there is a need for protocols that guarantees a secure Entity-to-Entity (E2E) communication, thus ensuring a smooth and safe production process.

In this paper, we present HASFAV, a lightweight and locality-aware key agreement and authentication protocol for IIoT systems, to enable efficient and secure E2E communication between devices in the same or different partitions. HASFAV fills the gap of considering a realistic system model simultaneously hosting different types of dynamics objects. We employ Honey lists (lists with algorithms used to prevent guessing passwords) and mutual authentication technologies in HASFAV to guarantee its security against different attacks, even in public-channel communication scenarios. Using the well-established Real-Or-Random (ROR) model, we proved the security of HASFAV in detail. We also provide a prototype implementation of HASFAV in the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. This tool confirms the results of our theoretical proofs, thus verifying the security of HASFAV. We also carried out a detailed comparative study of HASFAV against existing related authentication techniques. Compared to these techniques, HASFAV offers more functionality (serving more types of dynamic objects) and superior security (via proving backup plans for session keys establishment). Finally, we prove that HASFAV is practical by implementing it in a well-known network simulator, called Omnet++.

Keywords: Industrial IoT; Authentication protocols; Smart Drones; IoT Swarms; Smart Farming

1. Introduction

In 2021, the number of active Internet of Things (IoT) devices was greater than 10 billion. Although this number is already huge, it is expected to exceed 25.4 billion by the year 2030 [1]. The exponential increase in the use and manufacturing of the number of smart devices makes IoT services a huge part of our daily lives. These small, resource-constrained IoT devices are usually interconnected in multiple groups (called swarms, which collectively achieve specific tasks) while employed in different applications. In particular, they are referred to as *swarms of resource-constrained devices* [50].

Industrial Internet of Things (IIoT) is a subset of IoT, mainly used in industrial applications like smart factories, smart farming, and smart cities [16]. It is common for modern network topologies of IIoT to be organized in swarms. We refer to these topologies as swarm IIoT (SIIoT). In the SIIoT context, where real-time data is used to make decisions, the security, authenticity, confidentiality, and integrity of the data are essential requirements. It is estimated that the total expenses for global IoT will reach 15 trillion USA dollars by 2025 [1]. Since the user and the sensor nodes used in industrial communication communicate over insecure channels, the information can be intercepted and altered by an adversary during the transmission, and any modifications on data may negatively affect decisions that are based on the received data. Therefore, secure mutual authentication and session key negotiation schemes play a crucial role in securely transmitting data between users and industrial sensors [42]. Drones and vehicles provide realistic solutions to restrictions of terrestrial frameworks of ubiquitous networks, including SIIoT. The potential of drones and vehicles make them essential entities of modern SIIoT systems. However, these entities further increase the security challenges for SIIoT networks [38, 26].

Smart farming [17] is among the most common appli-

Email addresses: maelzawawy@cu.edu.eg (Mohamed A. El-Zawawy), pallavi.kaliyar@ntnu.no (Pallavi Kaliyar), conti@math.unipd.it (Mauro Conti), sokratis.katsikas@ntnu.no (Sokratis Katsikas)

cations of SIIoT that can greatly benefit from the integration of smart vehicles and drones [32]. A smart farming technology that is based on secure mutual authentication can make agricultural data more secure for the farmers. The mutual authentication scheme prevents the leakage and tampering of network data. This scheme adds user anonymity, transparency, and integrity to the data and gives the user confidence to use the correct data. The IoT technology offers many services that can improve the farming process (commonly termed as Production Internet of Things) and agricultural production [12]. Therefore, SIIoT and PIoT reflect the same concept. For instance, a swarm of smart sensors collects data on weather conditions, soil quality, cattle's health, crop growth progress, and more. The collected data can be analyzed to draw useful conclusions and changes. This way, the gathered data helps the farmer to monitor their current production and help them plan necessary modifications for the next season (e.g., choose one crop instead of another based on soil conditions). However, the IoT devices' low-cost and resource-constrained nature leads to an increased attack surface; thus, when used in sensitive applications, they become easy targets for various cyber attacks [33, 34].

1.1. Contribution

This paper presents a novel key management and authentication protocol, called `HASFAV`, for SIIoT which are supported with smart vehicles and drones. `HASFAV` is presented and demonstrated on the smart farming application of SIIoT. We carried a wide comparative study and detailed security analysis that proved that `HASFAV` overcomes related protocols in many aspects: providing more functionality characteristics, better security, requiring less communication, and computation costs. In addition, to prove the practical perspective of `HASFAV`, we carried out a precise implementation of `HASFAV` using a well-established networking simulation tool, Omnet++, and its simulation environment, Castalia. In summary, the key contributions of this work are as follows:

- We propose a lightweight locality-aware key agreement and authentication protocol, `HASFAV`, for SIIoT systems to enable the E2E communication between devices in the same and different partitions. To overcome vulnerability against many attacks and boost the security of our proposed protocol, we employ a Honey list and mutual authentication technologies in `HASFAV`. The security of `HASFAV` is proved using the well-established Real-Or-Random (ROR) model and Automated Validation of Internet Security Protocols and Applications (AVISPA) tool.
- To show the efficiency of `HASFAV`, we compared it against existing related authentication techniques. The results showed that `HASFAV` has more functionality characteristics and superior security. More precisely, `HASFAV` reduces the energy consumption, on average, up to 65% and the computational cost up to 77.5% compared to other related authentication protocols.

1.2. Organization

The rest of the article is organized as follows: Section 2 briefly introduces the related work. Section 3 describes the system and threat model. We propose and explain the working methodology of the proposed protocol in Section 4. Section 5 presents the security analysis of `HASFAV`. The evaluation of the protocol is discussed in Section 6. Finally, we conclude the paper with an outline of future work directions in Section 7.

2. Related Work

The authors of [43] presented a novel remote user authentication scheme based on the Internet of things (IoT). They claimed that their proposed scheme is safe against various IoT threats as well as providing mutual authentication among all parties involved, password protection, password change facility, and dynamic node addition to the system. However, the authors in [15] suggested that the solution presented in [43] is prone to smart card-stolen attack, sensor node spoofing attack, stolen verifier attack, and impersonation attacks, and they proposed two remote user authentication schemes for IoT to remedy these security pitfalls. In 2015, the authors of [22] implemented all the attacks presented in [43] and proposed an improvement to the authentication scheme of [43], which has been claimed to overcome all the security vulnerabilities of the latter. The authors of [8] improved the protocol presented in [22]. In their work, they initially proved that the solution presented in [22] is susceptible to known session-specific temporary information, offline password guessing, and the stolen smart card attacks and claimed that their solution provides a remedy to all the security vulnerabilities of the scheme of [22] and it is more suitable and efficient for IoT based networks.

The authors of [7] also crypt-analyzed the scheme presented in [43] and found some security vulnerabilities, such as inefficient authentication phase, which make sensor node impersonation, smart card theft, offline identity, and password guessing attacks feasible. In their work, they proposed a solution to overcome all these. Later, the authors of [28] showed that the scheme proposed in [7] is susceptible to the leak-able identity of users, offline password guessing, user impersonation, and session key temporary information attacks. In their work they proposed a solution to overcome these security weaknesses, and claimed that their scheme facilitates an elevated level of security for IoT networks. In 2017, the authors of [4] proposed a novel remote user authentication scheme for agriculture monitoring in IoT environment. Their scheme proposed an effective, lightweight solution for lightweight operations. In the sequel we compare the

proposed HASFAV scheme with the scheme presented in [4].

The authors of [21] proposed a key agreement and signature-based authentication scheme using ECC-based digital signatures for IoT environment. They claimed that their solution provides user untraceability and anonymity. Later, in [47], the author showed that the scheme proposed in [14] has a higher computation cost than other related schemes. The authors of [47] proposed a novel solution for secure remote user authentication in a generic IoT environment. However, the authors of [21] claimed that the scheme proposed in [47] is inefficient for resource-constrained networks, since it uses the XOR operation and a one-way hash function. They showed that the scheme stores its verification tables based on the gateway data; as such, an attacker getting hold of this table can be fatal for the complete system.

The authors of [39] proposed a remote user authentication scheme for the IoT environment using cloud infrastructure. However, in their scheme, they were not able to achieve timely typo detection and clock synchronization [25]. The authors of [41] proposed a remote authentication scheme for IoT environments using ECC for smart home applications. However, the author in [21] showed that the scheme suffers from insufficient computational and communication cost performance and is susceptible to privileged insider and parallel session attacks. The authors of [46] presented the new challenges they faced while designing sound multi-factor schemes, emphasizing the flaws they found in previous multi-factor schemes for multi-server environments. The authors of [37] proposed a novel three-factor remote user authentication scheme based on ECC for IoT environments, to preserve the smart device's privacy and data confidentiality and the communicating user. They claimed that their scheme is not vulnerable to attacks. In order to support their claim, they implemented and analyzed multiple cryptographic attacks, and they used the AVISPA formal security analysis method for simulating their scheme.

Recently, the authors of [40] proposed a new secure authentication scheme for IIoT systems, mainly focused on forwarding secrecy. They used the Rabin cryptosystem in their scheme and avoided using the usual password verification table. They claimed that their scheme provides the desired security and functionality for IIoT system-based realistic scenarios. They used formal proof and heuristic analysis to support their claim. The authors of [44] proposed a blockchain-based smart farming authentication scheme for smart agricultural applications in the IoT environment. They used a blockchain-enabled, smart-contract-based authentication key agreement mechanism in their scheme. The security analysis of the proposed scheme was done both formally and informally, using security verification tools. The authors claim that their scheme provides more functionality and superior security, compared to existing competing authentication protocols.

3. System and Threat Models

In this section we provide an overview of the considered scenario and the system model in Section 3.1. Then, we present the threat model in Section 3.2, discussing both the reasons behind an attack and the considered attacker model. We present important characteristics of smart farming, drones, and smart vehicles in Section 3.3, and details of our use of honey lists in Section 3.4.

3.1. System Model

Our network model for Intent of Things (IoT)-smart farming is supported with drones and autonomous agricultural vehicles as shown in Figure 1. Our model and its main entities are described as follows: The network of the farming field is divided into zones. Each zone is equipped with several IoT-smart devices. Each zone is also supported by a field server. Each IoT device communicates with the field server of its zone, which also monitors the device. The whole network is controlled and monitored by the trust Authority (TA) entity which is the only trusted party in the system. It is responsible for the registration of all other system parties: field servers, drones, IoT-devices, and smart agricultural vehicles. The registration must proceed with deploying these entities in the IoT-supported agriculture network. The model is supported with drones that communicate with the TA and the field servers. The system is also flexible enough to allow drones to communicate with IoT-devices. The model is supported by autonomous agricultural vehicles communicating with the TA and the field servers, but not with the IoT-devices.

Two types of mutual authentication occur during the authentication phase and before data transmission. The first type involves a drone, a field server, and an IoT-device. The other type involves an autonomous agricultural vehicle, the TA, and a field server. According to our system details, mutual authentication is done in a key management process that aims to achieve secure communication among different system entities. The honey list concepts are applied to prevent several attacks, as described later in the paper on different occasions. All in all, we propose a security protocol that guarantees validation and authentication of the system parties.

