future internet

MDPI

Article

H2O: Secure Interactions in IoT via Behavioral Fingerprinting

Marco Ferretti [1], Serena Nicolazzo [2] and Antonino Nocera [1,*]

[1] Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia PV, Italy; marco.ferretti@unipv.it
[2] Daisy Laboratory, Polytechnic University of Marche, 60131 Ancona AN, Italy; serena.nicolazzo.sn@gmail.com
* Correspondence: antonino.nocera@unipv.it

Abstract: Sharing data and services in the Internet of Things (IoT) can give rise to significant security concerns with information being sensitive and vulnerable to attacks. In such an environment, objects can be either public resources or owned by humans. For this reason, the need of monitoring the reliability of all involved actors, both persons and smart objects, assuring that they really are who they claim to be, is becoming an essential property of the IoT, with the increase in the pervasive adoption of such a paradigm. In this paper, we tackle this problem by proposing a new framework, called H2O (Human to Object). Our solution is able to continuously authenticate an entity in the network, providing a reliability assessment mechanism based on behavioral fingerprinting. A detailed security analysis evaluates the robustness of the proposed protocol; furthermore, a performance analysis shows the feasibility of our approach.

Keywords: IoT; Human to Object Network; behavioral fingerprint; reliability

Citation: Ferretti, M.; Nicolazzo, S.; Nocera, A. H2O: Secure Interactions in IoT via Behavioral Fingerprinting. *Future Internet* **2021**, *13*, 117. https://doi.org/10.3390/fi13050117

Academic Editors: Francesco Lelli and Stefano Modafferi

Received: 24 March 2021
Accepted: 28 April 2021
Published: 30 April 2021

1. Introduction

In this new era where pervasiveness and ubiquity of smart things is becoming part of our every day life, the need of security mechanisms for protecting users and objects increases as well. Classical Internet of Things (hereafter, IoT) is characterized by heterogeneous and connected devices sharing data and providing services to fit people needs. From connected fridges, cars, and so forth to wearable devices for health-care, the IoT creates opportunities in numerous domains.

With the adoption of this new paradigm the network can gain the capabilities of sensing information about the environment, collecting physiological measurements or operational data from sensors and identifying actors, such as: users, animals, events or other things in the environment. Moreover, as a result of the capability of handling data, new ways of processing, communicating and converting them into automated instructions enhance the power and autonomy of the IoT, eliminating the need of constant human interventions [1].

Clearly, in such a diverse, complex and heterogeneous ecosystem, numerous challenges arise. A crucial topic, which is currently interesting a lot of researchers, is entity continuous authentication to ensure reliable and secure communications between objects to humans, objects to objects and human to human [2]. Each entity in the IoT must be able to clearly identify and authenticate other entities to assess that neither a user has been impersonated nor an object hacked, corrupted or stolen before staring any interaction.

This is also important, in order to provide trust, confidentiality, integrity and availability of the information exchanged. Indeed, data sharing in IoT applications raises significant security concerns, with information being sensitive and vulnerable to attacks. For example, in an opportunistic trading scenario [3], sensitive personal information such as user location, mobility patterns and preferences may be used for marketing. Whereas, in healthcare applications [4,5] very sensitive medical information about people are aggregated, analysed and exchanged by smart objects. In critical or monitored workplace environments, continuous surveillance can breach the privacy of employees [6,7]. Therefore, the guarantee

to communicate with a reliable entity can protect the actors of the system from major security threats.

However, this process can be very challenging because of the heterogeneous nature of the entities involved in the IoT, including people, devices with low capabilities in term of processors, memory and data storage and devices with higher capabilities.

Typically, low-power devices differ from smarter ones also because they lack conventional user interfaces in the form of keyboards, mice, and touchscreens. This is due to reduce their cost, but also because often this kind of conventional interfaces are not suitable for the intended applications (think for instance to a smart fabric). Therefore, how objects are conceived and, consequently, how they behave in the IoT is peculiar of their capabilities and equipment and this increases the complexity of the above stated problem.

To make some examples, wearable devices equipped with an inertial measurement unit (IMU) can support biometric authentication based on how people move their limbs throughout the time the device is used. Through machine learning techniques these patterns could be learnt and applied to continuously authenticate the user. In this way, a device could monitor its owner continuously and authenticate the person's legitimacy before allowing him to perform operations. Similarly, user echocardiogram (ECG) signals can be exploited by devices containing photoplethysmogram (PPG) sensor [8]. Others more traditional biometrics-based user identification techniques rely on uniquely personalized features, such as: fingerprint [9], iris [10], or face [11,12]; and perform pattern recognition on these features to allow resources access to a user.

Others object-to-human techniques to continuously identify persons leverage the way people interact with their smartphones and the available applications. Most users have regular behavioral patterns that can be modeled and exploited for continuous recognition of behavioral signatures [13,14]. Therefore, behavioral characteristics of mobile users can allow continuous authentication of a user on a personal device.

On the other hand, also object-to-object communications should be continuously authenticated in the IoT in order to prevent security issues. Typically, remote attestation is a security service through which an object can ascertain the current state of a potentially compromised remote device before interacting with it. Remote attestation includes a variety of algorithms that range from heavy-weight secure hardware-based techniques, to light-weight software-based ones (e.g., control-flow integrity) [15]. Previous works leverage cryptographic primitives or authentication mechanisms [16–18], but, typically, they are computational demanding and rely on the robustness and validity of cryptographic keys.

Also in the field of human-to-human communications, the problem of fake identities and identity theft are issues whose relevance is increasing, especially in the social network domain [19]. Researchers often refer to this branch of security as trust and reputation management, essential for establishing an efficient collaboration among a network of participants that might not have sufficient prior knowledge about each other [20].

Due to these intrinsic differences of IoT entities, an effective method to provide objects and humans with the possibility to continuously and unobtrusively authenticate each others in the network is a demanding task and, to the best of our knowledge, a unique approach for all the entities to assess the reliability of an object or a person owning an object has not been found yet.

To tackle the above issues, we designed a complete framework, called H2O (Human To Object). The environment of our system is a classical IoT network, where persons interact with objects (with low or high computational capabilities) to exchange information or get services. The IoT network can be divided in some subgroups, namely the *PAN* (the personal area network of a user, composed of all the objects owned by the user himself) and the *PU* (the set of smart public things). Moreover, the objects can be divided also according to their computational capabilities.

All the objects of the network have a profile describing what they usually do (how they interact with the environment) and some patterns of communication (both with other objects and humans). The recent scientific literature in the context of IoT contains several

approaches to computing behavioral fingerprints [8,21–23]. Fingerprint is used to analyze the current behavior of an object and to assess whether it is congruent with the expected one. Moreover, our approach is collaborative, hence if a node wants to create a new relationship with another one for which it has no fingerprint, it can rely on the knowledge of its neighbors. In particular, it can query its peers to receive information about the new node. According to *(i)* the number of answers received from them, *(ii)* the expressed opinions, and *(iii)* the relationship the querying node has with each of them, a reliability assessment towards the unknown node can be performed. Following this evaluation, the node can decide whether to interact or not with the new node.

In our approach, also human-to-human interactions can be certified with a certain degree of reliability. This step is performed through continuous authentication techniques an object can provide to its owner. Therefore, when a human wants to interact with a person he has never met before, at the start of the communication, he will need some evidence of his identity. This proof can be produced by the objects held by the second person. Moreover, the objects belonging to a *PAN* can cooperate to provide a score for their owner in case of need. Indeed, if an object is not equipped with a way to authenticate its owner through a biometric mechanism, it can rely on the members of the *PAN* network it belongs to. Of course, the reliability of the object, providing the information about its owner identity, can be assessed through the aforementioned object-to-object solution.

Therefore, in our approach, the intrinsic heterogeneity of a IoT network turns out to be a strength rather than a limitation, and it can be exploited to find a homogeneous solution to certify the real identity of a network entity.

In summary, we can list the main contributions and the novelty of our approach as follows:

- we provide a unified network, called H2O, in which each entity, either an object or a human, can estimate the reliability of its/his contacts.
- we provide an approach leveraging state-of-the-art behavioral fingerprint techniques allowing an object to assess if another one (which it usually interacts with) has been hacked or is corrupted, before starting a communication.
- we provide a consensus-based collaborative approach to allow an object to assess if another object (never met before) is reliable.
- we provide a strategy based on a combination of object-to-object fingerprinting and human-to-object biometrics techniques to allow a human to assess if another human (never met before) is who he claims to be.

This paper is organized as follows. Section 2 presents the works related to our approach. Section 3 describes the general model underlying H2O framework. Section 4 provides the detail of the proposed strategy for computing objects and humans reliability through the behavioral fingerprint of objects. Section 5 sketches the experimental campaign carried out to evaluate the performance of our approach, whereas Section 6 is devoted to illustrate the security model conceived for our framework, along with the analysis of the corresponding security properties. Finally, Section 7 concludes the work and outlines future directions.

2. Related Works

This section is devoted to the analysis of the scientific literature related to our H2O framework. Many works have been proposed in the context of reliable human-to-human, human-to-object and object-to-object interactions. Hence, for the sake of clarity, we divide this section in four parts. The former is related to human-to-object approaches devoted to the assessment of people reliability by leveraging object capabilities. In this context, research on continuous authentication based on biometrics is analysed in details. The second subsection, instead, deals with reliability in network of objects and surveys the scientific literature about remote attestation, device fingerprinting, and approaches relying on the Blockchain technology. The third part of this section focuses on the approaches tackling the problem of assessing the reliability of a person in a network of humans. In particular,

we analyze proposals in the context of reliability and trust in Social Networks. Finally, we list in Table 1 the capabilities of the related approaches, in terms of the relationships considered to define a reliability measure, and compare them with our strategy.

2.1. Human-to-Object Approaches

Although continuous authentication is a relatively new type of verification, it is gaining attention of researchers and companies seeking new forms to protect sensitive data for unauthorized access [2]. Basically, continuous authentication performs an ongoing monitoring of user interactions with objects and builds a behavior or biometrics profile leveraging Machine Learning (ML) technology to re-verify the legitimacy of the connected nodes and assure cybersecurity protection.

The application of this new form of authentication is advantageous also when traditional forms of verification do not fit the appropriate level of security. Indeed, *(i)* single-factor authentication, which provides protection at login, and *(ii)* two-factor authentication, which adds a second device for security check at the login phase, do not offer continuous validation of a user's identification.

With the aim to balance security and usability, biometric approaches are gaining momentum. Given the nature of IoT devices and their closeness to users, using them for continuous authentication purposes is specially attractive [24]. For example, smart mobile devices like smartphones and tablets are equipped with various built-in sensors like camera for iris recognition, heart-bit sensor, fingerprint scanner and microphone for voice recognition among others. But if on one hand, they are more secure, since biometric traits like iris or face are difficult to be reproduced, on another this type of authentication may provide also false positives and true negatives [25].

The authors of [26] identify two categories of biometric identification: physiological identification and behavioral one. The former includes facial, voice and fingerprint recognition, which are mostly device dependent mechanisms and require costly processing units. The latter is a form of continuous identification which is based on behavioral traits and is less intrusive, because it relies on human habitual patterns like typing [27,28], walking [29], social interactions and communication.

In [8] the authors describe WifiU, a Wi-Fi based human authentication system which recognizes users extracting unique biometrics information from Wi-Fi signals and uses it to perform human authentication. In particular, it is based on user gait. Indeed, the authors start from the consideration that, Wi-Fi signal reflected by the human body generates unique, although small, variations in wireless channel metrics on the receiver, due to the well-known multipath effect of wireless signals.

A recent trend analysed by a number of researchers deals with continuous identification of users on mobile devices within the social IoT paradigm [13,14]. In particular, in [14] the authors propose a scheme based on online behaviometrics of mobile users collected via smartphones. This scheme is able to extract features from smartphone sensors and users' social network interactions. In [13], instead, the authors propose a mobile behaviometric framework that assesses users' social activity, and introduce sociability metrics to generate signatures of users' activities.

The papers cited above aim at continuously authenticating the user by exploiting his interactions with IoT devices or against third parties. With respect to these approaches, our proposals has a more extended objective. Indeed, with the aim of enabling reliable human-to-human interactions, our framework leverages some of the strategies described in this section to obtain continuous authentication of humans with their objects. Moreover, we develop an advanced mechanism to constantly verify whether the identity of an object or a human is unchanged.

2.2. Object-to-Object Approaches

Typically, we refer to remote attestation as the security service through which an object can ascertain the current state of a potentially compromised remote device before

interacting with it. The aim of remote attestation is to allow a remote system (i.e., challenger) to check the level of trust of another system (i.e., attestator). In [30], the authors present a Multiple-Tier Remote Attestation protocol, called MTRA, verifying program integrity in IoT devices. In particular, more powerful devices are monitored by means of trusted hardware through a Trusted Platform Module (TPM), while less capable ones are verified leveraging a software-based attestation. Still in this context, the paper presented in [31] describes a many-to-one attestation scheme for device swarms, which reduces the possibility of single point of failure verifier typical of architectures in which a single node (i.e., the verifier) has to attest the reliability of multiple IoT devices.

