A Dynamic Continuous Authentication Framework in IoT-Enabled Environments

Pantaleone Nespoli[1,*], Mattia Zago[1,*], Alberto Huertas Celdrán[2], Manuel Gil Pérez[1],
Félix Gómez Mármol[1], and Félix J. García Clemente[3]

[1]*Departamento de Ingeniería de la Información y las Comunicaciones*, *University of Murcia*, 30071 Murcia, Spain
[2]*Telecommunication Software & Systems Group*, *Waterford Institute of Technology*, Waterford, Ireland
[3]*Departamento de Ingeniería y Tecnología de Computadores*, *University of Murcia*, 30071 Murcia, Spain

***Abstract*—Modern authentication systems still suffer of some limitations that threaten users' data protection. To strengthen the overall security of authentication processes, the continuous authentication paradigm has been increasingly employed. Nonetheless, this fresh methodology still poses a number of challenges that remain unsolved. In this paper, we present a novel framework that is able to provide context-aware IoT-based continuous and non-intrusive authentication and authorization services. To do so, we propose a collection of ontologies that represents the defined information model. Hence, these ontologies are combined together with efficient authentication and authorization policies to build a full-fledged IoT Continuous Authentication Framework (IoTCAF). The conducted experiments demonstrate the feasibility and scalability of the proposed framework leveraging the characteristics of IoT pervasiveness.**

***Index Terms*—continuous authentication, IoT, authorization, security**

I. Introduction

The constant evolution of modern computer systems is undoubtedly changing our lives. Nowadays, they are not only effectively smaller, faster, and easier to use than before, but also they are cheaper and more pervasive. Despite of their benefits, over the last years there are a few critical aspects that have not evolved as expected. Among them, the authentication process can be highlighted as one of the most relevant.

Traditional authentication systems were based on choosing one of the next well-known Authentication Factors (AF) to identify users: some secret that a user knows, some token that a user has, or something that a user is. In contrast, modern authentication solutions combine them in different ways, such as the two-factor authentication system involving two independent channels for authenticating the user (e.g., a smart card and a PIN code). However, these systems have limitations due to their nature:

- *AFs that leverage on user's knowledge* such as passwords or PIN codes. Even if this secret is not trivial to guess and is safely stored, it remains vulnerable to social engineering attacks. Additionally, there is a concrete possibility that the user may forget this authentication secret.
- *AFs that leverage on user's possessions* such as smartcards. Limitations are similar to the previous category: a physical object can be forgotten or stolen. Additionally, the possession of a specific authentication object often conflicts with the usability of the entire authentication system, since the user is forced to bear it.
- *AFs that leverage on user's characteristics* such as fingerprint, or face recognition. The strong aspects of this category represent also their major limitations since biometric measures cannot be lost or changed but at the same time they cannot be revoked and updated.

In order to strengthen the overall security of the authentication process, the continuous authentication paradigm has been increasingly employed. Continuous authentication, also referred to as active authentication, was introduced to propose novel mechanisms to validate users' identity, thus addressing the problems showed by the usual techniques [1]. This methodology can *continuously* authenticate the legitimacy of a user over time by analyzing their behavioral profile. For example, a possible way to identify the user is to evaluate their interaction with a specific device [2]. Despite the numerous advantages, this paradigm still poses some challenges that will characterize future researches, such as accuracy of the authentication process, elapsed time, and processing complexity [3].

Recently, another technology is getting the interest of both the academia and the industry. The Internet of Things (IoT) is as an emerging technology with notable potential of development. IoT is considered as a part of the Internet of the future, where billions of "intelligent objects" will communicate to provide services to humans as an ultimate goal [4]. These devices may be located almost everywhere, from vehicles to buildings, home appliances, or cell phones, passively sensing the environment to collect relevant information. In this context, the versatility and ubiquity of the IoT devices can be used to build a full-fledged authentication system. By leveraging the events gathered from IoT devices, it is possible to create accurate behavioral profiles based on the interactions with the surrounding smart objects during the time. In the literature, most of the presented solutions propose to leverage few dimensions to model users' profile [5]. One could say that combing and correlating the events stemming from different devices would give a higher accuracy to the authentication system, thus making feasible the detection of potential anomalies.