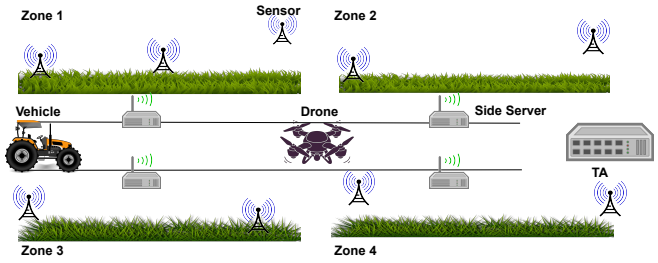

Figure 1: Proposed Network Model for Smart farming Assisted with Drone and Autonomous Vehicle.

3.2. Threat Model

Our proposed protocol considers two widely-known and accepted threat models: Dolev-Yao (DY) [20] and Canetti-Krawczyk (CK) [13]. Therefore, an adversary can inject, modify, and delete communicated messages in the wireless network of the farming area. Using session hijacking attacks and according to our opted threat models, an adversary can also compromise the content of insecure memories of different system entities. Such content may include session states and secret keys, and credentials. Except for trust authority, the adopted models do not treat other system entities (drones, smart vehicles, and smart IoT sensors) as secure ones. It is justified by the fact that in an agriculture field, the adversary has a good chance of physically compromising an autonomous vehicle, a drone, or a sensor. Power analysis attacks [29] can then be performed by the adversary on the physically compromised entities to obtain stored credentials. This gives the adversary a chance to launch other attacks, such as impersonation attacks.

3.3. Smart Farming, Agricultural Autonomous Vehicle, and Drone Characteristics

It is predicted that by the year 2050, the world population is expected to exceed 10 billion. This emphasizes the need for continuous growth in agricultural products. Smart farming [23, 11] relies on a huge number of different types of IoT-devices and sensors. These include Optical encoder (converting motion into pulses), Light Detection And Ranging (LIDAR), mechanical sensors (such as accelerometers, force, and pressure sensors), Geomagnetic Direction Sensor (GDS- capturing the magnetic field of earth), ultrasonic, Fiber Optic Gyroscope (FOG), Acoustic sensor (for measuring levels of sounds), and Laser Radar (LADAR).

Autonomous Agricultural Vehicles (AAVs) are the primary tools for smart farming and precision agriculture [35]. Hence, the characteristics and security of AAVs are the main issues in smart farming. Automated farm vehicles have three categories; drive assistance, autonomous, and automatic steering. AAVs have different types of navigation systems, including neural network and genetic algorithm, dead reckoning, fuzzy logic control, image processing, Kalman filter-based, and statistically based developed algorithms. These navigation systems are supported and enabled by long-life batteries and rich computational resources of AAVs [36]. Two main units of AAVs are the Time Kinematic-Global Positioning System (RTKGPS) unit which enhances GPS data signals to provide centimeter-level accuracy, and the Proportional–Integral–Differential (PID) unit, which is a closed-loop monitoring system necessary for determining optimal working conditions [31].

Drones play a prominent role in smart farming due to their characteristics [6]. A drone is a special type of Unmanned Aircraft System (UAS). The main components of a drone are [3]:

1. Chassis: the drone skeleton.
2. Propellers: the drone load and speed depends on their length. Compared to other domains, drone propellers in agricultural applications do not need to be long.
3. Flight Controller: the drone computer.
4. Electronic Speed Controller (ESC): controls the current supplied to drone motors.
5. Motors: a drone has one motor per propeller. Motors are characterized by revolution count per minute, measured in "kV" units.
6. Radio Receiver: receives signals.
7. Battery: lithium batteries.
8. Sensors: of different types such as GPS, cameras, and barometers.

3.4. Honey Lists

A Honey Encryption (HE) algorithm [5] stops an adversary from guessing the password and hence obtaining secret data. Such algorithm is triggered upon the adversary attempt of decrypting user credentials via a honeyword or wrong password. In response to an off-line guessing and brute-force attack, the HE algorithm generates and sends a fake message. Our protocol relies on honey lists, which can be thought of as some sort of an HE algorithm. During the authentication phase, many algorithms using a honey list can be used to prevent guessing attacks [37]. In this paper we adopt one of these algorithms that have recently proved to be successful in related protocols [10]. Upon login, the algorithm enables the TA to detect an intrusion by watching the adversary's login source. Our HE algorithm uses a predefined threshold for the allowed number of credentials guessing. Exceeding this threshold leads the TA to terminate the communication session, thus preventing the intrusion. Therefore, the algorithm continuously compares the count of elements in the honey list to the threshold.

4. HASFAV: Novel Authentication Protocol for Smart Farming

This section introduces HASFAV, a new and robust protocol for authentication in smart farming supported with drones and smart vehicles. HASFAV overcomes limitations and security issues of existing authentication protocols. We distinguish the following phases in the operation of HASFAV

1. **Registration phase:** In this phase, using a secure channel, the TA registers all system entities, including drones, vehicles, side servers, and sensors.
2. **Login and authentication phase:** It enables system entities to authenticate with the TA. Afterwards, session keys among system entities are created.

3. **Entity-2-Entity (E2E) communication phase:** It facilitates secure communication among different system entities. This covers entities of different field zones.
4. **Addition of new sensor phase:** It enables the addition of new sensors to already running systems.

Our protocol applies the honey list concept in the TA. Therefore, statistics of failed logins are recorded in the TA. Hence, attacks relying on off-line password guessing can be detected using a certain threshold for erroneous attempts. This results in the honey list stopping the authentication process.

The notations used in this paper are presented in Table 1. `HASFAV` uses clocks of system entities to protect against replay attacks. This is common for authentication protocols of different networking applications [19, 18, 27]. Therefore, we assume these clocks are synchronized.

Table 1: Notations used in this paper.

Notation	Meaning
K_{DT}	The shared secret key between drone and TA.
$\mathcal{I}^e$	The identity of system entity e.
$p\mathcal{I}^e$	The auxiliary identity of system entity e.
n_i^e	Random string generated during registration of system entity e.
u_i^k, r_i^k	Random string generated during log-in and authentication phase of the proposed protocol.
M_i^e	Message established during registration of system entity e.
T_i	Current timestamp.
N_i^k, L_i^k	Message established during log in and authentication phase of the proposed protocol.
$\|$	The concatenation operation.
$\oplus$	The Xor operations.
`GenRan()`	The action of generating a random string.
`Fetch`$((n_2^d, n_3^d) \mapsto p\mathcal{I}_1^d)$	Fetch from memory the pair (n_2^d, n_3^d) corresponding to $p\mathcal{I}_1^d$).
`UpdSynHoney`(M_2^{d*})	Update the honey list of the field server and synchronize with that of the server.
`SndOpn()`	The action of sending a message on an open channel.
`SndSec()`	The action of sending a message on a secure channel.
`Time_Stamp()`	The action of obtaining the current timestamp.

4.1. Registration Phase

For system communication, the sensors, drones, vehicles, and servers need to register with the TA. This has to be done through a secure channel. The details of the registration process follow.

4.1.1. Drone and Vehicle Registration

Algorithm 1 shows the steps for a drone or a vehicle, denoted by x, to register with the TA. Three procedures are involved, one that creates and sends a registration request by x to the TA (lines 6-12 of algorithm 1); one that sends the TA's response to x and also stores the generated credentials into the TA's and the field servers' secure memories (lines 13-21 of algorithm 1); and one that stores x's credentials to its smart card.

Algorithm 1 `Registration Details of Drones and Vehicles.`

Input: All the system model entities.
Steps:

1: **for each** $x \in$ `Drones` $\cup$ `Vehicles` **do**
2: Call `Drone-Vehicle`$_1(x)$;
3: Call `Trust_Authority`$_1(M_1^x, K_{DT})$;
4: Call `Drone-Vehicle`$_2(M_1^x, K_{DT})$;
5: **procedure** `Drone-Vehicle`$_1(x)$
6: Generate an ID, $\mathcal{I}_1^x$ and a random string s_I;
7: $\mathcal{I}_2^x \leftarrow h(\mathcal{I}_1^x \oplus s_I)$;
8: Generate a random string n_1^x;
9: $p\mathcal{I}_1^x \leftarrow h(\mathcal{I}_1^x \| n_1^x)$;
10: $p\mathcal{I}_2^x \leftarrow h(\mathcal{I}_2^x \| (n_1^x \oplus p\mathcal{I}_1^x)$;
11: $M_1^x \leftarrow (p\mathcal{I}_1^x, p\mathcal{I}_2^x \oplus n_1^x)$;
12: Send the message M_1^x, on a secure channel, to `Trust_Authority_1`();
13: **procedure** `Trust_Authority`$_1()(M_1^x, K_{DT})$
14: $p\mathcal{I}_1^x, p\mathcal{I}_2^x \oplus n_1^x \leftarrow M_1^x$;
15: Generate random strings n_2^x and n_3^x;
16: $M_2^x \leftarrow h(n_2^x \oplus (K_{DT} \| p\mathcal{I}_1^x))$;
17: $M_3^x \leftarrow M_2^x \oplus p\mathcal{I}_2^x \oplus n_1^x$;
18: $M_4^x \leftarrow h(M_2^x \| M_3^x)$;
19: $M_5^x \leftarrow (M_3^x, M_4^x, n_3^x)$;
20: Store $(n_2^x, n_3^x, p\mathcal{I}_1^x,$ `HONEY-LIST = []`$, K_{DT})$ in secure memories of TA and field servers;
21: Send the message M_5^x, on a secure channel, to `Drone-Vehicle`$_2()$;
22: **procedure** `Drone-Vehicle`$_2(M_5^x)$
23: $(M_3^x, M_4^x, n_3^x) \leftarrow M_5^x$;
24: $M_6^x \leftarrow h(\mathcal{I}_1^x \| \mathcal{I}_2^x) \oplus n_1^x$;
25: $M_7^x \leftarrow M_3^x \oplus n_1^x = M_2^x \oplus p\mathcal{I}_2^x$;
26: $M_8^x \leftarrow h(M_4^x \| p\mathcal{I}_2^x)$;
27: Store $(M_6^x, M_7^x, M_8^x, n_3^x)$ into smart card of the drone/vehicle;

4.1.2. Sensors and Field-Servers Registration

Algorithm 2 , shows the steps for a sensor or field server, denoted by y, to register with the TA. Again, three

procedures are involved, one that creates and sends a registration request by y to the TA (lines 5-11 of algorithm 2; one that sends the TA's response to y and also stores the generated credentials into the TA's and the field servers' secure memories (lines 12-21 of algorithm 2); and one that stores y's credentials to its secure memory.