Other works focusing on object-to-object interaction study a way to address the challenge of IoT device identification [8,21–23,32,33]. These works leverage the concept of device fingerprinting, that is a way to identify an object not relying on its classical network identities, such as IP or MAC address, but exploiting the information from the packets which the device exchanges over the network. In particular, the work presented in [21] tackles this issue by analyzing a sequence of packets from high-level network traffic and extracting from it a set of unique flow-based features to create a fingerprint for each device through machine learning techniques. The authors of [23] have the same goals, but they base their proposal on deep learning techniques. Whereas, the authors of [22] present an approach called IoT Sentinel able to automatically identifying vulnerable devices being connected to an IoT network and enforcing mitigation measures for them, so as to minimize damage resulting from their compromise. Also the proposal presented in [32] provides an IoT device identification method that models the behavior of the network packets communicated by the devices.

In the context of object-to-object interaction, a related concept is the definition of reliability and trust among things. However, due to the highly dynamic nature of the network and the large number of entities with heterogeneous computation abilities involved, it is difficult to directly apply to IoT classical approaches thought for sensor or P2P networks [16,34–36].

In particular, works leveraging cryptographic primitives [16,17,36] are computational demanding and they are not secure against internal malicious nodes having the valid cryptographic keys. On the other hand, nodes can be hacked or compromised, but they can also have hardware faults, and relying only on cryptographic mechanisms does not lead to the exclusion of these nodes from the network.

In [35], the authors describe an approach based on cryptographic primitives in which each entity has a unique and trustworthy identity. In addition, a trusted device evaluates the behavior and the performance of the nodes comparing a saved trust metric and the indirect information from a third node. In [37], the authors present IoTrust a trust architecture with a middleware layer performing authorization. The main drawback of these two schemes is that they are based on an external and reliable level that computes node reputation score. Furthermore, they defend only against some kind of attacks, such as: modification, replay, and message dropping attacks.

The proposals of [38–40] are based on Blockchain technology to provide forms of trust or authentication in a IoT network. In particular, in [38] the authors describe an Obligation Chain containing obligations generated by a number of nodes, called Service Consumers, which are first locally accepted by Service Providers and, then, shared to the rest of the network. This kind of framework is based on Islands of Trust, portion of the network where the trust is regulated by a full local PKI and Certification Authority. Also the approach in [40] relies on the security advantages provided by Blockchains, creating secure virtual zones (bubbles) where things can identify and trust each other. Moreover, although Blockchain technology provides decentralized security and privacy, it involves significant energy, delay, and computational overhead, not suitable for most resource-constrained IoT devices.

A new perspective of Internet of Things is provided by [41], that introduces the Social Internet of Things (hereafter, SIoT). This paradigm redefines the relationships among

objects putting into evidence objects autonomy. Things can navigate through the network to find resources and services of their interest, provided by other things, without human intervention. The formalization of methods and technologies allowing an object to crawl the network for finding other (possibly heterogeneous) objects and the analysis of the new social graphs thus obtained are two aspects analyzed in [42]. A step forward is done by [43], in which the authors investigate the trustworthiness management of a SIot starting from the concepts coming from P2P and social networks. They combine two models, such as a subjective and an objective one. In the former model, each node computes the trustworthiness of its friends on the basis of its own experience and on the opinion of the friends in common with the potential service providers. In the latter model, the information about each node is distributed and stored making use of a distributed hash table structure, so that any node in the network can make use of the same information.

Finally, the approach proposed in [44] deals with object reliability in a Multiple Internet of Things, defining also a profile for an object. Differently from our approach, the principle underlying the reliability of an instance in a MIoT is that it is directly proportional to: *(i)* the fraction of successful transactions performed by the instances, and *(ii)* the reliability of the corresponding objects.

It is worth observing that, our proposal exploits some of the concepts and strategies proposed by the approaches presented in this section. However, while these related solutions leverage direct interactions to compute objects reliability, our H2O scheme enhances machine-learning based fingerprinting with a consensus-based strategy to enable a wisdom-of-crowd attestation of objects behavior. In our paper, we define an object profile made up by different properties. These properties are object features related to how it usually interacts with other entities, the environment or its owner. A profile can be used as input of a machine learning solution, which computes a behavioral fingerprint for an object. This is the former step of our scheme, whose goal is to provide a complete framework equipping all entities involved (both objects and humans) with a mechanism to compute the reliability of their peers.

2.3. Human-to-Human Approaches

The problem of assessing reliability on a network of humans is a crucial task both in real and virtual world [45,46]. Typical solutions for identity deception attacks rely on legitimate community members and administrators, who are called to manually identify malicious accounts or persons [47]. The approach presented in [19] aims at computing a level of trust for each node of a network of humans on the basis of neighborhood recognition and behavioral biometric support. It describes keystroke dynamics as solution for continuous authentication for enhancing trust in social networks, in particular, biometric data are exploited as a feedback to a trust model to measure the trustworthiness of an online profile.

Trust in social networks is an extremely discussed topic, and often the boundaries between the real and virtual world are blurred in such a way that some approaches useful to compute trust between two persons in the real world can benefit measures and techniques coming from the virtual world, and vice-versa [48]. Indeed, authors of [49] state that the trust results in different communication behaviors among persons, this means that trusted communications are statistically different from random ones, and detecting trust-like behaviors allows researchers to develop a quantitative measure of who trusts whom in the network. In this context, the authors of [50] describe a model of a trust-based recommendation system which has the goal of filtering information for agents based on the agents' social network and trust relationships, thus providing recommendations for real entities through virtual ones.

Differently from these approaches, our work provides human-to-human reliable communications, by ensuring a degree of reliability. Therefore, when a human wants to interact with a peer never met before, he will need some evidence of his identity, at the start of the communication. This proof can be produced by the combination of the assessment

of an object owned by the second person and the continuous authentication technique an object can provide its owner with.

2.4. Comparison with Related Approaches

In this subsection, we summarize in Table 1 the comparison with all the works introduced in the previous sections. Specifically, we evaluate and compare all the approaches on the basis of the typology of relationship that they consider, namely:

- H-O: the cited article deals with an approach to compute the reliability of a person through an object or in the communication with an object;
- O-O: the considered paper proposes a method to assess the reliability of an object towards another object or in a network of objects.
- H-H: the cited approach proposes a scheme to assess the reliability of human-to-human communication;

The symbol 'x' denotes that the cited paper provides the corresponding property.

Table 1. Comparison of our approach with related ones.

Approach	Approach Type	H-O	O-O	H-H
Our approach	Fingerprint, Biometrics, Consensus	x	x	x
[8,27–29]	Biometrics	x	-	-
[13,14]	Behaviometrics	x	-	-
[30,31]	Remote Attestation	-	x	-
[21–23,32,44]	Device Fingerprint	-	x	-
[16,17,35–37]	Cryptographic	-	x	-
[38–40]	Blockchain	-	x	-
[43]	Social Network	-	x	-
[19]	Social Network and Biometrics	-	-	x
[50]	Social Network and Agent	-	-	x

3. General Model

In this section, we describe the general model adopted by our approach and the main actors involved in. Table 2 reports the abbreviations used throughout this paper.

Table 2. Notation used throughout this paper.

Parameter	Meaning
N	set of nodes of the network
E	set of edges of the network
n_i	ith node of the network
$\Gamma(n_i)$	the neighbors of n_i
N_{PAN}	subset of nodes belonging to a Personal Area Network (PAN)
N_{PU}	subset of public nodes of the network
N_{PPAN}	subset of nodes belonging to a PAN with high computational capabilities
N_{LPAN}	subset of nodes belonging to a PAN with low computational capabilities
N_{PPU}	subset of public nodes with high computational capabilities
N_{LPU}	subset of public nodes with low computational capabilities
P_{n_i,n_j}	profile of object n_i respect to object n_j
S	set of features
r	redundancy parameter

In the classical definition of the Internet of Things (hereafter, IoT) the network is represented as a graph $G = \langle N, E \rangle$, where N is a set of nodes representing objects and E is a set of edges representing relationships between pairs of objects. An edge is built if two objects got in touch somewhere in the past and exchanged some messages. Usually a directed graph is considered, so that an edge direction identifies the destination of the

communication. We can define the set of neighbors of a node n_i as the set $\Gamma(n_i) = \{n_j \in G : (n_i, n_j) \in E\}$.

In our model, N is partitioned into two subsets:

- The set of objects belonging to a person (denoted by N_{PAN}). All the objects belonging to this group maintain permanent physical contact with the same user during usage. This set is composed of various personal devices, that can form groups with each other when they are equipped with the short-range communication and sensing modules. Some examples are, for instance, mobile phones, PDAs and wearable devices, such as: human activity trackers, ECG readers, smartwatches, and semi-permanent insulin pumps.
- The set of nodes of public use in the environment (denoted by N_{PU}). This kind of objects are not related to humans, but can be accessed by anyone in the environment (e.g., printers or video surveillance cameras, smart multimedia object, and so forth [51]).

Moreover, a further classification, based on the processing and memory capabilities of an object, is possible. Therefore, we can divide the network in the following two groups:

- The set of nodes with high processing and memory capabilities. If these objects are of public use in the environment, they will be denoted as N_{PPU}, whereas, if they belong to a PAN, they will be denoted as N_{PPAN}. These nodes can process machine learning models and/or train classifiers for various tasks. Also smart devices that can leverage Cloud solutions to handle complex algorithms [52] belong to this category.
- The set of nodes with low capabilities. If these objects are of public use in the environment, they will be denoted as N_{LPU}, whereas, if they belong to a PAN, they will be denoted as N_{LPAN}. These devices have low computational resources and are not suitable to work with machine learning solutions. Some examples are intelligent thermostats, remotely controllable household equipment, and weather-based automated lawn irrigation systems.

Figure 1 shows a graphical representation of our scenario with the above subgroups involved. In particular, two users, namely Hope and Chad and their PANs, are depicted.

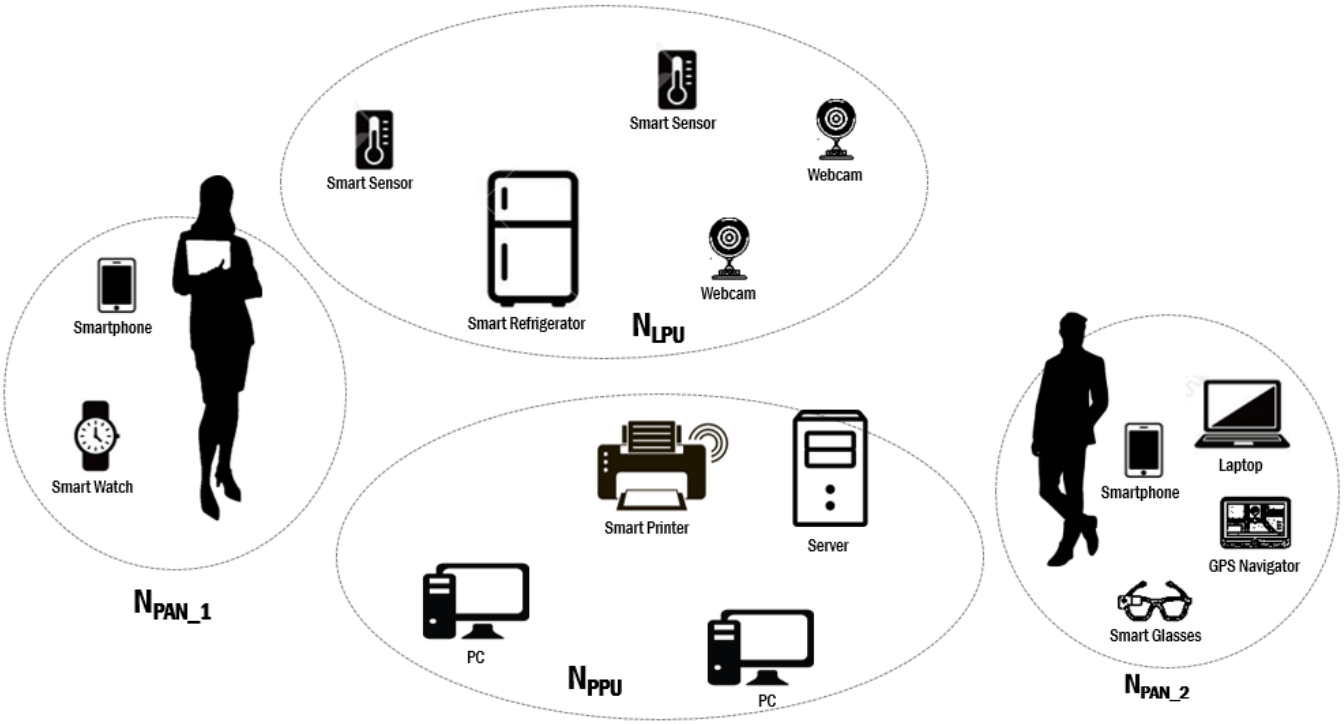

Figure 1. A representation different nodes' groups in H2O Network.