Additionally, the acquisition of the continuous flow of data generated by those ubiquitous devices allows one to

[*] Corresponding authors, email: {pantaleone.nespoli, mattia.zago}@um.es

passively identify and track users, without requiring direct user interactions or any intrusive authentication mechanism. The system confidence in users' identity is hence used as parameter for the authorization services, finally enabling the access to specific resources. This natively permits both to enforce non-repudiations properties and minimize the number of intrusive authorization requests within the framework boundaries.

The main contributions of the paper can be summarized as follows:

- We present a formal definition of the information model for the continuous authentication framework, which represents a solid base to develop the related ontologies for the IoT ecosystem.
- We introduce the authentication and authorization rules, in the form of policies, which are used from the proposed system to effectively identify users and to assign them the right permissions to perform certain actions.
- We propose the IoT-enabled Continuous Authentication Framework (IoTCAF), a novel architecture capable of providing a continuous and non-intrusive authentication and authorization solution for users according with their interaction with the surrounding IoT devices.
- We demonstrate the feasibility of the IoTCAF architecture through experiments on throughput and scalability.

The paper remainder is organized as follows. Section II gives an overview of the current state of the art of the continuous authentication paradigm in IoT scenarios. Section III defines the information model for the continuous authentication framework. Upon that formalization, Section IV characterizes the policies used to authenticate and eventually authorize the users within the system. Finally, Section V presents the IoTCAF architecture and Section VI its experimental evaluation. To sum-up and conclude, Section VII briefly discusses about the outcomes and the potential future works.

II. Related work

Although the IoT solutions are in a growing way, security features surrounding the "smart objects" remain questionable. The enormous amount of information flowing among the IoT devices attracts malicious entities that aim at gaining unauthorized access to unprotected data [6]. In addition, since most of the data are exchanged using wireless protocols, the communications between devices are inherently vulnerable to several attacks (e.g., eavesdropping, man in the middle, jamming, etc.). So, worldwide researchers are struggling to find efficient solutions that are able to address these security challenges. This evidence is reflected in the literature, where several works have been proposed to provide effective authentication methodologies for the IoT ecosystem. Among them, the methodologies that continuously authenticate users and/or devices within the IoT framework look promising, since they passively leverage the continuous data flow to achieve the authentication duties. In the following, the major proposals are analyzed and grouped in three main categories: *IoT authentication schema*, *IoT machine-to-machine continuous authentication*, and *IoT user-to-machine continuous authentication*.

A. IoT authentication schema

In this subsection, some of the most recent proposals regarding the authentication methodologies in IoT environments are analyzed. Firstly, in [7], the authors proposed an encryption-based authentication schema for smart homes. Specifically, the authors introduced a lightweight key establishment protocol aiming at verifying the identity of smart devices. The protocol makes use of symmetric key cryptography, and leverages the presence of a security service provider. The IoT devices exchange the secret though the security service provider, which is also responsible of generating the tokens. Furthermore, in [8], a security framework for smart devices in home appliances is proposed, with particular focus on the authentication procedure. The proposed framework deploys different modules within the devices to guarantee integrity, availability, and authentication of the nodes. The authentication process is performed through a pairing certification-based process, where IoT devices must exhibit an authentication certificate in order to be verified by the system. Similarly, in [9], a certification-based authentication schema is introduced, which takes advantage of IPv6 addresses and Software-Defined Networking (SDN) technologies to identify the *things* in the network correctly. On the other hand, in [10], a data collection mechanism for location-based authentication system is presented. In the context of industrial IoT, the authentication system leverages ambient information collected from the IoT devices to identify a particular node. Specifically, the freshness and the number of the collected data are dynamically evaluated to perform a valid collection and, consequently, a correct identification. Additionally, in [11], the authors present a lightweight authentication schema suitable for resource-constraint IoT devices. The protocol provides mutual authentication between each object and the remote user using nonces and hashes through gateway nodes. In such cases, the presented authentication protocols do not require any encryption primitive or certificate explicitly.