Algorithm 2 `Registration Details of Sensors and Field-Servers.`

Input: All the system model entities.
Steps:

1: **for each** $y \in$ (`Sensors` $\cup$ `Field-Servers`) **do**
2: Call `Sensor-Server`$_1(y)$;
3: Call `Trust_Authority`$_2(M_4^y)$;
4: Call `Sensor-Server`$_2(M_7^y)$;
5: **procedure** `Sensor-Server`$_1(y)$
6: Generate an ID, $\mathcal{I}^y$ and a random string n_1^y;
7: $M_1^y \leftarrow h(\mathcal{I}^y \oplus n_1^y)$;
8: $M_2^y \leftarrow M_1^y \oplus \mathcal{I}^y$;
9: $M_3^y \leftarrow M_1^y \oplus K_{DT}$;
10: $M_4^y \leftarrow (M_2^y, M_3^y)$;
11: Send the message M_4^y, on a secure channel, to `Trust_Authority_1`();
12: **procedure** `Trust_Authority`$_2(M_4^y)$
13: $(M_2^y, M_3^y) \leftarrow M_4^y$;
14: $M_1^y \leftarrow M_3^y \oplus K_{DT}$;
15: $\mathcal{I}^y \leftarrow M_1^y \oplus M_2^y$;
16: Generate a random string n_2^y;
17: $M_5^y \leftarrow h(\mathcal{I}^y \| K_{DT} \| n_2^y)$;
18: $M_6^y \leftarrow h(M_1^y \| \mathcal{I}^y \| K_{DT} \| n_2^y)$;
19: $M_7^y \leftarrow (M_5^y, M_6^y)$;
20: Store $(\mathcal{I}^y, n_2^y, M_1^y, K_{DT})$ in the secure memories of TA and the field server of y;
21: Send the message M_7^y, on a secure channel, to `Sensor-Server`$_2$();
22: **procedure** `Sensor-Server`$_2(M_7^y)$
23: Store (M_5^y, M_6^y, K_{DT}) into secure memory of y;

4.2. Login and Authentication Phase

If a system entity requests communication with any other entity, the former must login and authenticate with the TA or a field server. After successful login and authentication, the system entities can communicate securely. There are two types of authentication processes in our protocol. The first involves a drone, a field server, and a sensor. The other type involves a vehicle, the TA, and a field server. The details of these authentication processes are given below.

4.2.1. Type 1 Authentication (Sensor - Server - Drone)

Algorithm 3 presents how this authentication process works. The process involves five procedures, namely one

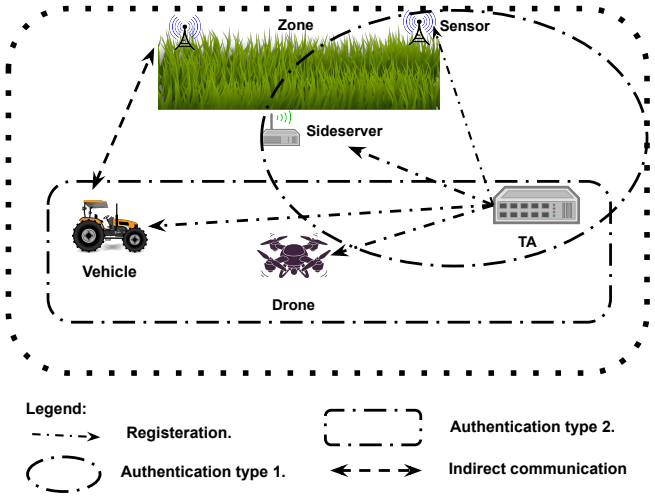

Figure 2: Visualization of the E2E Communication Process.

that creates an authentication request of the drone d and sends it to the field server (lines 2-14 of algorithm 3); one that checks the validity of the request, and then generates and sends the response to d or rejects the request and updates the honey list (lines 15-34 of algorithm 3); one that authenticates the field server to d, creates the necessary session key, and sends it to the field server (lines 35-51 of algorithm 3); one that checks the validity of the timestamp included in the previous message and re-calculates the session key at the server side before sending it to d (lines 52-69 of algorithm 3); and one that eventually starts an authenticated session, or aborts, depending on the result of a number of validity checks (lines 70-82 of algorithm 3).

4.2.2. Type 2 Authentication (Drone- TA - Vehicle)

The details of this authentication process are shown in Algorithm 4. Like in the case of algorithm 4, the process involves five procedures, namely one that creates an authentication request of the vehicle v and sends it to the TA (lines 2-13 of algorithm 4); one that allows the TA to check the validity of the request, and then generates and sends the response to the field server or rejects the request and updates the honey list (lines 14-33 of algorithm 4); one that authenticates the field server to v, creates the necessary session key, and sends it to the TA (lines 34-50 of algorithm 4); one that checks the validity of the timestamp included in the previous message and re-calculates the session key at the TA side before sending it to v (lines 51-68 of algorithm 4); and one that eventually starts an authenticated session, or aborts, depending on the result of a number of validity checks (lines 69-81 of algorithm 3).

4.3. The Entity-2-Entity Communication Phase

Sensor-2-TA Communication and Sensor-2-Vehicle Communication are common indirect communications in our assumed system. These indirect communications employe drones and established session keys and are guaranteed to be secure, as they involve session keys shared

Algorithm 3 Session-Key Sharing between Drones, Field-Servers, and Sensors.

Input: a drone d, a field-server f, and a sensor s.
Steps:

Call $\texttt{Drone}_1(d)$;
procedure $\textsc{Drone}_1(d)$
  $T_1 \leftarrow \texttt{Time\_Stamp}()$;
  Inset smart card;
  $n_1^d \leftarrow M_6^d \oplus h(\mathcal{I}_1^d \| \mathcal{I}_2^d)$;
  $p\mathcal{I}_1^d \leftarrow h(\mathcal{I}_1^d \| n_1^d)$;
  $p\mathcal{I}_2^d \leftarrow h(\mathcal{I}_2^d \| (n_1^d \oplus p\mathcal{I}_1^d))$;
  $r_1^k \leftarrow \texttt{GenRan}()$;
  $M_2^d \leftarrow M_7^d \oplus p\mathcal{I}_2^d$;
  $N_1^k \leftarrow h(r_1^k \oplus M_2^d)$;
  $N_2^k \leftarrow h(T_1 \| n_3^d \| N_1^k)$;
  $N_3^k \leftarrow h(M_2^d \| T_1) \oplus N_1^k$;
  $N_4^k \leftarrow (N_2^k, N_3^k, p\mathcal{I}_1^d, T_1)$;
  $\texttt{SndOpn}(N_4^k, \texttt{Server}_1)$;
procedure $\textsc{Server}_1(N_4^k)$
  $(N_2^k, N_3^k, p\mathcal{I}_1^d, T_1) \leftarrow N_4^k$;
  $T_2 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_2 - T_1| \leq \Delta T$ **then**
    $\texttt{Fetch}((n_2^d, n_3^d) \mapsto p\mathcal{I}_1^d)$
    $M_2^{k*} \leftarrow h(n_2^d \oplus (K_{DT} \| p\mathcal{I}_1^d))$;
    $N_1^{k*} \leftarrow h(M_2^{k*} \| T_1) \oplus N_3^k$;
    $N_2^{k*} \leftarrow h(T_1 \| n_3^d \| N_1^{k*})$;
    if $N_2^{k*} == N_2^k$ **then**
      $r_2^k \leftarrow \texttt{GenRan}()$;
      $N_5^k \leftarrow h(\mathcal{I}^s \| n_2^s \| M_1^s \| K_{DT})$;
      $N_6^k \leftarrow h(N_1^{k*} \| h(r_2^k \| N_5^k \| n_2^d)) \oplus h((h(\mathcal{I}^s \oplus n_2^s) \oplus K_{DT}) \| N_5^k \| T_2)$;
      $N_7^k \leftarrow h((h(\mathcal{I}^s \oplus n_2^s) \oplus K_{DT}) \| N_5^k \| T_2 \| h(N_1^{k*} \| h(r_2^k \| N_5^k \| n_2^d)))$;
      $N_8^k \leftarrow (N_6^k, N_7^k, T_2)$;
      $\texttt{SndOpn}(N_8^k, \texttt{Sensor}_1)$;
    else
      $\texttt{UpdSynHoney}(M_2^{d*})$;
  else
    $\texttt{Reject}$;
procedure $\textsc{Sensor}_1(N_8^k)$
  $(N_6^k, N_7^k, T_2) \leftarrow N_8^k$;
  $T_3 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_3 - T_2| \leq \Delta T$ **then**
    $N_9^k \leftarrow N_6^k \oplus h(M_3^s \| M_6^s \| T_2)$;
    $N_7^{k*} \leftarrow h((M_3^s \| M_6^s \| T_2 \| N_9^k)$;
    if $N_7^{k*} == N_7^k$ **then**
      $r_3^k \leftarrow \texttt{GenRan}()$;
      $\texttt{SKey} \leftarrow h(N_9^k \| h(r_3^k \| M_3^s) \| T_3)$;
      $N_{10}^k \leftarrow \texttt{SKey} \oplus h(M_3^s \| M_6^s \| N_6^k \| T_3)$;
      $N_{11}^k \leftarrow h(\texttt{SKey} \| T_2 \| T_3)$;
      $N_{12}^k \leftarrow (N_{10}^k, N_{11}^k, T_3)$;
      $\texttt{SndOpn}(N_{12}^k, \texttt{Server}_2)$;
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;
procedure $\textsc{Server}_2(N_{12}^k)$
  $(N_{10}^k, N_{11}^k, T_3) \leftarrow N_{12}^k$;
  $T_4 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_4 - T_3| \leq \Delta T$ **then**
    $\texttt{SKey} \leftarrow N_{10}^k \oplus h(h((h(\mathcal{I}^s \oplus n_2^s) \oplus K_{DT}) \| M_6^s \| N_6^k \| T_3)$;
    $N_{11}^{k*} \leftarrow h(\texttt{SKey} \| T_2 \| T_3)$;
    if $N_{11}^{k*} == N_{11}^k$ **then**
      $r_4^k \leftarrow \texttt{GenRan}()$;
      $N_{13}^k \leftarrow h(r_2^k \| N_{11}^{k*} \| T_4 \| T_2 \| T_1 \| n_3^d)$;
      $N_{14}^k \leftarrow \texttt{SKey} \oplus h(N_{11}^{k*} \| p\mathcal{I}_1^d \| T_4)$;
      $N_{15}^k \leftarrow N_{13}^k \oplus h(p\mathcal{I}_1^d \| n_3^d \| N_{11}^{k*} \| T_4)$;
      $N_{16}^k \leftarrow h(\texttt{SKey} \| N_{13}^k \| T_4)$;
      $N_{17}^k \leftarrow (N_{14}^k, N_{15}^k, N_{16}^k, T_4)$;
      $\texttt{SndOpn}(N_{17}^k, \texttt{Drone}_2)$;
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;
procedure $\textsc{Drone}_2(N_{17}^k)$
  $(N_{14}^k, N_{15}^k, N_{16}^k, T_4) \leftarrow N_{17}^k$;
  $T_5 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_5 - T_4| \leq \Delta T$ **then**
    $\texttt{SKey} \leftarrow N_{14}^k \oplus h((h(M_2^{k*} \| T_1) \oplus N_3^k) \| p\mathcal{I}_1^d \| T_4)$;
    $N_{13}^k \leftarrow N_{15}^k \oplus h(p\mathcal{I}_1^d \| n_3^d \| (h(M_2^{k*} \| T_1) \oplus N_3^k) \| T_4)$;
    $N_{16}^{k*} \leftarrow h(\texttt{SKey} \| N_{13}^k \| T_4)$;
    if $N_{16}^{k*} == N_{16}^k$ **then**
      Session is authenticated.
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;

among more than one entities. Hence, if a communicating entity is an adversary, the security of the communication is still not compromised. The aforementioned indirect communications can be achieved securely as follows: Let the system drone and vehicles be denoted by D and V, respectively. Assume that a sensor S in a zone Z whose field server is F wants to communicate with TA. Let the session key $SKey_1$ be the key established among S, D, and F and