Smart objects, even belonging to the different groups, can communicate with each other. Transactions can be performed to share data or to require measures/services offered by target smart objects. Figure 2 shows a graphical representation of possible communications in our scenario deployed in a smart office.

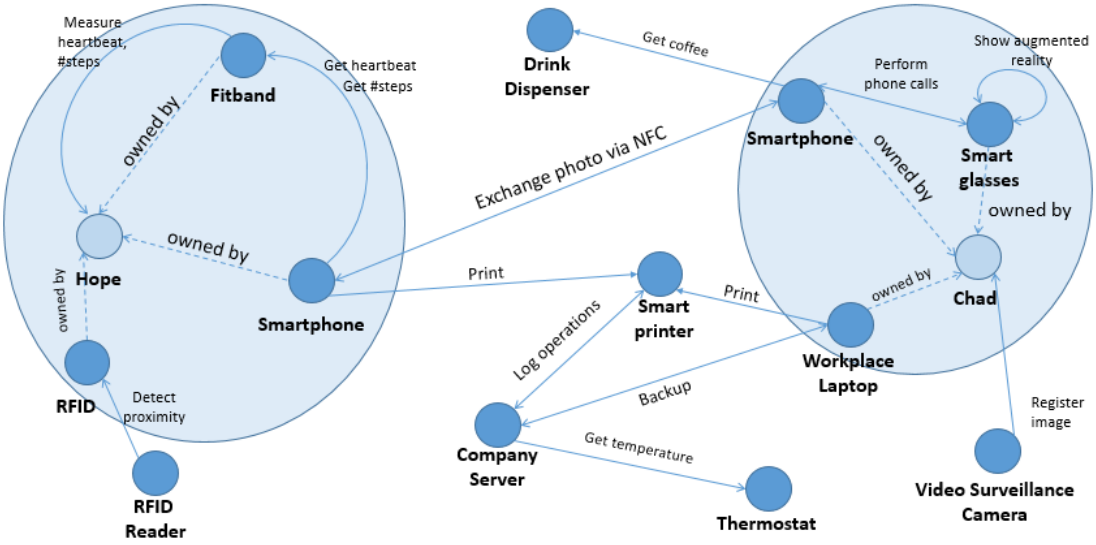

Figure 2. A representation of H2O Network with entity operations.

The main idea underlying our approach is that an object in the H2O network, when getting in touch with another object, can compare its behavior by exploiting a behavioral model [53]. Through this comparison, it can state if the object it wants to communicate with is reliable or not.

From the IoT scientific literature, we know that objects can have three different levels of sensing capabilities, namely: user awareness, ambient awareness, and social awareness [54]. In our context, the definition of such capabilities can be declined as follows:

- *Owner awareness* refers to the smart object ability to understand personal contexts and behavioral patterns (e.g., human mobility, human activity, preferences, etc.) referred to its owner. Observe that, for the N_{PPU} and N_{LPU} groups, we can use the term *Human awareness* instead of *Owner awareness*, because these objects can be used by more than one person in the network.
- *Environment awareness* concerns the capability of smart objects to derive information of a particular environment through their equipment (e.g., temperature or humidity if the object is a sensor, etc.).
- *Social awareness* represents the ability of smart objects to participate and exchange information with communities.

All these sensing capabilities can be translated to measurements and properties. Starting from them, an object profile, including a behavioral fingerprint, can be built. A fingerprint is a complex model considering different dynamics related to the object activities, such as: *(i)* how the object interacts with the other objects in the network, *(ii)* how the owner of the object interacts with it and *(iii)* the value of the metrics about the environment that the object can compute.

As stated in the Introduction, our approach leverages behavioral fingerprinting as a mean to build a mechanism to improve communication reliability in an IoT. Therefore, the first step concerns the construction of object profiles including such behavioral fingerprints. Due to the fully distributed nature of our application context, these models are built by objects with high computational capabilities, on the basis of interaction data among pair of nodes. To do so, we define a *safe starting phase*, in which all the interactions are considered safe and no malicious node is involved in the environment. During this phase, each node

is in charge of *(i)* acquiring interaction data with its peers and *(ii)* proceeding with the computation of the object profile and the behavioral fingerprint. In general, only nodes belonging to N_{PPU} or N_{PPAN} can build complex models; however, delegation strategies can be adopted to extend the approach also to nodes belonging to N_{LPPU} or N_{LPAN}. The details about the behavioral fingerprinting for H2O objects are described in Section 4.

After the *safe starting phase*, the network moves to a fully operational mode. During this second phase, network participants can leverage behavioral models trained in the previous phase. Each model refers to the relationship between the participant itself and one of its neighbors. Through these models, a node can continue to check if the actions performed by its neighbors are still compliant with the behavior observed during the previous phase. In the negative case, a peer can be assumed to be damaged or attacked and, therefore, no more reliable.

The H2O network can grow over time. This means that new nodes can be added and new edges can be created. If new nodes are added to the network after the *safe starting phase*, no information about their behavior recorded during a safe period (with total absence of attacks) is available. In this case, unless a new training phase is carried out, the reliability level for new nodes has to be set to a default value. Whereas, to support the possible evolution of the H2O network in terms of new interconnections between nodes, our approach relies on a collaborative mechanism. In particular, given an object, say n_i, which wants to communicate with a peer, namely n_j, if no interconnection between these two nodes has occurred during the *safe starting phase*, n_i can ask the set of its neighbors opinions about the reliability of n_j. If some past interactions between n_j and the latter set occurred, n_i will receive a number of responses from its neighbors. At this point, a reliability score about n_j can be computed by n_i, by averaging the obtained results. The details about how this reliability score is computed are described in Section 4.2.

In our scenario, also humans participate in the network by means of their personal devices (the objects belonging to N_{PAN}). Interestingly, by leveraging reliability scores derived from object interactions also human-to-human inter-communications can be made more secure (see Section 4.4). Indeed, let us suppose that a person, namely h_i, wants to communicate with another person, say h_j. If h_i has never met h_j in the past, sending private documents through its smartphone via NFC to h_j could result in security issues. To make this step more robust, our approach provides a mechanism which works as follows. First off, we refer to the sender's smartphone as n_i and to the receiver's smartphone as n_j. Before starting the file exchange, n_i can request to n_j the evidence that h_i is who he claims to be. Continuous authentication and biometrics solutions are acquiring a lot of attention from the scientific community (see Sections 2 and 4.3 for details about it). These strategies ensure the possibility of providing human identity confirmation and protection on an ongoing basis. Many approaches deal with the implementation of these security mechanisms through personal smart objects (such as smartphones, smartwatches, etc.) [55]. By leveraging such solutions, the object n_j can prove its owner identity, through biometrics mechanism. All the details about object-to-human interactions are described in Section 4.3. Observe that, if n_j is a low power device it can rely on a more powerful object belonging to its PAN to perform this task. Of course, before relying on n_j's answer, n_i has to check the reliability of n_j in the H2O network. It can do so, by performing the steps of the object-to-object interaction explained above and detailed in Section 4.2. Figure 3 summarizes the steps of our approach through an example of a human-to-human interaction in full operational mode. Observe that, this includes also the steps for the corresponding object-to-object interaction.

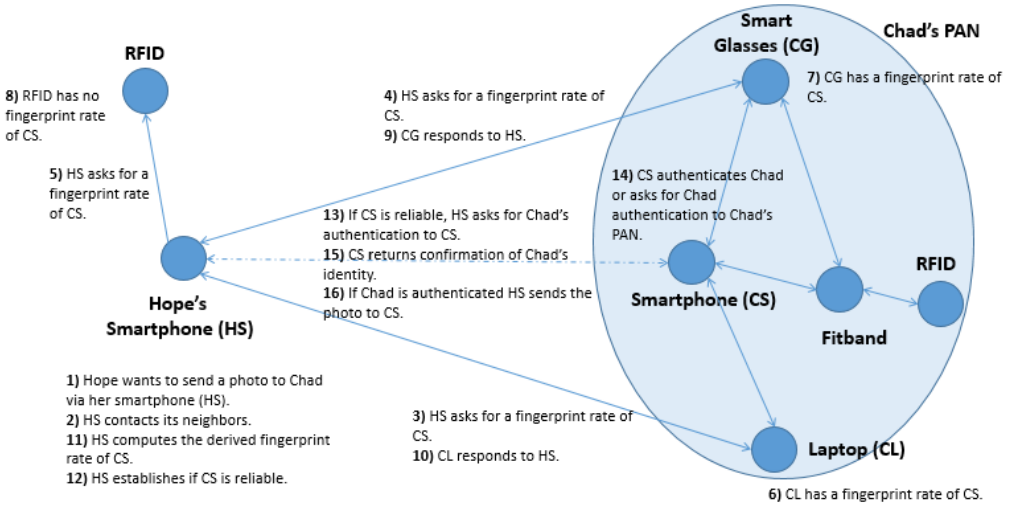

Figure 3. The steps of our approach for a human-to-human interaction.

4. Behavioral Fingerprint for Reliability

In this section, we present the main concepts underlying our approach and we detail all the steps of the main phases of our scheme. In particular, in Section 4.1, we describe how the behavioral fingerprint is computed for an object in the H2O network. This behavioral fingerprint will be used during all the object-to-object interactions of our H2O network, whose steps are detailed in Section 4.2. After that, in Section 4.3, we survey on some biometrics algorithms our scheme can rely on in order to assess persons authenticity. Finally, Section 4.4 is devoted to the description of the way our approach can make communications reliable, also in the context of human-to-human interactions.

4.1. Fingerprint Computation

In this section we deep dive in the description of the behavioral fingerprint computation. Behavioral fingerprint is a very timely research subjects and several investigations are already available in the scientific literature [8,21–23]. Most of the existing works deal with the problem of identifying general features that can be applied to any existing IoT device, but, such that, together can unequivocally represent a single specific object. Generally, two categories of fingerprinting strategy are adopted, namely: *(i)* non-behavioral fingerprinting and *(ii)* behavioral fingerprinting [53].

The first category of approaches leverages features related to physical and link layer characteristics [56,57]. For instance, the authors of [56] extracted 19 features from the 802.11 probe fields. The selected features have been identified by observing which variations are typically recorded when the devices are under attack or involved in anomalous situations in their environment. In particular, this approach works as follows. In the first phase, wireless network traffic from devices is collected, and probe request frames are filtered out to extract the data-link layer header from the frames. After removing the outliers, a set of unique, tamper-proof, non-reproducible features that can be used for the device type fingerprinting are chosen. Signature has been used to generate the profile for each device type. Finally, these profiles are then matched in the verification phase through some similarity measures to establish if the device type is registered in the system.

In the approach described in [57], instead, the authors focused on a different set of features related to clock characteristics and TCP timestamp. In particular, this approach is based on thirteen clock characteristics to remotely identify different IoT device models of various manufacturers. The authors define a set of machine learning features related to how monotonic timestamp clocks generate TCP timestamp values, and they use these features to detect model specific characteristics. Moreover, they compare different machine learning algorithms and identified Random Forest as the best classifier in this setting. In general,

non-behavioral fingerprinting can be successfully applied if the considered features can be adapted to the characteristic of the network, in which the object is located. In this way, particular situations in the network, like congestion or bottlenecks, do not impact on the fingerprinting model.

Behavioral fingerprinting, instead, focuses on more application-level features to model objects' traits. Therefore, instead of focusing only on the physical and link layers information, the approaches belonging to this category go further in the characteristic of the packets generated by extracting information. Among these characteristics there are: protocols, request-response sequences and any periodicity in specific typology of packets along with their sizes [58–60]. In particular, the approach of [58] focuses on TCP/IP connections and extracts features from them (such as, time-to-live, byte sent and received, etc.). With these features, the authors trained a Random Forest model and were able to distinguish the monitored objects from external ones with a precision of 97%.

Similarly, the approach described in [59] focuses on different typologies of traffic an object can generate, for instance queries to DNSs, packets related to user activity on known application (e.g., google home, maps, etc.), and interaction with known service. With these information, the authors adopted a deep learning solution based on LSTM-CNN model. This approach reaches a best accuracy value of 80.1%.