All the above-mentioned contributions are suitable for the IoT ecosystem, since they contemplate the resource-constraint inherent nature of the IoT nodes. Nonetheless, none of the above considers the possibility of employing a constant process of authentication. One could safely argue that a continuous authentication system would be of crucial importance in a dynamic environment such as IoT, where nodes are characterized by high mobility, thus needing a constant identification within the system.

B. IoT M2M continuous authentication

Regarding the Machine-to-Machine (M2M) continuous authentication in IoT, in [2], a lightweight continuous authentication protocol applicable to a variety of IoT environments is introduced. By applying hash function and XOR primitives, the proposed protocol achieves mutual authentication of the devices using light-computational operations. More specifically, the protocol utilizes tokens to support the continuous authentication mode in which the tokens contain the dynamic features calculated from the correspondent devices.

C. IoT U2M continuous authentication

In the context of IoT User-to-Machine (U2M) authentication, in [12], the author proposed a novel context-aware multi-attribute authentication solution to continuously authenticate a user based on contextual information associated with user mobility and the environment for energy utilization management in smart homes. Location and tasks' criticality nature are used as contextual information to select the authentication attributes. User profile is then dynamically updated over time, thus guaranteeing the adaptability of the entire procedure.

Furthermore, a continuous authentication system for wearable glasses, called GlassGuard, was presented in [13]. The system is automatically able to discriminate the real owner of the smart object from a potential impersonator using biometric features taken from touch gestures and voice commands. Experiments conducted on Google Glasses showed above 93% detection rate using the collected features.

Another proposal is presented in [14], where a transparent authentication system using brainwaves as bio-features for IoT networks was proposed. Extracting long-term memory ability from users' brainwaves, the authors collected the bio-features identified in brainwaves as authentication tokens to perform continuous identification in the background transparently.

Although the aforementioned works constitute a substantial progress, very few consider a real scenario in which users do not often possess any particular IoT device (e.g., biometric readers, smart glasses). To this extent, a system that is able to correlate the data coming from IoT devices to authenticate the users in a transparent, passive, and non-invasive manner is missing. This research paper is intended to fill this gap by presenting an autonomous system in which users can be authenticated based on their interactions with the IoT devices.

III. Information system model for continuous authentication in IoT

This section presents some definitions and outlines the information system model, which has been subsequently defined as an ontology to shape the different components of the proposed continuous authentication system.

A. System model

The components describing our IoT-enabled Continuous Authentication Framework (IoTCAF) can be found in Fig. 1, which is composed of a vector of three main components defined as $IoTCAF = (D, P, L)$. In Fig. 1, it is also shown the relationships between such components, in which the IoT Device component makes reference to the set D, Person to P, and Location to the L set. The definition and modeling of each set is described below.

The main element in any kind of IoT ecosystem is the set of devices shaping the scenario and their features to provide certain services to users, and also the system itself. They can be defined as a set of $l \upuparrows$ IoT devices (see Fig. 1), denoted by $D = \{D_1, D_2, ..., D_l\}$, where l is usually a high number of "thin" devices, which could be i) security devices, willing to support protection, ii) sensing devices to acquire given information for monitoring and detecting events of interest, and iii) leisure devices used by users in their daily lives, just to name a few. These IoT devices will provide different services depending on their functionalities, for example, a service such as a surveillance camera, from which to capture videos and images of the environment, or a mail service that users can consume through their smartphones, tablets, or PCs.