Algorithm 4 `Session-Key Sharing between Vehicles, Trust authority, and Field-Servers.`

Input: a vehicle v, the truest authority t, and a field-server f.
Steps:

Call $\texttt{Vehicle}_1(v)$;
procedure $\textsc{Vehicle}_1(v)$
  Inset smart card and assign current time stamp to T_1;
  $n_1^v \leftarrow M_6^v \oplus h(\mathcal{I}_1^v \| \mathcal{I}_2^v)$;
  $p\mathcal{I}_1^v \leftarrow h(\mathcal{I}_1^v \| n_1^v)$;
  $p\mathcal{I}_2^v \leftarrow h(\mathcal{I}_2^v \| n_1^v \oplus p\mathcal{I}_1^v)$;
  $u_1^k \leftarrow \texttt{GenRan}()$;
  $M_2^v \leftarrow M_7^v \oplus p\mathcal{I}_2^v$;
  $L_1^k \leftarrow h(u_1^k \oplus M_2^v)$;
  $L_2^k \leftarrow h(T_1 \| n_3^v \| L_1^k)$;
  $L_3^k \leftarrow h(M_2^v \| T_1) \oplus L_1^k$;
  $L_4^k \leftarrow (L_2^k, L_3^k, p\mathcal{I}_1^v, T_1)$;
  $\texttt{SndOpn}(L_4^k, \texttt{TA}_1)$;
procedure $\texttt{TA}_1(L_4^k)$
  $(L_2^k, L_3^k, p\mathcal{I}_1^v, T_1) \leftarrow L_4^k$;
  $T_2 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_2 - T_1| \leq \Delta T$ **then**
    $\texttt{Fetch}((n_2^v, n_3^v) \mapsto p\mathcal{I}_1^v)$
    $M_2^{k*} \leftarrow h(n_2^v \oplus (K_{DT} \| p\mathcal{I}_1^v))$;
    $L_1^{k*} \leftarrow h(M_2^{k*} \| T_1) \oplus L_3^k$;
    $L_2^{k*} \leftarrow h(T_1 \| n_3^v \| L_1^{k*})$;
    if $L_2^{k*} == L_2^k$ **then**
      $u_2^k \leftarrow \texttt{GenRan}()$;
      $L_5^k \leftarrow h(\mathcal{I}^f \| n_2^f \| M_1^f \| K_{DT})$;
      $L_6^k \leftarrow h(L_1^{k*} \| h(r_2^k \| L_5^k \| n_2^v)) \oplus h((h(\mathcal{I}^f \oplus n_2^f) \oplus K_{DT}) \| L_5^k \| T_2)$;
      $L_7^k \leftarrow h((h(\mathcal{I}^f \oplus n_2^f) \oplus K_{DT}) \| L_5^k \| T_2 \| h(L_1^{k*} \| h(r_2^k \| L_5^k \| n_2^v)))$;
      $L_8^k \leftarrow (L_6^k, L_7^k, T_2)$;
      $\texttt{SndOpn}(L_8^k, \texttt{Server}_3)$;
    else
      $\texttt{UpdSynHoney}(M_2^{v*})$;
  else
    $\texttt{Reject}$;
procedure $\textsc{Server}_3(L_8^k)$
  $(L_6^k, L_7^k, T_2) \leftarrow L_8^k$;
  $T_3 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_3 - T_2| \leq \Delta T$ **then**
    $L_9^k \leftarrow L_6^k \oplus h(M_3^f \| M_6^f \| T_2)$;
    $L_7^{k*} \leftarrow h((M_3^f \| M_6^f \| T_2 \| L_9^k)$;
    if $L_7^{k*} == L_7^k$ **then**
      $u_3^k \leftarrow \texttt{GenRan}()$;
      $\texttt{SKey} \leftarrow h(L_9^k \| h(u_3^k \| M_3^f) \| T_3)$;
      $L_{10}^k \leftarrow \texttt{SKey} \oplus h(M_3^f \| M_6^f \| L_6^k \| T_3)$;
      $L_{11}^k \leftarrow h(\texttt{SKey} \| T_2 \| T_3)$;
      $L_{12}^k \leftarrow (L_{10}^k, L_{11}^k, T_3)$;
      $\texttt{SndOpn}(L_{12}^k, \texttt{TA}_2)$;
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;
procedure $\texttt{TA}_2(L_{12}^k)$
  $(L_{10}^k, L_{11}^k, T_3) \leftarrow L_{12}^k$;
  $T_4 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_4 - T_3| \leq \Delta T$ **then**
    $\texttt{SKey} \leftarrow L_{10}^k \oplus h(h((h(\mathcal{I}^f \oplus n_2^f) \oplus K_{DT}) \| M_6^f \| L_6^k \| T_3)$;
    $L_{11}^{k*} \leftarrow h(\texttt{SKey} \| T_2 \| T_3)$;
    if $L_{11}^{k*} == L_{11}^k$ **then**
      $u_4^k \leftarrow \texttt{GenRan}()$;
      $L_{13}^k \leftarrow h(r_2^k \| L_{11}^{k*} \| T_4 \| T_2 \| T_1 \| n_3^v)$;
      $L_{14}^k \leftarrow \texttt{SKey} \oplus h(L_{11}^{k*} \| p\mathcal{I}_1^v \| T_4)$;
      $L_{15}^k \leftarrow L_{13}^k \oplus h(p\mathcal{I}_1^v \| n_3^v \| L_{11}^{k*} \| T_4)$;
      $L_{16}^k \leftarrow h(\texttt{SKey} \| L_{13}^k \| T_4)$;
      $L_{17}^k \leftarrow (L_{14}^k, L_{15}^k, L_{16}^k, T_4)$;
      $\texttt{SndOpn}(L_{17}^k, \texttt{vehicle}_2)$;
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;
procedure $\textsc{vehicle}_2(L_{17}^k)$
  $(L_{14}^k, L_{15}^k, L_{16}^k, T_4) \leftarrow L_{17}^k$;
  $T_5 \leftarrow \texttt{Time\_Stamp}()$;
  if $|T_5 - T_4| \leq \Delta T$ **then**
    $\texttt{SKey} \leftarrow L_{14}^k \oplus h((h(M_2^{k*} \| T_1) \oplus L_3^k) \| p\mathcal{I}_1^v \| T_4)$;
    $L_{13}^k \leftarrow L_{15}^k \oplus h(p\mathcal{I}_1^v \| n_3^v \| (h(M_2^{k*} \| T_1) \oplus L_3^k) \| T_4)$;
    $L_{16}^{k*} \leftarrow h(\texttt{SKey} \| L_{13}^k \| T_4)$;
    if $L_{16}^{k*} == L_{16}^k$ **then**
      Session is authenticated.
    else
      $\texttt{Abort}$;
  else
    $\texttt{Abort}$;

the session key $SKey_2$ be the key established among V, D, and TA. In this case, D encrypts $SKey_1$ with $SKey_2$ and sends the result to TA. Hence, TA can securely communicate with S via its session key. Now suppose that V wants to communicate with S. In this case, D encrypts $SKey_2$ with $SKey_1$ and sends the encrypted key to S. Hence, the

sensor can securely communicate with V via its session key. Figure 2 visualizes this Sensor-2-Vehicle communication case.

It is worth noting that HASFAV provides partial credentials backup via the credentials preserved in field side servers. This also enables side servers to play (at least partially) the role of TA at times of emergency. Hence, HASFAV implicitly provides a backup plan for establishing session keys.

4.4. Addition of New Sensor Phase

In this subsection we present the process for adding a new sensor s^{new} to an existing system. The process starts by s^{new} choosing an ID $\mathcal{I}^{s^{new}}$ and generating a random secret $n_1^{s^{new}}$. Then s^{new} calculates $M_1^{s^{new}}, M_2^{s^{new}}$ and $M_3^{s^{new}}$ as $h(\mathcal{I}^{s^{new}} \oplus n_1^{s^{new}}), M_1^{s^{new}} \oplus \mathcal{I}^{s^{new}}$, and $M_1^{s^{new}} \oplus K_{DT}$, respectively. Then s^{new} sends the message $(M_2^{s^{new}}, M_3^{s^{new}})$ to the TA via a secure channel. The TA extracts $M_1^{s^{new}}$ from $M_3^{s^{new}}$ and $\mathcal{I}^{s^{new}}$ from $M_2^{s^{new}}$. Next, the TA generates a random secret $n_2^{s^{new}}$. This secret and the key K_{DT} are used to calculate $M_5^{s^{new}}$ and $M_6^{s^{new}}$ as $h(\mathcal{I}^{s^{new}} \| K_{DT} \| n_2^{s^{new}})$, and $h(M_1^{s^{new}} \| \mathcal{I}^{s^{new}} \| K_{DT} \| n_2^{s^{new}})$, respectively. Then the TA stores $(\mathcal{I}^{s^{new}}, n_2^{s^{new}}, M_1^{s^{new}}, K_{DT})$ into its secure memory and sends them via a secure channel to the field server in charge of the sensor zone. Finally, the TA sends the message $(M_5^{s^{new}}, M_6^{s^{new}})$ to s^{new} via a secure channel.

5. Security Analysis

In this section we analyze the security of HASFAV. This is done considering the widely-used RORM (Real-Or-Random model) [2] and via applying formal and non-mathematical analysis. We show that HASFAV is secure and resilient to session-key and other possible attacks. In particular, we apply the RORM to prove that the session keys built by HASFAV are secure. Our protocol involves several parties, namely sensors, servers, drones, and agricultural vehicles. The RORM considers an adversary $\mathcal{A}$ that can modify, insert, delete, and know communicated messages in the system. The RORM uses queries to simulate real attacks. These queries are as follows:

- Communicate: $\mathcal{A}$ can communicate with participants by sending and receiving messages.
- SCardManipulation: $\mathcal{A}$ can get smart card data of a legitimate drone's or vehicle's user.
- SKGain: $\mathcal{A}$ can see the session key of a running session.
- Listen: $\mathcal{A}$ eavesdrops on the communicated messages on public channels.
- Check: This is the query for checking the security of session keys according to the RORM. The result of this query is taken based on the result of flipping an unbiased coin C before starting games (flipping of coin). If the adversary executes *Check*, then the result is as follows:

$$Check = \begin{cases} \text{a random number} & \text{if } C = 1, \\ \text{a session key} & \text{if } C = 0, \\ \text{Null} & \text{Otherwise.} \end{cases}$$

The adversary $\mathcal{A}$ (as well as any other widely communicating party, such as the agricultural autonomous vehicle) has access to a collision-resistant hash function (H) that models a random oracle. We call each use case of this function an H query.