Very close to our application scenario is the solution described in [60]. In this paper, the authors describe a distributed solution for behavioral IoT fingerprinting. Indeed, because models for fingerprint must be kept updated in accordance to possible variation in the devices' configuration, a centralized solution would suffer from scalability issues. For this reason, they identify some nodes, i.e. the gateways, inside the IoT that can monitor objects using trained classification models, thus providing scalability to the solution. The training part, instead, is performed by special controller nodes, which in this scenario are considered as part of the ISP. Inside such an architecture, the authors identify a feature vector with 111 dimensions. They tested the performance of their approach obtaining an accuracy of 97% when classifying known devices.

With the evolution of the IoT technology, nowadays paradigms such as Internet of Multimedia Things (hereafter, IoMT) have been introduced. In such settings, the devices are assumed to have higher computational capabilities than classical sensors belonging to the Internet of Things. With this in mind, we argue that, in modern general IoT scenarios, we can consider that a combination of Multimedia Things and simple classical smart objects coexists. Therefore, in our H2O network, we consider that the role of controller nodes can be played by the top powerful devices, belonging to N_{PPU}, which are in charge of training the models by leveraging edge/cloud computing solutions [61]. Whereas, the monitoring nodes can be more common objects with sufficient computational capabilities to execute a trained model. Finally, low-power objects, belonging to N_{LPU} or N_{LPAN} can leverage delegation strategies to gain reliability information about other nodes. All the controller devices use the same algorithm to build object fingerprints and the same set of features, say S. During the *safe starting phase*, objects with monitoring capabilities interact with their peers to acquire the training set to build the corresponding models. The training set is, hence, used by the controller nodes to fit the model. With that said, in our scenario each monitoring node, say n_i, associates and maintains a *profile* with each of its contacts, say n_j which were added before the *safe starting phase*. In particular, a profile can be defined as follows:

$$P_{n_i,n_j} = \langle Net_{n_j}, F_{n_i,n_j} \rangle \tag{1}$$

Here, Net_{n_j} is a set containing information related to connectivity aspects (such as, the IP address, the MAC address, and so forth), and F_{n_i,n_j} represents the fingerprint of n_j maintained by n_i. In particular, the fingerprint that n_i associate with n_j can be defined as:

$$F_{n_i,n_j} = \langle m_{n_i,n_j}, S \rangle \tag{2}$$

where, m_{n_i,n_j} represents the model built during the *safe starting phase* according to the data exchanged between n_i and n_j. Whereas, the set S represents the set of features considered in the model. After the *safe starting phase* F_{n_i,n_j} is used by the monitoring node n_i to assess whether the behavior of n_j is unchanged. This information can be, then, used by the network members to estimate a reliability score for the others. In the next section, we will describe this aspect in more detail.

4.2. Object-to-Object Interaction

In this section, we describe how the reliability of an object is assessed by leveraging object-to-object interactions. In order to detail all the steps of this part of our approach, we preliminary define a redundancy parameter r, that is a positive integer representing a minimum consensus level to suitably estimate the reliability of an object. Moreover, given the set of neighbors of n_i, namely $\Gamma(n_i)$, we denote by $R(n_i) \subseteq \Gamma(n_i)$ the set of nodes which have been tested by n_i and whose identity is monitored (either directly or through delegation strategies) by comparing the current interactions with their behavioral fingerprints.

As stated before, we assume that there exists a *safe starting phase* in which neither fraudulent access nor physical damage to legitimate nodes can be performed. During this phase the machine learning algorithms, described in Section 4, can be trained to produce the fingerprints for each objects. In a subsequent non-safe phase, objects can leverage such models referring to other nodes which they usually communicate with. Periodically, a node n_i can test one of its neighbors (whose behavior has been modeled during the *safe starting phase*), to assess whether it has been corrupted or hacked. Let us denote with $n_j \in \Gamma(n_i)$ the node to be tested. To do this, n_i exploits the behavioral fingerprinting model providing data extracted from a set of transactions done with n_j, as input. The output of the model is a normalized fingerprint rate, represented as $fr(n_i, n_j)$, whose values range in $(0, 1)$. In order to assume n_j reliable, the fingerprint rate must be greater than a give assurance threshold $0 < 1 - \epsilon \leq 1$; in other words, the fingerprint rate should be such that $1 - \epsilon \leq fr(n_i, n_j) \leq 1$ to consider n_j still reliable.

After this check, n_i stores $fr(n_i, n_j)$ in a internal table for future transactions.

Now, let us suppose that the node n_j wants to interact with n_z, but it has not met n_z during the *safe starting phase*. In this case, no information about the reliability of n_z is available, directly. To address this situation, our approach adopts a strategy based on the consensus of a suitable portion of the neighbors of n_j to compute a derived fingerprint rate.

In particular, let $\Delta(n_j, n_z) = \{n_x \mid n_x \in \Gamma(n_j) \cap \Gamma(n_z) \wedge n_z \in R(n_x)\}$ be the set of objects belonging to both the neighbors of n_j and n_z, which have computed a normalized fingerprint rate for n_z. The derived fingerprint rate for n_z can be computed by n_j as follows:

$$dfr(n_j, n_z) = \frac{\sum_{n_x \in \Delta(n_j,n_z)} fr(n_x, n_z)}{|\Delta(n_j, n_z)|} \tag{3}$$

At this point, n_z can be considered reliable by n_j if the following conditions are true: *(i)* $dfr(n_j, n_z) \geq 1 - \epsilon$ and *(ii)* $|\Delta(n_j, n_z)| \geq r$.

Therefore generalizing the above reasoning, in order to be reliable for n_j, a node n_z has to match one of the following set of conditions:

- *(i)* $n_z \in R(n_j)$ and *(ii)* $fr(n_i, n_z) \geq 1 - \epsilon$; that is, n_j holds a fingerprint model of n_z and the normalized fingerprint rate is distant from the maximum value no more than a certain threshold ϵ;
- *(i)* $|\Delta(n_j, n_z)| \geq r$ and *(ii)* $dfr(n_j, n_z) \geq 1 - \epsilon$, that is at least r nodes in $\Gamma(n_j)$ should have tested n_z and expressed a positive check (greater than or equal to $1 - \epsilon$).

We call this last property r-redundancy. It is based on the assumption that multiple confirmation of the reliability of a node can be considered sufficient to trust the behavior of that node [19]. Obviously, the higher the value of r, the higher the reliability about the object behavior.

To better understand the steps above, let us consider the example represented in Figure 4. This figure reports a portion of an H2O network containing the nodes n_j and n_z along with their neighbors. Black edges represent interactions assured by the presence of a fingerprint model. Hence, the two nodes linked by such edges have first met during the *safe starting phase*. Moreover, consider an example configuration in which $\epsilon = 0.3$ and $r = 2$.

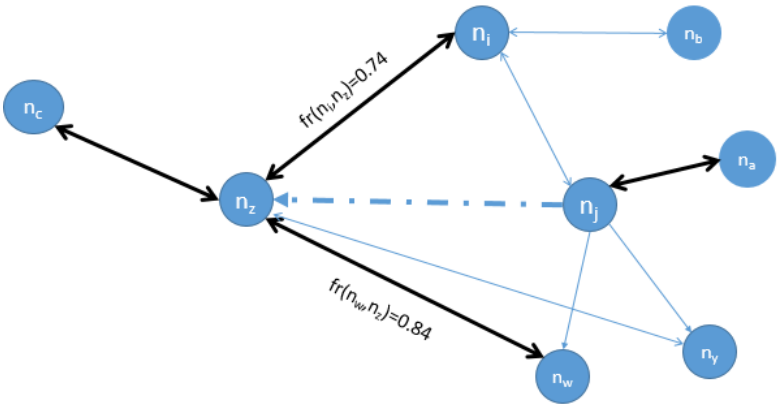

Figure 4. A network portion of H2O.

Suppose again that the node n_j wants to interact with n_z, but it has not met n_z during the *safe starting phase*. The dashed edge in Figure 4 represents the possible future interaction between n_j and n_z. Before starting the communication, n_j checks whether $n_z \in R(n_j)$. If n_j had a direct access to a fingerprint model for n_z, it could check if $fr(n_j, n_z) \geq 1 - \epsilon$. However, in this example, n_j has no direct access to a behavioral model for n_z. Therefore, n_j contacts the objects in its neighborhood ($\Gamma_{n_j} = \{n_a, n_i, n_y, n_w\}$ in the example reported in Figure 4) to obtain information about n_z. In particular, n_j looks for objects in its neighborhood, who can leverage a behavioral model of n_z to assess its reliability, which, as a consequence, belong also to the neighborhood of n_z. In the example, two nodes, namely n_i and n_w, belong to the interception set $\Gamma(n_j) \cap \Gamma(n_z) = \{n_i, n_y, n_w\}$ and have access to a behavioral fingerprint of n_z ($\Delta(n_j, n_z) = \{n_i, n_w\}$). These two nodes answer to n_j with $fr(n_i, n_z)$ and $fr(n_w, n_z)$, respectively.

Having this two values, n_j computes the formula in Equation (3), which becomes:

$$dfr(n_j, n_z) = \frac{fr(n_i, n_z) + fr(n_w, n_z)}{2} = \frac{0.74 + 0.84}{2} = 0.79$$

At this point, n_j will consider n_z reliable because $dfr(n_j, n_z) \geq 0.70$ and $r \leq 2$.

Algorithm 1 summarizes the steps of our approach for object-to-object reliability assessment.

As for the overall computational complexity of the above consensus mechanism, we can express it in terms of number of messages sent during the object-to-object reliability assessment. Therefore, leveraging asymptotic analysis, we can state that this solution guarantees a linear computational complexity, $O(m)$, in the dimension, say m, of the number of neighbors for a node. It is worth noting that, in this analysis, we do not consider the computational cost of the fingerprint rate $fr(n_i, n_j)$, that depends on the fingerprint algorithm used during the *safe start phase* (see Section 4).

4.3. Human-to-Object Interaction

This section is devoted to detail the various biometric authentication techniques our approach can rely on, in order to make an object in the H2O network, capable of continuously identifying its owner. As will be clearer in the next, this step is essential to provide a mean to assess entities reliability also in human-to-human communications. Biometric identification is defined as the mechanism to automatic identify a person through

the analysis of his biological (physiological, anatomical) or behavioral traits. This may happen since most of the biological and behavioral characteristics used are peculiar of an individual and can uniquely identify him. Indeed, in the last years, the use of ID cards, keys, passwords, or other standard systems has been replaced in lots of contexts by these more dependable forms of authentication [62]. Obviously, not all the human characteristics can be chosen to be used in biometric authentication. Therefore, according to the National Institute of Standards and Technology (NIST) (https://www.nist.gov, accessed on 30 April 2021), there are some features that make a human trait distinctive and hence, eligible for acceptance. These features are: universality, uniqueness, permanence, measurability, performance, acceptability and circumvention [63]. Examples falling in the aforementioned biological category of biometric mechanisms are: fingerprint, hand geometry, iris, face, and ear. Whereas, examples of behavioral biometrics mechanisms include: gait, signature, and keystroke dynamics.

Algorithm 1: Algorithm for object-to-object reliability computation

Result: Node reliability computation
n_j wants to communicate with n_z;
if $n_z \in R(n_j)$ **and** $fr(n_j, n_z) \geq 1 - \epsilon$ **then**
  n_z is reliable;
else
  while $n_i \in \Gamma(n_j) \cap \Gamma(n_z)$ **do**
    n_i computes $fr(n_i, n_z)$;
    n_i sends $fr(n_i, n_z)$ to n_j;
  end
  n_j computes $dfr(n_j, n_z)$;
  if $(dfr(n_j, n_z) \geq 1 - \epsilon)$ **and** $(|\Delta(n_j, n_z)| \geq r)$ **then**
    n_z is reliable;
  else
    n_z is not reliable;
  end
end

As already said, the diversity of devices and applications in the IoT universe leads to a variety of solutions to the problem of continuous authentication of users (through biometrics). Such a diversity is mostly due to the different capabilities of the devices themselves.

According to this reasoning, we can divide IoT devices that interact with humans and which compute biometrics to authenticate their owners, into two sets. The former is composed of objects having a permanent physical contact with the person during usage, such as: activity trackers, smartwatches, and insulin pumps. The latter consists of devices that do not maintain permanent contact with humans, such as: intelligent thermostats, occupancy sensors, and smart household appliances. To the first set belong wearable devices holding an inertial measurement unit (IMU), which are, in turn, comprised of an accelerometer and a gyroscope, and that can verify and authenticate their owner by his gait. The concept underneath is that humans move their limbs in unique patterns, and these patterns can be sensed be the device. Leveraging machine learning techniques to learn these patterns, software inside smartphones, fitbands, smartwatches and specific health sensors can be able to verify a human through his gait.