User passwords typically follow Zipf's law [45]. This menas that it is convenient to follow Zipf's law in verifying the security of session keys. Our analysis starts with the following definition:

Definition 1. *The session key security of HASFAV is equal to the advantage (probability) Ad of an adversary $\mathcal{A}$ to successfully, in polynomial time, obtain the session key. This is subject to $\mathcal{A}$ utilizing only message eavesdropping RORM queries.*

Theorem 1. *Assume:*

1. *At time t, $\mathcal{A}$ attacks HASFAV and tries to obtain two session keys, namely $SKey_1$ and $SKey_2$, generated by Algorithms 3 and 4, respectively.*
2. *$\mathcal{H}_n, C_n$, and $\mathcal{R}_n$ are numbers of the H and Communicate queries and the range space of $h(\cdot)$, respectively.*
3. *$\mathcal{Z}_1$ and $\mathcal{Z}_2$ are Zipf's parameters [45] contributing to distribution calculations of user-generated passwords.*

Then, the average probability of $\mathcal{A}$ in breaking the security of HASFAV and hence getting $SKey_1$ and $SKey_2$ is limited by:

$$\mathcal{P}(t, (SKey_1, SKey_2)) \leq 2.Z_2.C_n^{Z_1} + \frac{(\max\{\mathcal{H}_n^{SKey_1}, \mathcal{H}_n^{SKey_2}\})^2}{\mathcal{R}_n}. \quad (1)$$

Proof. *The proof of the security of the session key is developed using a series of four games denoted by $\{\mathcal{G}_i \mid i \in \{1, 2, 3, 4\}\}$. In game $\mathcal{G}_i$, we let a_i denote the action of $\mathcal{A}$ predicting successfully and correctly a bit b of session key $SKey_1$ or $SKey_2$. The probability of a_i occurring is denoted by $\mathcal{P}(a_i)$. We also let $\mathcal{P}(\mathcal{G}_i)$ denote the probability of $\mathcal{A}$ to win $\mathcal{G}_i$. Hence $\mathcal{P}(\mathcal{G}_i) = \mathcal{P}(a_i)$.*

$\mathcal{G}_1$. An attack on the RORM is launched in the first game. The attack is executed by the random selection of a bit b at the start of the game. As a result, we have

$$\mathcal{P}(t, SKey_1) = |2(\mathcal{W}(\mathcal{G}_1) - 1)| = |2(\mathcal{P}(a_1)) - 1|, \text{and} \quad (2)$$

$$\mathcal{P}(t, SKey_2) = |2(\mathcal{W}(\mathcal{G}_1) - 1)| = |2(\mathcal{P}(a_1)) - 1|. \quad (3)$$

$\mathcal{G}_2$. The second game considers eavesdropping attacks in which $\mathcal{A}$ obtains communicated messages among system parties. Therefore, for this game $\mathcal{A}$ knows:

1. *$N_4^k = (N_2^k, N_3^k, p\mathcal{I}_1^d, T_1)$ from drone to server.*
2. *$N_8^k = (N_6^k, N_7^k, T_2)$ from server to sensor.*

3. $N_{12}^k = (N_{10}^k, N_{11}^k, T_3)$ *from sensor to server.*
4. $N_{17}^k = (N_{14}^k, N_{15}^k, N_{16}^k, T_4)$ *from server to drone.*
5. $L_4^k = (L_2^k, L_3^k, p\mathcal{I}_1^v, T_1)$ *from vehicle to trust authority.*
6. $L_8^k = (L_6^k, L_7^k, T_2)$ *from trust authority to server.*
7. $L_{12}^k = (L_{10}^k, L_{11}^k, T_3)$ *from server to trust authority.*
8. $L_{17}^k = (L_{14}^k, L_{15}^k, L_{16}^k, T_4)$ *from trust authority to vehicle.*

Running `HASFAV` *while utilizing the* `Listen` *query can lead to this game situation. Afterwards* $\mathcal{A}$ *can test the validity of the obtained session keys via the* `SKGain` *and* `Check` *queries. Random values (such as* u_3^k *and* r_3^k*), timestamps (such as* T_3*), and long term secret keys that are not revealed to* $\mathcal{A}$ *are used in calculating session keys* $SKey_1$ *and* $SKey_2$*. Hence eavesdropping on the exchanged messages can not lead* $\mathcal{A}$ *to have a winning probability of this game or increase the success chance of this game. Therefore, the winning probabilities of this and previous games are not distinguishable. Therefore,*

$$\mathcal{P}(\mathcal{G}_1) = \mathcal{P}(\mathcal{G}_2). \tag{4}$$

$\mathcal{G}_3$*. This game simulates the modeling of the H query as an active game. The communicated messages of the protocol, namely* $N_4^k, N_8^k, N_{12}^k, N_{17}^k, L_4^k, L_8^k, L_{12}^k$*, and* L_{17}^k *are hashed-map-protected. Moreover, timestamps and random numbers contribute to other calculations of* `HASFAV`*. It is worth noting that obtaining a nonce from communicated messages is typically not feasible computationally, when the hash map is collision-resistant. Therefore, this game is indistinguishable from the previous one, except that this game includes the* `H` *query. By applying the birthday paradox, we get:*

$$|\mathcal{P}(\mathcal{G}_2) - \mathcal{P}(\mathcal{G}_3)|_{SKey_1} \leq \frac{(\mathcal{H}_n^{SKey_1})^2}{2\mathcal{R}_n}. \tag{5}$$

$$|\mathcal{P}(\mathcal{G}_2) - \mathcal{P}(\mathcal{G}_3)|_{SKey_2} \leq \frac{(\mathcal{H}_n^{SKey_2})^2}{2\mathcal{R}_n}. \tag{6}$$

Therefore, in general we have

$$|\mathcal{P}(\mathcal{G}_2) - \mathcal{P}(\mathcal{G}_3)| \leq \frac{(\max\{\mathcal{H}_n^{SKey_1}, \mathcal{H}_n^{SKey_2}\})^2}{2\mathcal{R}_n}. \tag{7}$$

$\mathcal{G}_4$*. This final game of the proof builds on* $\mathcal{G}_3$ *by executing the* `Communicate` *and* `SCardManipulation` *queries. Hence, by launching power analysis attacks,* $\mathcal{A}$ *can know the drone and vehicle credentials, i.e.,* $(M_6^d, M_7^d, M_8^d, n_3^d)$*, and* $(M_6^d, M_7^d, M_8^d, n_3^d)$*, respectively. However, to obtain the secret values* n_1^d, n_2^d, n_1^v*, and* n_2^v *from the obtained credentials,* $\mathcal{A}$ *has to have previous knowledge of* $\mathcal{I}_1^d, \mathcal{I}_1^v$*,and* K_D*. Hence it is not feasible computationally for* $\mathcal{A}$ *to obtain the legitimate parameters (passwords). Accordingly, this game is pretty much similar to the previous one, and running the* `Communicate` *query is enough for* $\mathcal{A}$ *to guess the bit b and win the game. Therefore*

$$\mathcal{P}(\mathcal{G}_4) = 0.5. \tag{8}$$

Bearing in mind that the system tolerates only limited attempts of wrong passwords and by applying Zipf's passwords law [45], we have:

$$|\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4)| \leq Z_2.C_n^{Z_1}. \tag{9}$$

Therefore, we have

$$\begin{aligned}
&\frac{1}{2}\mathcal{P}(t, (SKey_1, SKey_2)) \\
&= |(\mathcal{P}(\mathcal{G}_1) - \frac{1}{2})|, \text{ by Equation 2 and 3} \\
&= |(\mathcal{P}(\mathcal{G}_2) - \frac{1}{2})|, \text{by Equation 4} \\
&= |(\mathcal{P}(\mathcal{P}_2) - \mathcal{P}(\mathcal{G}_4))|, \text{ by Equation 8} \\
&\leq |(\mathcal{P}(\mathcal{G}_2) - \mathcal{P}(\mathcal{G}_3))| + |(\mathcal{P}(\mathcal{G}_3) - \mathcal{P}(\mathcal{G}_4))| \\
&\leq Z_2.C_n^{Z_1} + |(\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4))|, \text{ by Equation 9} \\
&\leq Z_2.C_n^{Z_1} + \frac{(\max\{\mathcal{H}_n^{SKey_1}, \mathcal{H}_n^{SKey_2}\})^2}{2\mathcal{R}_n}, \text{ by Equation 5.}
\end{aligned} \tag{10}$$

and

$$\mathcal{P}(t, (SKey_1, SKey_2)) \leq 2.Z_2.C_n^{Z_1} + \frac{(\max\{\mathcal{H}_n^{SKey_1}, \mathcal{H}_n^{SKey_2}\})^2}{\mathcal{R}_n}. \tag{11}$$

This completes the proof.

We now prove the security features and the resilience of `HASFAV` against other attacks. We use drones/vehicles in our proofs. However, the results apply to all system entities.