Similarly, photoplethysmogram (PPG) sensors generate signals according to the amount of blood that flows in a person's veins, which depends on frequency heartbeat. In particular, this last trait that can be analysed through echocardiogram signals (ECG) can be monitored by smartphones, fitbands, smartwatches and specific health sensors. It contains enough information to enable user authentication. Potentially this kind of information can be combined with the one coming from IMU to provide a multiple-biometrics

measure, more accurate and stable, that traces ECG according also to changes in speed of user' movements.

Always in the context of biological traits, specifically anatomical ones, distinctive physical human characteristic are the iris, the shape and appearance of ears, the face, the fingers and the hands' palms. These features are invariant during the growth period of children and in adult lifetime [62]. For instance, the visible colored rings around pupil that compose the iris have a unique size for each individuals. Also the shape and appearance of ears are unique and have relatively little change during the lifetime of an adult. Moreover, also fingerprints and palms of the skin of the hands have unique pattern of ridges and valleys. All this traits can be analysed by a camera or via specific sensors for fingerprint installed in a laptop or smartphone.

On the other hand, there is a number of mechanisms performing continuous authentication based on the analysis of human behaviour. All these approaches can be referred as behaviometrics, that is the analysis of a person's behavior, rather than his physical characteristics, with the aim to identify uniquely that person. One of the most popular is keystroke [64], which tries to identify the authenticity of a user when he is working via a keyboard. In particular, this process analyzes the way a user types at a terminal by monitoring the keyboard with the aim to identify users based on habitual typing rhythm patterns.

Always in the context of behavioral biometrics, recent works rely on the particular way a person uses the devices he usually interacts with. For instance, in [65] the authors extracted some side-channel features from network traffic generated by smartphones. Specifically, they state that the use of the most popular smartphone applications, such as Facebook, WhatsApp, Skype and Dropbox generate a network traffic peculiar for a single individual; and it can be used to reliably identify the owner's smartphone. Also the use of Social Networks through the smartphone can be consider characteristic for an individual. Indeed, the works presented in [13,14] deal with this challenge, proposing some mobile behaviometric frameworks assessing users' social activity, and introducing sociability metrics to generate signatures of users' activities. Typically, to compute behavioral biometrics a user has to rely on powerful devices, such as smartphones or personal laptops.

In summary, the are various possibilities for IoT devices to assess, in any moment, if their owner is who he claims to be. Specifically, in H2O network, each device can perform continuous authentication of its owner according to its own capabilities. For instance, a camera can rely on face recognition, whereas a fingerprint sensor on human fingerprint. The set of possibilities for a smartphone are, surely, wider, since this kind of personal device has a number of capabilities spanning from biological human traits to behaviometrics. The choice of the particular mechanism for each device is orthogonal to our study, as long as a device can provide an answer to another object about its owner identity, once requested. Anyway, as already stated, in H2O network not all the devices are equipped with biometric sensors able to perform a continuous authentication task. If the device belongs to a N_{LPAN}, our approach leverages delegation strategies ensuring to this device the possibility to rely on a more powerful one, always belonging to the same PAN. The whole list of steps and the conditions for which a device can rely on another device's to identify the common owner, will be explained in the following section.

4.4. Human-to-Human Interaction

In this section we describe the second reliability mechanism our H2O network is equipped with. Indeed, in Section 4.2, we detailed how the interaction between two objects can be made more robust against attacks aiming at corrupting or damaging a node. Indeed, our approach provides a way for a node to assess the reliability of a second node it is interacting with. At the same time, if a node requests a service from another one for the first time, it can rely on our approach to know what its neighbors think about that node and assess whether it can be considered reliable or not.

By applying a similar reasoning, our approach can be extended also to human-to-human interactions. A sequence diagram showing our solution in this particular case is reported in Figure 5.

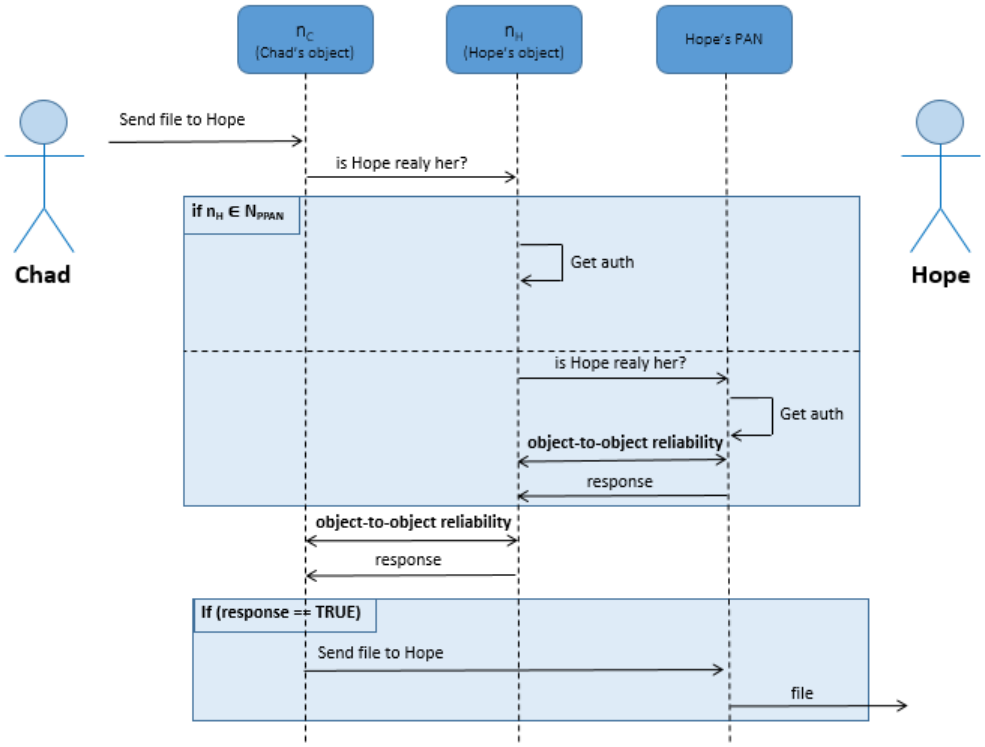

Figure 5. The sequence diagram for a human-to-human communication.

In particular, if a human wants to interact with another human in the H2O network our protocol will perform a sequence of steps useful to assess the reliability of the second human. In particular, as showed in Figure 5, let us hypothesize that Chad wants to interact with Hope to send a personal file. For this task, Chad relies on one of his object, let us say n_c, whereas Hope's object, that has to receive the file, is identified with n_h. In this scenario, two cases may happen. The first possibility is that n_h is an object with sufficient computational power (i.e., it belongs to Hope's N_{PPAN}) and it is equipped with an algorithm for the computation of biological or behavioral biometrics mechanisms (showed in Section 4.3). In this case, before n_c sends the file to n_h, it will request the result of the biometrics authentication mechanism n_h is equipped with. At this point, our scheme provides n_c with a way to assess n_h reliability, too. In order to do so, it will perform the sequence of steps described in Section 4.2 for object-to-object reliability assessment. If n_h is reliable and its response about Hope reliability is positive, the file exchange can safely take place.

The second case happens if n_h is a lower power object (i.e., it belongs to Hope's N_{LPAN}), hence, it is not equipped with a way to authenticate its owner, Hope. Our approach allows n_h to rely on more powerful objects belonging to Hope's PAN for this computation. To do so, an approach based on consensus, similar to the one described in Section 4.2, can be adopted. Specifically, let $\Delta^H(n_h)$ be the set of objects belonging to the PAN of n_h, which have the capability of computing biological or behavioral biometrics mechanisms to authenticate their owner. Let $\widehat{\Delta^H(n_h)} \subseteq \Delta^H(n_h)$ be the set of nodes of $\Delta^H(n_h)$ that can be considered reliable by n_h according to the strategy described in Section 4.2. To authenticate its owner, n_h can relay on any node belonging to $\widehat{\Delta^H(n_h)}$; moreover, n_c must adopt again the strategy described in Section 4.2 to confirm the reliability of n_h's answer. Finally, if all the involved nodes are reliable and the response about Hope's reliability is positive, the file exchange can safely take place.

Algorithm 2 summarizes the steps of our approach for human-to-human reliability assessment.

Concerning the computational complexity of this functionality, we can consider two contributions. The former is the cost of the continuous authentication mechanism, say $auth(h_i)$, that strictly depends on the specific biometrics approach adopted. For these reason, we generically refer to this cost as C_{auth} and we do not include it in our analysis.

The latter, which is the cost introduced by our approach, can be expressed, once again, in terms of the number of messages exchanged by the objects involved in this task. In particular, the human starting the communication will use one of his object for this task. This object will exploit the object-to-object mechanism, described in Section 4.2, to assess the reliability of the object owned by the target human. As stated before this task has a linear computational complexity in the dimension of the number of neighbors for a node, say m. Moreover, in the worst case, the object owned by the target human will query his whole PAN. Assuming that the maximum size of the set of objects of a PAN is s, then this further step will have a linear cost with respect to it. Overall, the cost of this strategy can be estimated as $O(m \cdot s)$.

Algorithm 2: Algorithm for human-to-human reliability computation

Result: Human reliability computation
h_i and h_j are two humans;
h_i wants to communicate to h_j;
h_i owns n_i;
h_j owns n_j;
n_i has already assessed the reliability of n_j via Algorithm 1;
if $n_j \in N_{PPAN_j}$ **then**
  n_j sends $auth(h_j)$ to n_i;
else
  $n_w \in \Gamma(n_j) \cup N_{PPAN_j}$ sends $auth(h_j)$ to n_j;
  if $n_w \in R(n_j)$ **and** $fr(n_j, n_w) \geq 1 - \epsilon$ **then**
    n_j sends $auth(h_j)$ to n_i;
  else
    while $n_t \in \Gamma(n_j)$ **do**
      n_t sends $fr(n_t, n_w)$ to n_j;
    end
    n_j computes $dfr(n_j, n_w)$;
    if $(dfr(n_j, n_w) \geq 1 - \epsilon)$ **and** $(|\Delta(n_j, n_w)| \geq r)$ **then**
      n_j sends $auth(h_j)$ to n_i;
    else
      h_j is not reliable;
    end
  end
end
if *auth(h_j) == true* **and** n_j *is reliable* **then**
  h_j is reliable;
else
  h_j is not reliable;
end

5. Evaluating the Performances of H2O Network

In this section, we describe the experimental campaign we carried out in order to test the feasibility of our proposal. This set of experiments are also useful to tune the parameters of our approach and determine the best configuration for the different application scenarios.

We leveraged on the online dataset provided by [66]. This dataset is based on real IoT objects available in the city of Santander (https://www.smartsantander.eu/, accessed on 30 April 2021). This old Spanish seaport on the north coast became one of the first smart city in the world in 2010. During that period, indeed, 12,500 sensors have been placed in and around the city's downtown district, to measure and monitor a lot of things from the number of available parking spaces to the amount of trash in containers, or crowds on the sidewalks. In addition, sensors on public vehicles were added to measure air pollution levels and traffic conditions.

The dataset objects are categorized following the typologies and data model introduced in the FIWARE Data Models (https://www.fiware.org/developers/data-models, accessed on 30 April 2021). This kind of data models has been thought to enable data portability for different applications in the IoT context, such as: Smart Cities, Smart Environment, Smart Sensoring, Smart Energy, Smart Water and so on.

The dataset collects the information of both private mobile devices, which are carried by the users during their movements, and static objects, left in a single location (e.g., the users' house). Among the former category of private mobile devices there are: smartphones, cars, tablets, smart fitness devices and smartwatches. The latter category includes: personal computers, smart printers and home sensors. Also public devices of the city of Santander are available in the dataset, such as: environment and weather monitoring objects, transportation vehicles (i.e., taxis or buses), digital indicators, garbage trucks, city street lights, parking locations and security alarms or traffic monitoring devices. Users' movement is simulated through a mobility model called Small World in Motion (SWIM) [67,68]. SWIM generates traces for ad-hoc networking having the same statistical properties of real traces. The parameters used for this simulation along with some numeric of the dataset are shown in the Table 3. The map in Figure 6 is a graphical representation of the objects' positions in the Santander city used for our experiments. For the sake of presentation, in this figure, the object position has been scaled down to a unitary square area [66].

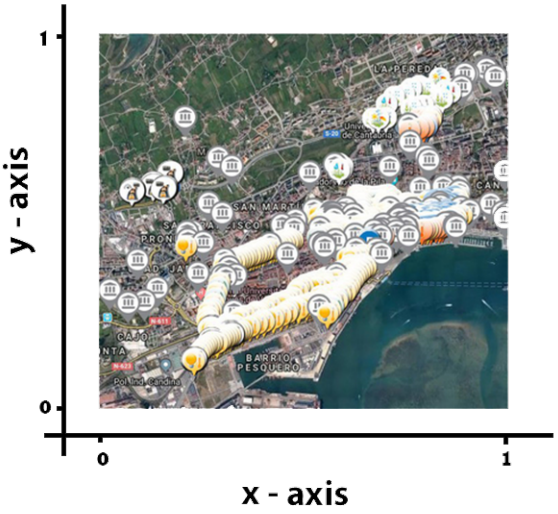

Figure 6. The maps of smart things of Santander used in our experiments.

To carry out our experimental evaluation of the performance and features of our approach, we built a simulator using Python language on an Ubuntu Server equipped with 32 GB of RAM and a 8 core CPU at 5.10 GHz.

Our first experiment is devoted to the comparison of our approach with a naive one, in terms of the load generated for the H2O, that is the number of transactions sent in the communications. According to the naive approach, each object sends data to another one in the network directly, without applying any strategy to assess if the target of its communication is reliable or not.

Table 3. Information about the dataset.

Parameter	Value
Total Number of objects	16,216
Number of private objects	14,600
Number of public objects	1616
Number of objects unique relation	550,939
Average objects degree	67.95
Number of persons	4000
Range of movements	2000 m
Simulation time	10 days
Number of transactions	1,101,878

This experiment gives us a better understanding of the impact of the reliability mechanism introduced by our approach to the traffic generated in the H2O network during the fully operational phase. Indeed, while the naive strategy generates only the number of transactions related to the communication between two objects, our approach introduces an overhead related to the assessment of objects' reliability. The difference, in term of percentage of transactions for a time period of 10 days and $r = 2$, is shown in Figure 7.

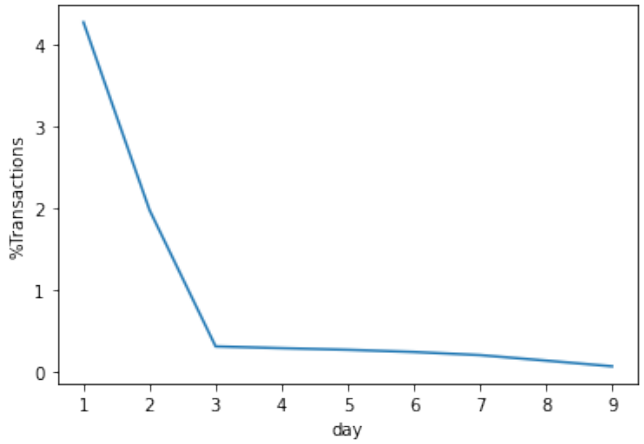

Figure 7. Different percentage of transaction between our solution and a naive approach.

Assuring reliable communications is a matter of costs and adding overhead in the network can be tolerated if we think how valuable the data and the process exchanged are. With that said, it is clear that such overhead in terms of number of transactions in the network is correlated to: *(i)* the total number of neighbors for each nodes; and *(ii)* the number of transactions created during the *safe starting phase*. For this experiment, we have set the percentage of transactions in the *safe starting phase* to 10% of the number of total transactions. For the results, we can see that the percentage of additional transactions has a pick right at the beginning of the simulation. This is due to the fact that the fingerprinting rates (and, of course, the derived fingerprint rates) are mainly computed when the first interaction between objects takes place. Therefore, during the initial period nodes start to interact with each other and the first set of interactions is used to test the reliability of their peers. Later, nodes tend to communicate with the consolidated list of contacts and can re-use the computed fingerprint rates.

To deepen the study about the tuning of the different parameters of our model, we performed a second experiment in which we focus on the percentage of nodes for which our approach can provide an estimation of the fingerprint rates and, therefore, of their reliability. Of course, such percentage is strictly related to the distribution of the number of contacts for the nodes in the network. Therefore, for starters, we report in Figure 8 the degree distribution of the analyzed H2O network.

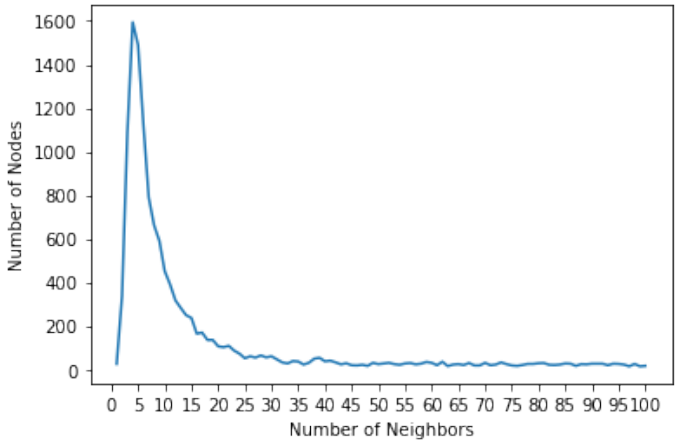

Figure 8. Nodes degree distribution in H2O network.

Observe that the distribution follows a power law which is a typical distribution of social systems [69]. According to this distribution, we can observe that most of the nodes have a number of neighbors around 6, whereas a very small percentage of nodes have up to 100 contacts.

With that said, we focus on the network after the *safe starting phase*. During this phase some nodes can be hacked or corrupted, moreover, new transactions between nodes not holding the fingerprint model can be established. In this experiment, we considered that 10% of the nodes was able to create fingerprint models with their acquaintances during the *safe starting phase*. In this case, the r-redundancy parameter plays a fundamental role, indeed, it imposes that at least r neighbors of a node n_i must own fingerprint rates of a target node n_j to allow n_i to compute a derived fingerprint rate for n_j and, hence, to assess whether it can be considered reliable or not. In Figure 9, the percentages of assessable nodes over the total number of node in the H2O network, for different values of r parameter, are shown. In this experiment, the r-redundancy threshold assumes integer values in the interval $[2, 20]$. Observe that, when we talk about assessable nodes, we refer to nodes for which either a direct or a derived fingerprint rate can be computed. This score can be, hence, used to establish the reliability of nodes (i.e., a node will be considered reliable if this rate is greater than the threshold $1 - \epsilon$, see Section 4.2).

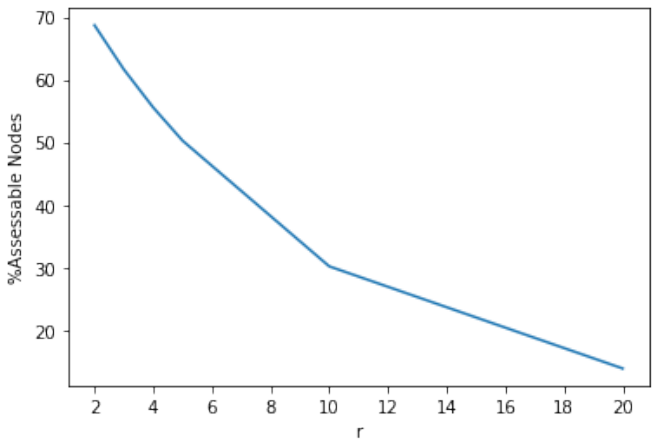

Figure 9. The percentage of assessable nodes for different values of r.

This result is useful to choose the appropriate value of the redundancy parameter r basing on the percentage of not reliable nodes tolerated in the H2O. From this figure, we can see that by setting $r = 5$, which is a value in line with the average number of neighbors for a node in the network, with just the 10% of nodes owning a direct fingerprint model

with others, our approach can estimate the reliability of more than the 50% of nodes in the H2O network.

The last experiment focuses on the capability of our approach to inhibit possibly malicious transactions originating from hacked/corrupted nodes in H2O. For this experiment, we considered the following different percentages of hacked/corrupted nodes: $[1\%, 2\%, 4\%, 10\%, 20\%, 50\%]$. Whereas, the r parameter assumes the following values: $r = [2, 5, 10]$.

We measured how many times a transaction towards an hacked/corrupted node has been prevented by means of our solution.

Figure 10 shows the absolute number of prevented attacks against the number of hacked/corrupted nodes and with different values of the r parameter.

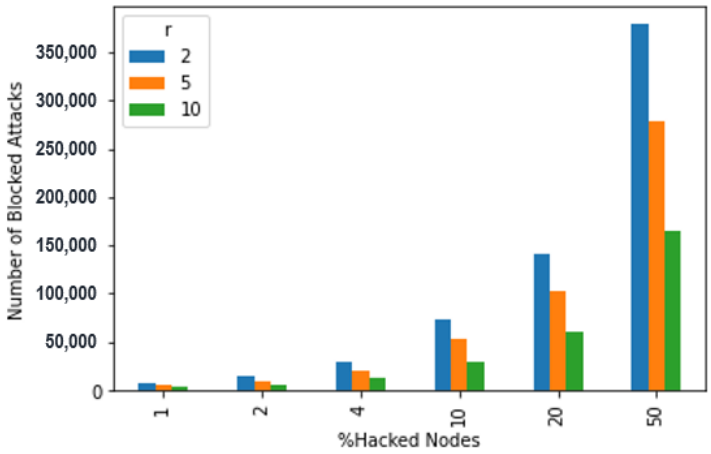

Figure 10. Number of prevented attacks against the number of hacked/corrupted nodes with different values of r

From the analysis of this figure, we can observe that the number of blocked attacks increases exponentially as the percentage of hacked/corrupted nodes increases. Whereas, the impact of r, once again, depends on the average number of neighbors for a node in H2O. In the setting of this experiment, the best trade off between consensus robustness and number of blocked attacks is 5, which is very near to the most probable neighborhood cardinality in the considered H2O network.

6. Security Model of H2O Network

The IoT infrastructure is very prone to well-known security attacks that may impact things and routing [70]. In this section, we illustrate the security model of our approach. In particular, we present both the attack model and a security analysis, showing that our framework addresses its objectives also in presence of threats. The security properties of our scheme are demonstrated by leveraging the characteristic and security features of our approach as well as the assumptions described in the attack model.

6.1. Attack Model

As a preliminary assumption, we consider a realistic scenario in which a sufficient number of nodes is available both in the starting phase, called *safe starting phase*, and in the fully operational one. Therefore, we do not consider anomalous situations, in which the number of the nodes available in the network is less than the minimum necessary one, preventing to implement our scheme successfully. Moreover, we include in our threat model the following assumptions:

A.1 At most $t < r$ smart objects can collude to break the security properties of the protocol.
A.2 A.2 An attacker can neither take control of the whole network, nor own all the smart objects.
A.3 A.3 There is a *safe starting phase* during which no node is corrupted or human impersonated.

As for the first assumption, we recall that our protocol is collaborative and a node n_i can compute the reliability score of another nodes n_j through the responses of the set of its neighbors Γ_{n_i}. Some nodes might be corrupted, but we assume that most of them are honest, as done in [71,72].

The list of the security properties (hereafter, SP) of our framework is the following:

SP.1 Resistance to Node Tampering Attacks.
SP.2 Resistance to Side Channel Attacks.
SP.3 Resistance to Physical Damage Attacks.
SP.4 Resistance to Sniffing Attacks.
SP.5 Resistance to Denial of Service Attacks.
SP.6 Resistance to Fake Node Attacks.
SP.7 Resistance to Sybil Attacks.

6.2. Security Analysis

In this section, we detail the security properties cited above, analysing how our approach can guarantee each of them.

6.2.1. **SP.1**—Resistance to Node Tampering Attacks

This attack occurs when a malicious user has physical access to an object and can modify its whole software/hardware or just part of it. More specifically, an adversary physically compromises a node by inserting malicious code to the node that will give him illegal access to the system [70]. Moreover, this type of attack can include also the altering of some sensitive information, such as cryptographic keys or routing table [73].

Having the control of the node, the attacker can compromise the strategy adopted by the node to gather information and allow for the computation of the fingerprint of its contacts. This would lead to issues during the training phase of the classifiers during the *safe starting phase* and, hence, to wrong fingerprint rates estimation during the subsequent phase. The node, as breached, would provide wrong values once requested by its neighbor. It is worth noting that, in this case, the behavior of the node would not be altered, because the node will continue to work normally, but it will leverage wrong models to produce fingerprint rates. However, thanks to Assumption **A.3** such an attack could not happen, because during the *safe starting phase* all the objects are considered safe.

Always in the context of this kind of attack, instead of jeopardizing the training phase, an attacker could modify directly the response to a query of another object during the operational phase. In this case, our approach enforces two protection mechanisms. On one hand, the use of behavioral fingerprinting models would detect an anomalous behavior of the attacked object; indeed, related literature in this setting proved that an attacker controlling an object would alter some of the features, thus impacting the object's fingerprint [53]. On the other hand, this scenario does not invalidate the computation of the reliability for any node; indeed this score is obtained by a consensus of at least r nodes and for Assumption **A.1** colluding nodes are at most $t < r$.