Theorem 2. 1. `HASFAV` *secures anonymity and resists tracing attacks.*
2. `HASFAV` *is secure against guessing attacks.*
3. `HASFAV` *is secure against impersonation attacks.*
4. `HASFAV` *is secure against ESL (ephemeral secret leakage) attacks.*

Proof. 1. *We assume that the adversary* $\mathcal{I}_1^x$ *tries to learn the identity of the drone/vehicle. In* `HASFAV` *protocol, this identity,* $\mathcal{I}_1^x$ *is combined with a random value* n_1^x*. Also, the drone/vehicle communicates messages via the secret* $n3^x$ *received from TA (and stored in the field server). Moreover, the field server builds a new secret* N_{13}^k *from* $n3^x$ *in the authentication phase, to increase resistance to adversary guessing. It is worth noting that all communicated messages are salted with timestamps and random numbers. Therefore communicated messages are continuously changing in all sessions. Hence, we conclude that the adversary cannot reveal the identity of the vehicle/-drone; this completes the proof.*

2. *The drone/vehicle credentials* $\mathcal{I}_1^x$ *and* $\mathcal{I}_2^x$ *can not be guessed by an adversary* $\mathcal{A}$*. However,* $\mathcal{A}$ *may apply power analysis techniques to learn drone/vehicle credentials* $(M_6^x, M_7^x, M_8^x, n_3^x)$ *from the smart card of a stolen entity. Even these credentials will not help* $\mathcal{A}$ *to learn secret credentials as* M_6^x, M_7^x*, and* M_8^x *are secured with random strings and with the collision-resistant one-way hash function* $h(.)$*. Also, sub-operators of the credentials that the adversary obtained are secured with the secret*

key K_{DT} and the parameters $\mathcal{I}_1^x$ and $\mathcal{I}_2^x$. Moreover, guessing times of survived brute-force and off-line guessing attacks are limited by virtue of the honey list technique. Therefore, **HASFAV** *can withstand guessing attacks.*

3. *For an adversary $\mathcal{A}$ to impersonate a system entity, $\mathcal{A}$ needs to generate legitimate messages. However, communicated messages of* **HASFAV** *are secured with random numbers including n^x, u^k, r^k and secret keys, including K_{DT}; $\mathcal{A}$ can not obtain these secrets. Moreover, L_7^k and L_{11}^k are encrypted by using random numbers and the secret key K_{DT}. Hence,* **HASFAV** *can protect system entities against impersonation attacks.*
4. *Several entities contribute to computing the session keys* **SKey** $= h(L_9^k \parallel h(u_3^k \parallel M_3^f) \parallel T_3)$ *and* **SKey** $= h(N_9^k \parallel h(r_3^k \parallel M_3^s) \parallel T_3)$*. Therefore, ephemeral secrets u_3^k and r_3^k contribute to computing session keys. Hence, according to the CK-threat model, compromising these secrets is not enough for the adversary to know the session keys. This is justified by the fact that no values that contribute to the computation of session keys are sent over public channels. Moreover, all messages including* **SKey** *are also implicitly encrypted with n_2^s and K_{DT}. Hence, the long-term key K_{DT} is as well encrypted with n_2^s. Therefore, compromising session keys requires compromising long and short-term secret credentials. Since it is not computationally possible to compromise these credentials,* **HASFAV** *is secure against ESL attacks.*

Theorem 3. 1. **HASFAV** *achieves perfect forward secrecy, mutual authentication, and key agreement properties and* **HASFAV** *is secure against session key disclosure attacks.*
2. **HASFAV** *is secure against replay, man-in-the-middle, and privileged insider attacks.*

Proof. 1. *We assume that the adversary $\mathcal{A}$ knows the key K_{DT} and tries to compute session keys. It is worth noting that K_{DT} is wrapped with the hash function and random value n_2^x. Therefore $\mathcal{A}$ does not know necessary sensitive parameters to generate session keys. Furthermore, all communicated messages are changing in every session due to the use of timestamps and random numbers. Hence,* **HASFAV** *preserves perfect forward secrecy. All system entities authenticate each other in* **HASFAV** *via checking $N_2^k, N_7^k, N_{11}^k, N_{16}^k, L_2^k, L_7^k, L_{11}^k$, and L_{16}^k. Furthermore, all communicated messages change from one session to another using timestamps and random numbers. Therefore,* **HASFAV** *guarantees mutual authentication and key agreement. The above reasoning also proves that* **HASFAV** *resists session key disclosure attacks.*

2. *Suppose that the adversary $\mathcal{A}$ can obtain messages communicated in public channels and also smart card information of drone/vehicle. However, as discussed above (Theorem 2.3), it is not computationally possible for $\mathcal{A}$ to generate a legitimate login request. Furthermore, in all sessions, the communicated messages are protected by random numbers and timestamps. Hence, system entities cannot be impersonated and* **HASFAV** *is resilient to man-in-the-middle and replay attacks. We assume that a privileged-insider entity of the system is an adversary $\mathcal{A}$ and knows registration information $(p\mathcal{I}_1^x, p\mathcal{I}_2^x \oplus n_1^x)$. We also suppose that $\mathcal{A}$ obtains smart card information $(M_6^x, M_7^x, M_8^x, n_3^x)$, using power analysis attacks. Even under all these strong assumptions, $\mathcal{A}$ will not be able to generate vehicle/drone identity parameters $(\mathcal{I}_1^x, \mathcal{I}_2^x \oplus n_1^x)$ due to the the use of secret keys and random numbers n_1^x and K_{DT}. Therefore,* **HASFAV** *is resilient to privileged-insider attacks.*

6. Verification and Evaluation

In this section, we present the results of the experimental evaluation of HASFAV. The evaluation involved extensive experiments aiming at comparing HASFAV to related state-of-the-art schemes [4, 48, 37, 40, 44]. We used AVISPA, Automated Validation of Internet Security Protocols and Applications [9] to show formally that HASFAV is secure against replay and man-in-the-middle attacks. This was done while considering passive and active adversary communications. The details of AVISPA verification are presented in Section 6.1.

The security attributes and functionality features are among the most important parameters of authentication protocols. These parameters also include computation, communication, and energy consumption costs. We carried a precise comparison for HASFAV against state-of-the-art schemes on these parameters. The results of these comparisons show the Superiority of HASFAV compared to these schemes. Section 6.2 presents these comparisons in details.

We also carried out a precise implementation of HASFAV to prove its practical perspective. The implementation was achieved using a well-established networking simulation tool, Omnetpp[1]. We also used another famous Omnetpp simulation environment, Castalia[2] simulator. Section 6.3 shows the results and details of the Omnetpp implementation. We performed the experiments on a Dell (Vostro) Intel(R) Core(TM) i7-3612 QM CPU @ 2.10 GHz, 8.00 GB RAM on Windows 10 (64-bits) OS. Files we obtained from different simulations tools that we used to evaluate HASFAV are available in a repository[3].

6.1. FORMAL SECURITY VERIFICATION

We used the AVISPA tool to simulate HASFAV and verify its security against replay and man-in-the-middle attacks. One reason that makes AVISPA convenient for our simulation is that AVISPA relies on the DY-threat model [20], as does HASFAV. Four backends are offered by AVISPA:

- CL-AtSe: Constraintlogic based Attack Searcher,

[1] https://omnetpp.org/
[2] https://github.com/boulis/Castalia
[3] https://github.com/maelzawawy/HASFAV

- OFMC: On-the-fly mode-checker,
- TA4SP: Tree Automata based on Automatic Approximations for the Analysis of Security Protocols, and
- SATMC: SAT-based Model Checker.

Only OFMC and CL-AtSe backends implement XOR operation (bitwise exclusive OR). Therefore, our simulation relies on these backends. AVISPA is equipped with HLPSL, High-Level Protocol Specification Language. HLPSL facilitates the implementation process utilizing the "role" concept. For HASFAV, we have a role per system entity. Therefore, our implementation defined a sensor role, a TA role, a server role, an agriculture vehicle rule, and a drone role. Moreover, the simulation uses sessions that are composed of composite roles, a goal, and an environment.

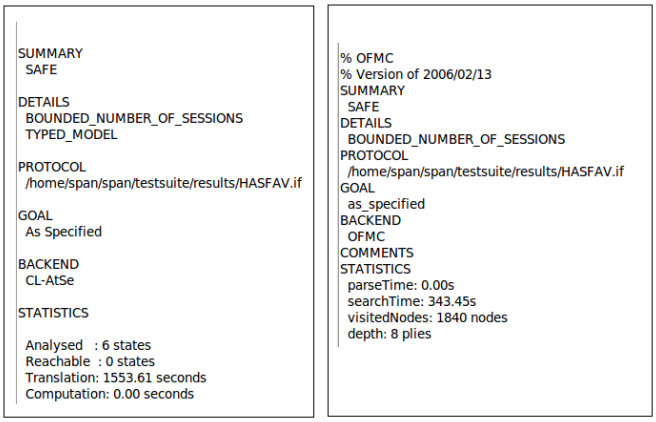

Figure 3: AVISPA Simulation Results for HASFAV.

Listing 1: Part of the drone code in AVISPA.

```
State=0 /\ RC1(start) =|>
State'=1 /\ Ix1'=new() /\ S1'=new()
/\ Ix2'= H(xor( Ix1',S1')) /\ Nx1'=new()
/\ PIx1'= H(Ix1'.Nx1')
/\ PIx2'= Hb(Ix2'.xor( Nx1',PIx1'))
/\ Mx1'= PIx1'.xor(H(Ix2'.xor( Nx1',PIx1')),Nx1')
```

The results of our simulation are shown in Figure 3. The results confirm that our proposed protocol, HASFAV, is resilient to man-in-the-middle and replay attacks [49]. These result files are available in a repository[4]. Listing 1 presents a part of the simulation code of the drone. In the code, the drone transfers from the initial state (State=0) to a new state (State'=1).

6.2. Functionality and Costs

In this section, we present the results of comparing HASFAVagainst related existing state-of-the-art schemes [4, 48, 37, 40, 44]. The security attributes and functionality features as well as computation, communication, and energy consumption costs are included in the comparisons.

[4]https://github.com/maelzawawy/HASFAV

The comparisons of computation costs are based on the login and authentication phase. We use the execution times (denoted by t_h) of one-way hash map $H(.)$ reported in [44, 27]. One common example of the hash map is SHA-256. On average, each execution of an one-way hash map $H(.)$ takes 0.055 ms on a server. On average, for building a session key, a vehicle, a drone, a sensor node, a field-side server in HASFAVexecutes 10, 11, 6, and 13 hash maps, respectively. The average comparable (only including entities treated by all protocols) total computational cost for HASFAV is 0.8525 ms. Table 2 shows computational costs needed for each system component to build a session key for HASFAV and related protocols. This table shows that, on average, HASFAV is more efficient regarding the computational cost than are state-of-the-art protocols.

Among the main factors affecting the energy consumption of HASFAV are the size and speed of messages communicated in the system during the protocol execution. In agricultural applications, messages among sensors and servers have more weight than other parties' messages. Typically, the message exchange is controlled by control units of system entities and is accomplished in the physical protocol layer of the network. It is common for entities of IoT systems, like the one we focus on in this paper, to follow IEEE 802.11p for data transmissions. For IoT network systems the following parameter values of the IEEE standards are convenient. The frequency, channel bandwidth, data rate ($\mathcal{R}_d$), and transmit power, can be assigned the values 5.8 GHz, 10 MHz, 6 Mbps, and 25 dBm, respectively [30, 24, 27]. Two main parameters contribute to the total energy consumption cost for generating a session key i in HASFAV. These parameters are denoted by $\mathcal{J}_k^i$ and $\mathcal{J}_l^i$ [30, 24, 27]. The former captures energy consumed for key generation and the latter captures energy consumed in the login and authentication phase. The calculation of the total energy consumption is performed as follows:

$$\mathcal{J} = \frac{\sum_{i \in \mathsf{SeKey}} (\mathcal{J}_k^i + \mathcal{J}_l^i)}{|\mathsf{SeKey}|}, \text{ and } \quad \mathcal{J}_k^i = \mathcal{P}_c^i \times \mathcal{P}_{cpu} \tag{12}$$

$$\mathcal{J}_l^i = \frac{\mathcal{M}_s^i \times \mathcal{P}_{cpu}}{\mathcal{R}_d} \tag{13}$$

where we let $\mathcal{P}_{cpu}, \mathcal{P}_c$, and $\mathcal{M}_s$ denote CPU maximum power, total computational cost, and message size, respectively. Our calculations rely on assigning $\mathcal{P}_{cpu}$ the standard value 10.88 W of wireless systems [30, 24, 27]. We present the energy consumption of HASFAV and related state-of-the-art schemes in Table 3. Regarding energy consumption, our experiments prove that HASFAV is more efficient than related state-of-the-art schemes.