6.2.2. **SP.2**—Resistance to Side Channel Attacks

Related to the previous category of attacks, in this case attackers leverage low level information like power consumption, time consumption and electromagnetic radiation from sensor nodes to attack encryption mechanisms and have the control of the node. Since our H2O network is made up heterogeneous devices, this attack can target only sensor nodes for which additional information obtained from side channels can reveal encryption mechanisms. However, the countermeasure described above for attacks based on node

tampering (see Section 6.2.1) are valid also in this case. Therefore, thanks to Assumptions **A.3** and **A.1**, this attack cannot happen.

6.2.3. **SP.3**—Resistance to Physical Damage Attacks

The adversary can physically damage the IoT device with the aim to perform a denial of service for that node. If an attacker damage only a node, the reasoning underneath the resistance to this type of attack is the same of the one explained in Section 6.2.1 for Node Tampering Attacks. Hence, once again, thanks to Assumptions **A.3** and **A.1** these attacks are contrasted. Whereas, if the attacker tries to damage all the IoT devices providing a certain services or all the nodes belonging to a neighborhood of a node, the mechanism to assess the reliability could be invalidate for loss of responses. However, for Assumption **A.2** an attacker cannot take control of all the smart objects. Moreover, as already said above, for Assumption **A.1**, colluding nodes can be at most $t < r$.

6.2.4. **SP.4**—Resistance to Sniffing Attacks

An adversary can place a sensor near a simple IoT object (i.e., an object of N_{LPPU} or N_{LPAN}) to sense the connectivity/identity (RFID tag) information exchanged and to act as a man-in-the-middle. In this way, he can send fake data on behalf of the original object and compromise the system [74]. As said, this attack targets only those devices belonging to N_{LPPU} or N_{LPAN}, that are low power devices for either public or personal use. This kind of devices are not directly involved in the construction of fingerprint models, whose construction is instead pursued by objects belonging to N_{PPU} and N_{PPAN} based on the information obtained by other nodes. However, during the *safe starting phase* for Assumption **A.3**, no such attacks can occur, thus guaranteeing the integrity of these models.

As for the subsequent phase, attacked objects will be detected through the available models; moreover, by properly tuning the r parameter, as described in the experimental campaign section, our mechanism can be fully robust to any corrupted node.

6.2.5. **SP.5**—Resistance to Denial of Service Attacks

Classically, this attack may happen when a malicious user overburdens the network with traffic above its capacity and makes it unavailable for legitimate services. To do so, an adversary can execute the attack from a single node (classical DoS [75]) or from multiple nodes (Distributed DoS [76]).

A particular category of DoS attacks, very popular in an IoT scenario, are the Sleep Deprivation Attacks. The objective of the attacker is to cause great consumption of power for a target node to reduce its lifetime. Providing a solution to classical DoS attacks is orthogonal to our proposal. However, any existing strategy, such as those described in [77–79], could be included in our approach. However, it is worth nothing that, in case the attacker should exploit a node for which a behavioral model has been built during the *safe starting phase* to perform DoS attacks against others, the victims can leverage our approach to quickly identify any change in its behavior and, hence, isolate it. Although, this would not solve classical DoS attacks against the network, for which existing state-of-the-art approaches can be adopted, our strategy would provide a solution to Sleep Deprivation ones.

6.2.6. **SP.6**—Resistance to Fake Node Attacks

An attacker can add a fake node to the system. He could do this for a number of reasons, the classical one is injecting malicious data through this fake node in the network, thus making low power devices busy and consuming their energy. If a malicious user wants to perform this kind of attack, he has to add the node during the *safe starting phase*, so that the other nodes can built a fingerprint model for it. But this cannot happen because of the Assumption **A.3**, indeed during the *safe starting phase* all the objects are assumed not malicious.

By contrast, if the attacker tries to add the node during the subsequent phase no object will have a model for it and, consequently, the other nodes cannot evaluate its reliability. In this case, contacting this object may be considered a risk and, therefore, it could be either isolated or not involved in critical applications.

6.2.7. **SP.7**—Resistance to Sybil Attacks

In this kind of attack, an adversary can control a single node, in such a way that this node claims the identity of many nodes and pretends to be them, thus distributing false routing information and/or trying to attack the consensus approach for the reliability assessment. However, thanks to Assumption **A.3**, this attack can happen only in the fully operational phase. At this point, objects will have fingerprint models to evaluate the behavior of their peers. According to the Sybil attack strategy, the malicious node must pretend to be several nodes and, therefore, it cannot maintain a stable behavior. As a consequence, our approach is resistant to this typology of attack.

7. Conclusions

In the last years, IoT devices have gained great autonomy and have become pervasive in everyday tasks. IoT incorporates heterogeneous hardware, communication protocols, and services. In such a diverse and complex ecosystem, numerous security challenges arise. The huge number of devices deployed and connected to the Internet can be hacked, corrupted or stolen. Moreover, also humans, owning smart devices or interacting with them, can be considered actors of the IoT network, especially, in according to the new SIoT paradigm. In such a scenario, each entity must be able to clearly identify and authenticate other entities to assess, before a communication, that neither the user has been impersonated nor the object corrupted. To tackle this issue, in this paper, we have presented a complete framework, called H2O (Human to Object) that provides a mechanism to assess if an object or a human are really who they claim to be. During a *safe starting phase*, the nodes interacting with each others, participate to the construction of suitable fingerprint models. Nodes having access to such models can continue to assess their neighbors reliability also in fully operational state. The other nodes can leverage the knowledge coming from their reliable peers to compute a derived score for nodes that they do not directly know. We evaluate the performances of our approach through an experimental campaign, useful also to tune the system parameters. Moreover, we report the attack model along with the security analysis of our solution, which, ultimately, shows that our framework addresses its objectives also in presence of attacks.

The research issues addressed in this paper can be considered only a starting point for further efforts that we want to perform in the future. For instance, we plan to include in H2O a mechanism to generalize stereotypical fingerprints for specific typology of attackers. In this way, even in the absence of a direct model for a specific object, it could be possible to predict, according to the known attacker stereotypes, whether the node is a genuine or it roughly behaves as a possible attacked node.

Author Contributions: Conceptualization, S.N. and A.N.; Data curation, S.N.; Formal analysis, S.N. and A.N.; Investigation, M.F., S.N. and A.N.; Methodology, S.N. and A.N.; Software, S.N.; Supervision, M.F. and A.N.; Validation, S.N. and A.N.; Writing—original draft, S.N. and A.N.; Writing—review & editing, M.F., S.N. and A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable, the study does not report any data.

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

Abbreviations

The following abbreviations are used in this manuscript:

DoS	Denial of Service
ECG	Electrocardiogram
H2O	Human To human and Object to object
IMU	Inertial Measurement Unit
IoT	Internet of Things
MIoT	Multiple Internet of Things
ML	Machine Learning
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
P2P	Peer to Peer
PAN	Personal Area Network
PDA	Personal Digital Assistant
PKI	Public Key Infrastructure
PPG	Photoplethysmogram
SIoT	Social Internet of Things
SP	Security Property
SWIM	Small World in Motion
RFID	Radio Frequency Identification

References

1. Elkhodr, M.; Shahrestani, S.; Cheung, H. The internet of things: New interoperability, management and security challenges. *arXiv* **2016**, arXiv:1604.04824.
2. Gonzalez-Manzano, L.; Fuentes, J.M.D.; Ribagorda, A. Leveraging user-related internet of things for continuous authentication: A survey. *ACM Comput. Surv. CSUR* **2019**, *52*, 1–38. [CrossRef]
3. Guo, B.; Zhang, D.; Wang, Z.; Yu, Z.; Zhou, X. Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. *J. Netw. Comput. Appl.* **2013**, *36*, 1531–1539. [CrossRef]
4. Kodali, R.K.; Swamy, G.; Lakshmi, B. An implementation of IoT for healthcare. In Proceedings of the 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, India, 10–12 December 2015; pp. 411–416.
5. Nicolazzo, S.; Nocera, A.; Ursino, D.; Virgili, L. A privacy-preserving approach to prevent feature disclosure in an IoT scenario. *Future Gener. Comput. Syst.* **2020**, *105*, 502–519. [CrossRef]
6. Buccafurri, F.; Lax, G.; Nicolazzo, S.; Nocera, A. A privacy-preserving localization service for assisted living facilities. *IEEE Trans. Serv. Comput.* **2016**, *13*, 16–29. [CrossRef]
7. Ball, K. Workplace surveillance: An overview. *Labor Hist.* **2010**, *51*, 87–106. [CrossRef]
8. Shahzad, M.; Singh, M.P. Continuous authentication and authorization for the internet of things. *IEEE Internet Comput.* **2017**, *21*, 86–90. [CrossRef]
9. Pan, S.B.; Moon, D.; Gil, Y.; Ahn, D.; Chung, Y. An ultra-low memory fingerprint matching algorithm and its implementation on a 32-bit smart card. *IEEE Trans. Consum. Electron.* **2003**, *49*, 453–459.
10. Thavalengal, S.; Bigioi, P.; Corcoran, P. Iris authentication in handheld devices-considerations for constraint-free acquisition. *IEEE Trans. Consum. Electron.* **2015**, *61*, 245–253. [CrossRef]
11. Kim, D.J.; Chung, K.W.; Hong, K.S. Person authentication using face, teeth and voice modalities for mobile device security. *IEEE Trans. Consum. Electron.* **2010**, *56*, 2678–2685. [CrossRef]
12. Lee, K.; Byun, H. A new face authentication system for memory-constrained devices. *IEEE Trans. Consum. Electron.* **2003**, *49*, 1214–1222.
13. Anjomshoa, F.; Catalfamo, M.; Hecker, D.; Helgeland, N.; Rasch, A.; Kantarci, B.; Erol-Kantarci, M.; Schuckers, S. Mobile behaviometric framework for sociability assessment and identification of smartphone users. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 1084–1089.
14. Anjomshoa, F.; Aloqaily, M.; Kantarci, B.; Erol-Kantarci, M.; Schuckers, S. Social behaviometrics for personalized devices in the internet of things era. *IEEE Access* **2017**, *5*, 12199–12213. [CrossRef]
15. Abera, T.; Asokan, N.; Davi, L.; Koushanfar, F.; Paverd, A.; Sadeghi, A.R.; Tsudik, G. Things, trouble, trust: On building trust in IoT systems. In Proceedings of the 53rd Annual Design Automation Conference, Austin, TX, USA, 5–9 June 2016; pp. 1–6.
16. Chen, H.; Han, P.; Yu, B.; Gao, C. A new kind of session keys based on message scheme for sensor networks. In Proceedings of the 2005 Asia-Pacific Microwave Conference Proceedings, Suzhou, China, 4–7 December 2005; Volume 1; p. 4.
17. Karlof, C.; Sastry, N.; Wagner, D. TinySec: A link layer security architecture for wireless sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 162–175.
18. Deng, J.; Han, R.; Mishra, S. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Information Processing in Sensor Networks*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 349–364.

19. Buccafurri, F.; Lax, G.; Migdal, D.; Nicolazzo, S.; Nocera, A.; Rosenberger, C. Contrasting false identities in social networks by trust chains and biometric reinforcement. In Proceedings of the 2017 International Conference on Cyberworlds (CW), Chester, UK, 20–22 September 2017; pp. 17–24.
20. Liu, L.; Shi, W. Trust and reputation management. *IEEE Internet Comput.* **2010**, *14*, 10–13. [CrossRef]
21. Hamad, S.A.; Zhang, W.E.; Sheng, Q.Z.; Nepal, S. IoT device Identification via network-flow based fingerprinting and learning. In Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 5–8 August 2019; pp. 103–111.
22. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.R.; Tarkoma, S. Iot sentinel: Automated device-type identification for security enforcement in iot. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184.
23. Aneja, S.; Aneja, N.; Islam, M.S. IoT device fingerprint using deep learning. In Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), Bali, Indonesia, 1–3 November 2018; pp. 174–179.
24. Guennoun, M.; Abbad, N.; Talom, J.; Rahman, S.M.M.; El-Khatib, K. Continuous authentication by electrocardiogram data. In Proceedings of the 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH), Toronto, ON, Canada, 26–27 September 2009; pp. 40–42.
25. Down, M.P.; Sands, R.J. Biometrics: An overview of the technology, challenges and control considerations. *Inf. Syst. Control. J.* **2004**, *4*, 53–56.
26. Bo, C.; Zhang, L.; Jung, T.; Han, J.; Li, X.Y.; Wang, Y. Continuous user identification via touch and movement behavioral biometrics. In Proceedings of the 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 5–7 December 2014; pp. 1–8.
27. Messerman, A.; Mustafić, T.; Camtepe, S.A.; Albayrak, S. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. In Proceedings of the 2011 International Joint Conference on Biometrics (IJCB), Washington, DC, USA, 11–13 October 2011; pp. 1–8.
28. Monaco, J.V.; Bakelman, N.; Cha, S.H.; Tappert, C.C. Developing a keystroke biometric system for continual authentication of computer users. In Proceedings of the 2012 European Intelligence and Security Informatics Conference, Odense, Denmark, 22–24 August 2012; pp. 210–216.
29. Roggen, D.; Wirz, M.; Tröster, G.; Helbing, D. Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods. *arXiv* **2011**, arXiv:1109.1664.
30. Tan, H.; Tsudik, G.; Jha, S. MTRA: Multiple-tier remote attestation in IoT networks. In Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 9–11 October 2017; pp. 1–9.
31. Kuang, B.; Fu, A.; Yu, S.; Yang, G.; Su, M.; Zhang, Y. ESDRA: An efficient and secure distributed remote attestation scheme for IoT swarms. *IEEE Internet Things J.* **2019**, *6*, 8372–8383. [CrossRef]
32. Kostas, K.; Just, M.; Lones, M.A. IoTDevID: A Behaviour-Based Fingerprinting Method for Device Identification in the IoT. *arXiv* **2021**, arXiv:2102.08866.
33. Bezawada, B.; Bachani, M.; Peterson, J.; Shirazi, H.; Ray, I.; Ray, I. Behavioral fingerprinting of iot devices. In Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, New York, NY, USA, 15–19 October 2018; pp. 41–50.
34. Sicari, S.; Rizzardi, A.; Grieco, L.; Coen-Porisini, A. Security, privacy and trust in Internet of Things: The road ahead. *Comput. Netw.* **2015**, *76*, 146–164. [CrossRef]
35. Chen, D.; Chang, G.; Sun, D.; Li, J.; Jia, J.; Wang, X. TRM-IoT: A trust management model based on fuzzy reputation for Internet of Things. *Comput. Sci. Inf. Syst.* **2011**, *8*, 1207–1228. [CrossRef]
36. Ganeriwal, S.; Kumar, R.; Han, C.; Lee, S.; Srivastava, M. *Location & Identity based Secure Event Report Generation for Sensor Networks*; NESL Technical Report; Springer: Berlin/Heidelberg, Germany, 2004 .
37. Chen, J.; Tian, Z.; Cui, X.; Yin, L.; Wang, X. Trust architecture and reputation evaluation for internet of things. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3099–3107. [CrossRef]
38. Pietro, R.D.; Salleras, X.; Signorini, M.; Waisbard, E. A blockchain-based Trust System for the Internet of Things. In Proceedings of the ACM International Symposium on Access Control Models and Technologies (SACMAT'18), Indianapolis, IN, USA, 13–15 June 2018; pp. 77–83.
39. Lin, J.; Shen, Z.; Miao, C. Using blockchain technology to build trust in sharing LoRaWAN IoT. In Proceedings of the International Conference on Crowd Science and Engineering (ICCSE'17), Beijing, China, 6–9 July 2017; pp. 38–43.
40. Hammi, M.T.; Hammi, B.; Bellot, P.; Serhrouchni, A. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. *Comput. Secur.* **2018**, *78*, 126–142. [CrossRef]
41. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]
42. Atzori, L.; Iera, A.; Morabito, G. From "smart objects" to "social objects": The next evolutionary step of the internet of things. *IEEE Commun. Mag.* **2014**, *52*, 97–105. [CrossRef]
43. Nitti, M.; Girau, R.; Atzori, L. Trustworthiness management in the social internet of things. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1253–1266. [CrossRef]

44. Ursino, D.; Virgili, L. Humanizing IoT: Defining the profile and the reliability of a thing in a multi-IoT scenario. In *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 51–76.
45. Sherchan, W.; Nepal, S.; Paris, C. A survey of trust in social networks. *ACM Comput. Surv. CSUR* **2013**, *45*, 1–33. [CrossRef]
46. Gligor, V.; Wing, J. Towards a theory of trust in networks of humans and computers. In Proceedings of the International Workshop on Security Protocols, Cambridge, UK, 28–30 March 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 223–242.
47. Tsikerdekis, M.; Zeadally, S. Detecting and preventing online identity deception in social networking services. *IEEE Internet Comput.* **2015**, *19*, 41–49. [CrossRef]
48. Podobnik, V.; Striga, D.; Jandras, A.; Lovrek, I. How to calculate trust between social network users? In Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2012), Split, Croatia, 11–13 September 2012; pp. 1–6.
49. Adali, S.; Escriva, R.; Goldberg, M.K.; Hayvanovych, M.; Magdon-Ismail, M.; Szymanski, B.K.; Wallace, W.A.; Williams, G. Measuring behavioral trust in social networks. In Proceedings of the 2010 IEEE International Conference on Intelligence and Security Informatics, Vancouver, BC, Canada, 23–26 May 2010; pp. 150–152.
50. Walter, F.E.; Battiston, S.; Schweitzer, F. A model of a trust-based recommendation system on a social network. *Auton. Agents Multi-Agent Syst.* **2008**, *16*, 57–74. [CrossRef]
51. Alvi, S.A.; Afzal, B.; Shah, G.A.; Atzori, L.; Mahmood, W. Internet of multimedia things: Vision and challenges. *Ad Hoc Netw.* **2015**, *33*, 87–111. [CrossRef]
52. Sewak, M.; Singh, S. IoT and distributed machine learning powered optimal state recommender solution. In Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 22–24 January 2016; pp. 101–106.
53. Bezawada, B.; Ray, I.; Ray, I. Behavioral fingerprinting of Internet-of-Things devices. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, e1337. [CrossRef]
54. Guo, B.; Zhang, D.; Yu, Z.; Zhou, X.; Zhou, Z. Enhancing spontaneous interaction in opportunistic mobile social networks. *Commun. Mob. Comput.* **2012**, *1*, 1–6. [CrossRef]
55. Kumar, R.; Phoha, V.; Serwadda, A. Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns. In Proceedings of the 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), Niagara Falls, NY, USA, 6–9 September 2016; pp. 1–8.
56. Dalai, A.K.; Jena, S.K. Wdtf: A technique for wireless device type fingerprinting. *Wirel. Pers. Commun.* **2017**, *97*, 1911–1928. [CrossRef]
57. Oser, P.; Kargl, F.; Lüders, S. Identifying devices of the internet of things using machine learning on clock characteristics. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Melbourne, Australia, 11–13 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 417–427.
58. Meidan, Y.; Bohadana, M.; Shabtai, A.; Guarnizo, J.D.; Ochoa, M.; Tippenhauer, N.O.; Elovici, Y. ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis. In Proceedings of the Symposium on Applied Computing, Marrakech, Morocco, 3–7 April 2017; pp. 506–509.
59. Bai, L.; Yao, L.; Kanhere, S.S.; Wang, X.; Yang, Z. Automatic device classification from network traffic streams of internet of things. In Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 1–4 October 2018; pp. 1–9.
60. Thangavelu, V.; Divakaran, D.M.; Sairam, R.; Bhunia, S.; Gurusamy, M. DEFT: A distributed IoT fingerprinting technique. *IEEE Internet Things J.* **2018**, *6*, 940–952. [CrossRef]
61. Kozik, R.; Choraś, M.; Ficco, M.; Palmieri, F. A scalable distributed machine learning approach for attack detection in edge computing environments. *J. Parallel Distrib. Comput.* **2018**, *119*, 18–26. [CrossRef]
62. Kataria, A.N.; Adhyaru, D.M.; Sharma, A.K.; Zaveri, T.H. A survey of automated biometric authentication techniques. In Proceedings of the 2013 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, 28–30 November 2013; pp. 1–6.
63. Bartlow, N.; Waymire, D.; Zektser, G. Holistic Evaluation of Multi-Biometric Systems. In Proceedings of the 2010 National Institute of Standards and Technology, Washington, DC, USA, 1–5 March 2009.
64. Shanmugapriya, D.; Padmavathi, G. A survey of biometric keystroke dynamics: Approaches, security and challenges. *arXiv* **2009**, arXiv:0910.0817.
65. Stöber, T.; Frank, M.; Schmitt, J.; Martinovic, I. Who do you sync you are? smartphone fingerprinting via application behaviour. In Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, 17–19 April 2013; pp. 7–12.
66. Marche, C.; Atzori, L.; Pilloni, V.; Nitti, M. How to exploit the Social Internet of Things: Query Generation Model and Device Profiles' Dataset. *Comput. Netw.* **2020**, *174*, 107248. [CrossRef]
67. Mei, A.; Stefa, J. SWIM: A simple model to generate small mobile worlds. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 2106–2113.
68. Kosta, S.; Mei, A.; Stefa, J. Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking. In Proceedings of the 2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Boston, MA, USA, 21–25 June 2010; pp. 1–9.

69. Adamic, L.A.; Lukose, R.M.; Puniyani, A.R.; Huberman, B.A. Search in power-law networks. *Phys. Rev. E* **2001**, *64*, 046135. [CrossRef]
70. Ali, I.; Sabir, S.; Ullah, Z. Internet of things security, device authentication and access control: A review. *arXiv* **2019**, arXiv:1901.07309.
71. Fouque, P.; Poupard, G.; Stern, J. Sharing decryption in the context of voting or lotteries. In Proceedings of the International Conference on Financial Cryptography (FC'00), Anguilla, British West Indies, 20–24 February 2000; pp. 90–104.
72. Cramer, R.; Gennaro, R.; Schoenmakers, B. A secure and optimally efficient multi-authority election scheme. *Eur. Trans. Telecommun.* **1997**, *8*, 481–490. [CrossRef]
73. Perrig, A.; Stankovic, J.; Wagner, D. Security in wireless sensor networks. *Commun. ACM* **2004**, *47*, 53–57. [CrossRef]
74. Mitrokotsa, A.; Rieback, M.R.; Tanenbaum, A.S. Classification of RFID attacks. *Gen* **2010**, *15693*, 14.
75. Liang, L.; Zheng, K.; Sheng, Q.; Huang, X. A denial of service attack method for an iot system. In Proceedings of the 2016 8th international conference on Information Technology in Medicine and Education (ITME), Fuzhou, China, 23–25 December 2016; pp. 360–364.
76. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
77. Bouabdellah, M.; Kaabouch, N.; Bouanani, F.E.; Ben-Azza, H. Network layer attacks and countermeasures in cognitive radio networks: A survey. *J. Inf. Secur. Appl.* **2018**, *38*, 40–49. [CrossRef]
78. Chouhan, N.; Saini, H.; Jain, S. Internet of Things: Illuminating and Study of Protection and Justifying Potential Countermeasures. In *Soft Computing and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 21–27.
79. Yang, W.; Wang, Y.; Lai, Z.; Wan, Y.; Cheng, Z. Security Vulnerabilities and Countermeasures in the RPL-based Internet of Things. In Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'18), Zhengzhou, China, 18–20 October 2018; pp. 49–495.

Short Biography of Authors

Marco Ferretti is a Full Professor at the University of Pavia. Marco Ferretti received the Laurea (cum laude) in Electronics Engineering from the University of Pavia (Italy) in 1979, where he is full professor since 1994. From 1984 to 1998 he was Chaiman of the IAPR TC for Special Purpose Architectures and from 2004 to 2008 he was President of GIRPR, the Italian branch of IAPR. He is Fellow of IAPR and member of IEEE. He published in excess of 150 articles in international peer reviewed journals and conference proceedings. He is inventor of 2 international patents in hardware for image compression. He is currently in the advisory board of CINECA that hosts the largest HPC facility in Italy and in the executive board of CINI, a consortium of some 50 universities active in ICT. His current research themes are in the exploitation of cloud resources for HPC and massive data analysis.

Serena Nicolazzo got a PhD in Information Engineering at the University Mediterranea of Reggio Calabria in 2016. From 2016 to 2017 she was Research Fellow at the University Mediterranea of Reggio Calabria. Her research interests include: Data Science, Security, Privacy and Multi-Social Network Analysis. From 2019 she is Research Engineering Senior Engineer in the Architecture and KPI group of Sky Italia. She collaborates with the Innovation Group and she is responsible for the design of end-to-end architectures for Sky services. She is involved in several TPCs and editorial board of prestigious International Conferences and Journal in the context of Data Science and Cybersecurity. Currently, she is a Visiting Researcher at Middlesex University of London and she is collaborating with the Daisy Lab, Polytechnic University of Marche.

Antonino Nocera is currently Assistant Professor at the University of Pavia from February 2019. He received his PhD in Information Engineering at the University Mediterranea of Reggio Calabria in March 2013. He actively collaborates with researchers of prestigious international research and academic institutions in different fields spanning over several research contexts, including: Security, Privacy, Social Network Analysis, Machine Learning, Recommender Systems, Internet of Things, Cloud Computing and Blockchain. In these research fields, he published about 70 scientific papers. He is involved in several TPCs of prestigious International Conferences in the context of Data Science and Cybersecurity and he is Associate Editor of Information Sciences (Elsevier).