Table 4 presents security attributes of HASFAVand functionality offered by HASFAV. The table confirms that HASFAV provides more functionality required in modern smart farming than other state-of-the-art protocols do. The table also confirms that HASFAV has better security attributes than these protocols.

Table 2: Computational cost of HASFAV and of state-of-the art protocols.

#	Participant	[4]	[40]	[44]	HASFAV($SKey_1$)	HASFAV($SKey_2$)
1	Sensor node	$11 \times t_h$	$13 \times t_h$	$9 \times t_h$	$6 \times t_h$	–
2	Field side server	$8 \times t_h$	$7 \times t_h$	$9 \times t_h$	$19 \times t_h$	$6 \times t_h$
3	Comparable Total [ms]	$19 \times t_h$ ≈ 1.045	$20 \times t_h$ ≈ 1.1	$18 \times t_h$ ≈ 0.99	$25 \times t_h$ ≈ 1.375 ≈ 0.8525 (average)	$6 \times t_h$ ≈ 0.33
4	Drone	–	–	–	$11 \times t_h$	–
5	Autonomous agricultural vehicles	–	–	–	–	$10 \times t_h$

The symbol – denotes that the corresponding protocol does involve the corresponding participant.

Table 3: Energy consumption of HASFAV and of state-of-the art protocols.

#	Parameter	[4]	[40]	[44]	HASFAV($SKey_1$)	HASFAV($SKey_2$)
1	$\mathcal{J}_k^i$ [mJ]	11.369	11.968	10.771	14.96	1.184
2	$\mathcal{J}_l^i$ [mJ]	9.980	13.810	4.177	4.874	4.874
3	$\sum_{i \in \mathtt{SeKey}}(\mathcal{J}_k^i + \mathcal{J}_l^i)$ [mJ]	21.349	25.778	14.948	19.834	6.058
4	$\mathcal{J}$ [mJ]	21.349	25.778	14.948	12.946	

Table 4: Functionality offered by HASFAV and security attributes of HASFAV and of state-of-the art protocols.

#	Functionality Attribute	[4]	[40]	[37]	[44]	HASFAV
1	Supporting autonomous agricultural vehicles.	×	×	×	×	✓
2	Supporting honey-list utilization.	×	×	×	×	✓
3	Supporting internet of drones in smart farming.	×	×	×	×	✓
4	Supporting partial credentials backup in field side servers.	×	×	×	×	✓
5	Providing implicit backup plan for session keys establishment.	×	×	×	×	✓
6	Anonymity and untraceability.	×	✓	×	✓	✓
7	Resilience to offline guessing attacks.	✓	×	✓	✓	✓
8	Resilience to man-in-the-middle attacks and supports mutual authentication.	✓	✓	✓	✓	✓
9	Resilient replay attacks.	✓	✓	✓	✓	✓

The symbol ✓ denotes that the corresponding protocol supports the corresponding attribute.

The symbol × denotes that the corresponding protocol does not support the corresponding attribute.

To compute the communication cost of HASFAV, we follow the common literature assumption of assuming sizes of the hash output (SHA-256 hashing), random values, timestamp, and identity as 256, 160, 32, and 160 bits, respectively. As shown in Table 5, HASFAV implies a communication cost of 2688 bits while exchanging four messages for building each of the session keys. On the other hand, the other protocols that we compare against, namely [4], [40], and [37], imply five messages (5504 bits), four messages (7616 bits), and four messages (5248 bits), respectively. Therefore, the communication costs of HASFAV are less than that of these related state-of-the-art schemes.

6.3. Practical Perspective

We evaluated practical aspects of HASFAVby implementing it in a widely-accepted simulation tool for networks. The used tool is omnetpp[5], augmented with the well-known Castalia[6] simulator. Castalia output files of our experiments are available in repository[7]. The command `CastaliaResults` used on the result files can produce information concerning the configurations of experiments and other metadata. Our Omnetpp experiments consider characteristics of smart farming, agricultural smart vehicles, and drones described in Section 3.3.

The following are the configurations of our Omnetpp-Castalia experiments. We simulated a network managing a farming area whose dimensions are 50 m and 200 m. Hence, the field is 50 $m \times 200$ m. We deployed one drone, one autonomous vehicle, one trust author-

[5]https://omnetpp.org/

[6]https://github.com/boulis/Castalia

[7]https://github.com/maelzawawy/HASFAV

Table 5: Communication cost of HASFAV and of state-of-the art protocols.

#	Parameter	[4]	[40]	[37]	HASFAV($SKey_1$)	HASFAV($SKey_2$)
1	No. of messages	5	4	4	4	4
2	No. of bits	5504	7616	5248	2688	2688

Table 6: Notations used in the Omnetpp-Castalia experiments.

Notation	Semantics
$\mathcal{I}_s$	Scenarios ID
$\mathcal{N}_e$	Number of system entities
$\mathcal{S}_v$	Speed of autonomous vehicle (MPH)
$\mathcal{S}_d$	Speed of assisting drone (MPH)
$\mathcal{S}e_1$	Time required to establish a session
$\mathcal{S}e_2$	Time required to establish two sessions
$\mathcal{E}$	Energy consumed, on average, per system entity [mJ]
$\mathcal{T}$	Number of packets transmitted on average per entity.
$\mathcal{R}$	Number of packets received, on average, per entity.

ity, two field-side servers, and 100 − 150 IoT sensor devices. The devices are deployed uniformly across the farming field (in a matrix form). The side servers are deployed at coordinates (20, 50) and (20, 150). The drone and the vehicle are deployed at coordinates (0, 25). This makes them move on a line across the middle of the field. The mobility model of the vehicle and the drone is `LineMobilityManager`. The speed of the vehicle and of the drone ranges from 10 mph to 15 mph and from 20 mph to 25 mph, respectively. We apply the communication protocol 802.15.4 MAC. The routing protocol of our simulation is `MultipathRingsRouting`. The bandwidth of channel and noise of our simulation is 20 MHz and 194 MHz, respectively. The data rate and modulation type of our system are 250 KBPS and `PSK`, respectively. Finally, the system noise floor is −100 DBM.

Our Omnetpp-Castalia experiments are based on six scenarios for simulating HASFAVThe parameters used in the scenarios are shown in Table 7; the semantics of the titles of the columns of this table are shown in Table 6. The details of the scenarios are as follows:

$\mathcal{I}_{S1}$: The scenario has 100 IoT sensor devices, 1 trust authority, 1 drone, 2 field-side servers, and 1 agricultural autonomous vehicle. The vehicle and the drone are moving at speeds of 10 MPH and 20 MPH, respectively.

$\mathcal{I}_{S2}$: The second scenario builds on the first one, $\mathcal{I}_{S1}$, by only increasing the drone speed to 25 MPH.

$\mathcal{I}_{S3}$: The third scenario builds on the second one, $\mathcal{I}_{S2}$, by only increasing the vehicle speed to 15 MPH.

$\mathcal{I}_{S4}$: The scenario has 150 IoT sensor devices, 1 trust authority, 1 drone, 2 field-side servers, and 1 agricultural autonomous vehicle. The vehicle and the drone are moving at speeds of 10 MPH and 20 MPH, respectively.

$\mathcal{I}_{S5}$: The fifth scenario builds on the fourth one, $\mathcal{I}_{S4}$, by only increasing the drone speed to 25 MPH.

$\mathcal{I}_{S6}$: This scenario builds on the fifth, $\mathcal{I}_{S5}$, by only increasing the vehicle speed to 15 MPH.

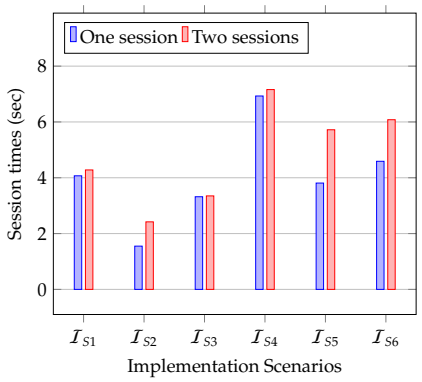

Figure 4: Times needed to construct one session and two sessions in HASFAV implementation scenarios.

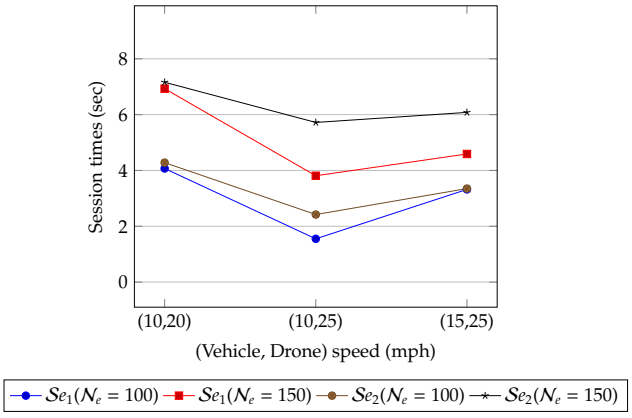

Figure 5: Times needed to construct one session and two sessions in HASFAV for different vehicle and drone speeds.

Table 7 summarizes the results of our Omnetpp-Castalia experiments. These include the time required to establish a session ($\mathcal{S}e_1$), the time required to establish two sessions ($\mathcal{S}e_2$), the average energy consumed per system

Table 7: Scenarios used for testing practical perspectives of HASFAV with their results.

$\mathcal{I}_s$	$\mathcal{N}_e$	$\mathcal{S}_v$	$\mathcal{S}_d$	$\mathcal{S}e_1$	$\mathcal{S}e_2$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{R}$
$\mathcal{I}_{S1}$	100	10	20	4.07	4.28	6.798	58.552	59.162
$\mathcal{I}_{S2}$	100	10	25	1.55	2.42	6.798	65.638	67.515
$\mathcal{I}_{S3}$	100	15	25	3.32	3.35	6.798	72.886	74.333
$\mathcal{I}_{S4}$	150	10	20	6.93	7.16	6.796	375.565	374.045
$\mathcal{I}_{S5}$	150	10	25	3.81	5.72	6.796	275.632	275.006
$\mathcal{I}_{S6}$	150	15	25	4.59	6.08	6.797	188.097	188.316

entity ($\mathcal{E}$), the average number of packets transmitted per entity ($\mathcal{T}$), and the average number of packets received per entity ($\mathcal{R}$). Of these, $\mathcal{S}e_1$ and $\mathcal{S}e_2$ are visualized in Figure 4 for all scenarios. We notice that the extra time needed to establish the second session after establishing the first one is generally much smaller than $\mathcal{S}e_1$. In other words, $\mathcal{S}e_2 - \mathcal{S}e_1 \leq \mathcal{S}e_1$. For the different scenarios these differences are $0.21, 0.87, 0.03, 0.23, 1.91$, and 1.49 sec. On average, this difference is 0.79 sec. This small average proves the practicality of establishing two session keys. Figure 5 shows the times needed to establish session keys for different combinations of vehicle and drone speeds. The figure shows that increasing the speed does not increase the required time for session establishment. This is so as increasing speeds lead to better chances of message arrival. On average, each system entity consumed almost 6.797 mJ. This confirms the good practicality attributes of HASFAV. Furthermore, the closeness in all scenarios between the average (per node) number of received and transmitted packets reported in Table 7 adds to the good practicality characteristics of HASFAV.

7. Conclusion and Future Work

In this paper, we proposed HASFAV a lightweight locality-aware key agreement and authentication protocol, that is provably secure, and robust against various attacks in smart agricultural environments. HASFAV uses a honey list and mutual authentication technologies. We proved the security of HASFAV by performing a security analysis of it, proving three theorems and using a well-established Real-Or-Random (ROR) model. We implemented HASFAV using the AVISPA tool; the results indicate that HASFAVoutperformed all other similar state-of-the-art security protocols.

Interesting directions for future work include the following. Integrating parties receiving the productions of IIoT systems into the system model is becoming a persistent need in real-life applications. This is so as watching the production stages may well affect the plans of these parties. This has to be done conservatively, to preserve the principle of least privilege for the IIoT system. Therefore, it seems promising to integrate a blockchain into the network model to record authentication activities. This can help analyze the history of such activities and hence reveal hidden patterns of malicious actions.

References

[1] Dataprot - 45 fascinating iot statistics for 2021: The state of the industry. https://dataprot.net/statistics/iot-statistics/. Accessed: February 21, 2022.

[2] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In *International Workshop on Public Key Cryptography*, pages 65–84. Springer, 2005.

[3] S Ahirwar, R Swarnkar, S Bhukya, and G Namwade. Application of drone in agriculture. *International Journal of Current Microbiology and Applied Sciences*, 8(01):2500–2505, 2019.

[4] Rifaqat Ali, Arup Kumar Pal, Saru Kumari, Marimuthu Karuppiah, and Mauro Conti. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Generation Computer Systems*, 84:200–215, 2018.

[5] Randa Almadhoun, Maha Kadadha, Maya Alhemeiri, Maryam Alshehhi, and Khaled Salah. A user authentication scheme of iot devices using blockchain-enabled fog nodes. In *2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA)*, pages 1–8. IEEE, 2018.

[6] Saeed Hamood Alsamhi, Faris Almalki, Ou Ma, Mohammad Samar Ansari, and Brian Lee. Predictive estimation of optimal signal strength from drones over iot frameworks in smart cities. *IEEE Transactions on Mobile Computing*, 2021.

[7] Ruhul Amin and G.P. Biswas. A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Netw.*, 36(P1):58–80, jan 2016.

[8] Ruhul Amin, SK Hafizul Islam, G.P. Biswas, Muhammad Khurram Khan, Lu Leng, and Neeraj Kumar. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.*, 101(C):42–62, jun 2016.

[9] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, P Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, et al. The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*, pages 281–285. Springer, 2005.

[10] Basudeb Bera, Anusha Vangala, Ashok Kumar Das, Pascal Lorenz, and Muhammad Khurram Khan. Private blockchain-envisioned drones-assisted authentication scheme in iot-enabled agricultural environment. *Computer Standards & Interfaces*, 80:103567, 2022.

[11] Achilles D Boursianis, Maria S Papadopoulou, Panagiotis Diamantoulakis, Aglaia Liopa-Tsakalidi, Pantelis Barouchas, George Salahas, George Karagiannidis, Shaohua Wan, and Sotirios K Goudos. Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: a comprehensive review. *Internet of Things*, page 100187, 2020.

[12] Jenna Burrell, Tim Brooke, and Richard Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive computing*, 3(1):38–45, 2004.

[13] Ran Canetti and Hugo Krawczyk. Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 337–351. Springer, 2002.

[14] Sravani Challa, Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Alavalapati Goutham Reddy, Eun-Jun Yoon, and Kee-Young Yoo. Secure signature-based authenticated key establish-

ment scheme for future iot applications. *IEEE Access*, 5:3028–3043, 2017.

[15] Chin-Chen Chang and Hai-Duong Le. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Transactions on Wireless Communications*, 15(1):357–366, 2016.

[16] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[17] Rahul Dagar, Subhranil Som, and Sunil Kumar Khatri. Smart farming–iot in agriculture. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1052–1056. IEEE, 2018.

[18] Ashok Kumar Das, Saru Kumari, Vanga Odelu, Xiong Li, Fan Wu, and Xinyi Huang. Provably secure user authentication and key agreement scheme for wireless sensor networks. *Security and Communication Networks*, 9(16):3670–3687, 2016.

[19] Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Athanasios V Vasilakos, and Joel JPC Rodrigues. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment. *IEEE Internet of Things Journal*, 5(6):4900–4913, 2018.

[20] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[21] Moneer Fakroon, Mohammed Alshahrani, Fayez Gebali, and Issa Traoré. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things*, 9:100158, 2020.

[22] Mohammad Sabzinejad Farash, Muhamed Turkanović, Saru Kumari, and Marko Hölbl. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. *Ad Hoc Netw.*, 36(P1):152–176, jan 2016.

[23] Maanak Gupta, Mahmoud Abdelsalam, Sajad Khorsandroo, and Sudip Mittal. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access*, 8:34564–34584, 2020.

[24] Daojing He, Chun Chen, Sammy Chan, and Jiajun Bu. Secure and efficient handover authentication based on bilinear pairing functions. *IEEE Transactions on Wireless Communications*, 11(1):48–53, 2011.

[25] Khalid Hussain, NZ Jhanjhi, Hafiz Mati ur Rahman, Jawad Hussain, and Muhammad Hasan Islam. Using a systematic framework to critically analyze proposed smart card based two factor authentication schemes. *Journal of King Saud University - Computer and Information Sciences*, 33(4):417–425, 2021.

[26] Hyunbum Kim, Jalel Ben-Othman, Lynda Mokdad, Junggab Son, and Chunguo Li. Research challenges and security threats to ai-driven 5g virtual emotion applications using autonomous vehicles, drones, and smart devices. *IEEE Network*, 34(6):288–294, 2020.

[27] JoonYoung Lee, GeonHwan Kim, Ashok Kumar Das, and YoungHo Park. Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks. *IEEE Transactions on Network Science and Engineering*, 8(3):2412–2425, 2021.

[28] Yanrong Lu, Lixiang Li, Haipeng Peng, and Yixian Yang. An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks. *Sensors*, 16(6):837, Jun 2016.

[29] Thomas S Messerges, Ezzat A Dabbish, and Robert H Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE transactions on computers*, 51(5):541–552, 2002.

[30] Zeeshan Hameed Mir and Fethi Filali. Lte and ieee 802.11 p for vehicular networking: a performance evaluation. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):1–15, 2014.

[31] Hossein Mousazadeh. A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50(3):211–232, 2013.

[32] Anandarup Mukherjee, Sudip Misra, Anumandala Sukrutha, and Narendra Singh Raghuwanshi. Distributed aerial processing for iot-based edge uav swarms in smart farming. *Computer Networks*, 167:107038, 2020.

[33] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys Tutorials*, 21(3):2702–2733, 2019.

[34] Nikolaos-Foivos Polychronou, Pierre-Henri Thevenon, Maxime Puys, and Vincent Beroulle. A comprehensive survey of attacks without physical access targeting hardware vulnerabilities in iot/i-iot devices, and their detection mechanisms. *ACM Trans. Des. Autom. Electron. Syst.*, 27(1), sep 2021.

[35] Ali Roshanianfard, Noboru Noguchi, Hiroshi Okamoto, and Kazunobu Ishii. A review of autonomous agricultural vehicles (the experience of hokkaido university). *Journal of Terramechanics*, 91:155–183, 2020.

[36] Akmal Rustamov and Kongratboy Sharipov. Implementation assessments of ero-glonass navigation system in agricultural autonomous vehicles in the territory of uzbekistan. *Acta of Turin Polytechnic University in Tashkent*, 9(3):28–31, 2019.

[37] Dipanwita Sadhukhan, Sangram Ray, GP Biswas, Muhammad Khurram Khan, and Mou Dasgupta. A lightweight remote user authentication scheme for iot communication using elliptic curve cryptography. *The Journal of Supercomputing*, 77(2):1114–1151, 2021.

[38] Nader Samir Labib, Grégoire Danoy, Jedrzej Musial, Matthias R Brust, and Pascal Bouvry. Internet of unmanned aerial vehicles—a multilayer low-altitude airspace model for distributed uav traffic management. *Sensors*, 19(21):4779, 2019.

[39] Geeta Sharma and Sheetal Kalra. A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-iot applications. *J. Inf. Secur. Appl.*, 42:95–106, 2018.

[40] Mengxia Shuai, Ling Xiong, Changhui Wang, and Nenghai Yu. A secure authentication scheme with forward secrecy for industrial internet of things using rabin cryptosystem. *Computer Communications*, 160:215–227, 2020.

[41] Mengxia Shuai, Nenghai Yu, Hongxia Wang, and Ling Xiong. Anonymous authentication scheme for smart home environment with provable security. *Comput. Secur.*, 86:132–146, 2019.

[42] Koen Tange, Michele De Donno, Xenofon Fafoutis, and Nicola Dragoni. A systematic survey of industrial internet of things security: requirements and fog computing opportunities. *IEEE Communications Surveys & Tutorials*, 22(4):2489–2520, 2020.

[43] Muhamed Turkanović, Boštjan Brumen, and Marko Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, 20(Complete):96–112, 2014.

[44] Anusha Vangala, Anil Kumar Sutrala, Ashok Kumar Das, and Minho Jo. Smart contract-based blockchain-envisioned authentication scheme for smart farming. *IEEE Internet of Things Journal*, 2021.

[45] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, and Gaopeng Jian. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 12(11):2776–2791, 2017.

[46] Ding Wang, Xizhe Zhang, Zijian Zhang, and Ping Wang. Understanding security failures of multi-factor authentication schemes for multi-server environments. *Computers Security*, 88:101619, 2020.

[47] Mohammad Wazid, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, Mauro Conti, and Minho Jo. Design of secure user authenticated key management protocol for generic iot networks. *IEEE Internet of Things Journal*, 5(1):269–282, 2018.

[48] Hsin-Te Wu and Chun-Wei Tsai. An intelligent agriculture network security system based on private blockchains. *Journal of Communications and Networks*, 21(5):503–508, 2019.

[49] SungJin Yu, JoonYoung Lee, KyungKeun Lee, KiSung Park, and YoungHo Park. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors*, 18(10):3191, 2018.

[50] Ouarda Zedadra, Antonio Guerrieri, Nicolas Jouandeau, Giandomenico Spezzano, Hamid Seridi, and Giancarlo Fortino. Swarm intelligence-based algorithms within iot-based systems: A review. *Journal of Parallel and Distributed Computing*, 122:173–187, 2018.