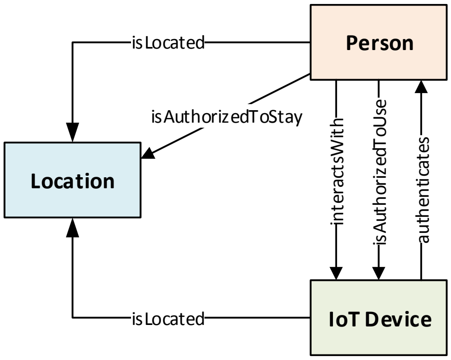

Fig. 1. Main components for continuous authentication in IoT.

Many of the IoT devices are directly used by users, interacting with them for working or leisure purposes, while others are deployed in the environment to enable security functions (e.g., security cameras or proximity devices, among others). Each user is represented in Fig. 1 as Person, whose terms (user and person) are used in this paper interchangeably. Persons are modeled as $P = \{P_1, P_2, ..., P_m\}$, where m is the number of users of the environment. A person $P_i \in P$ can be shown as a user who can interact with IoT devices, whose relationship can be expressed as $D(P_i) = \{D_{i_1}, D_{i_2}, ..., D_{i_x}\}$, although there may also be users who could be authenticated in a continuous way to grant them certain permissions; for example, to stay in a certain area of the environment.

Location also represents an important component that needs to be modeled, since the continuous authentication system should consider location-based information in its decision-making processes. Depending on where users are, or the IoT devices' placement, decision-making processes could grant or deny the corresponding request. Fig. 1 depicts certain links between Location and Person and IoT Device in order to trace both users and devices, respectively, within the managed environment. In this sense, location is modeled as $L = \{L_1, L_2, ..., L_n\}$, where n is the number of places such as areas, rooms, etc. Given that users and IoT devices expose a tight relationship with location, in order to know in which part of the environment they are located, such locations are modeled as $L(P_k) = \{L_{k_1}, L_{k_2}, ..., L_{k_z}\}$ and $L(D_j) = \{L_{j_1}, L_{j_2}, ..., L_{j_y}\}$, respectively. For the sake of simplicity, this subsection presents only the model of a reduced number of relationships between components, which will be presented in detail in Section III-B.

It is worth noting that the system model introduced in this section, defining the main components shown in Fig. 1, needs to be extended with finer granularity in order to characterize them in more specific subcomponents. For example, the IoT devices in D should be defined in more detail depending on their functions and features, such as security devices, sensing devices, leisure devices, etc., as mentioned above.

B. Ontology

Based on the modeling of the main components of the system described above, a collection of ontologies has been developed to enable the sharing of knowledge between the three main components (see Fig. 1) and the use of semantic reasoning procedures to infer new knowledge according to the information gathered by the IoT devices.

Fig. 2 shows the collection of ontologies designed, developed, and managed by our continuous authentication framework IoTCAF, which formally shapes the three main concepts managed by the framework:

- *Location Ontology* modeling a given smart space, which is structured, for example, in different parts of a smart office or home;
- *Person Ontology* to represent any user who is under the framework domains and gets the benefits it offers when using the IoT Devices; and
- *IoT Devices Ontology*, which constitutes the central model of the architecture, as it is composed by the devices providing information about the scenario status and the ones that will apply the reactions decided by the framework to grant or deny actions to users.

In each ontology, the top-level class of each of them is shown by following the same colors used in Fig. 1.

The Location Ontology is defined through several subclasses to refine the different spaces in which a given environment is structured, where *Location* is the top-level class from which the rest of subclasses inherit. As shown in Fig. 2, this ontology uses a hierarchical model for shaping location, using four subclasses with different levels of size and detail; namely (from the largest to the smallest): *Building*, *Block*, *Floor*, and *Area*. The latter has in turn been divided into more specific spaces: *Hallway*, *Stairs*, and *Room*. The location hierarchical model has been modeled taking into account that our system is oriented to smart offices and smart homes, so that other structures in space could have been modeled considering other location-related subclasses.

The Location class of the Location Ontology is linked to the top-level class of the other two ontologies through several properties. These properties determine the location in which a Person or an IoT Device is located, by making use of the *isLocated* property. In addition, the *isAuthorizedToStay* property has also been defined to determine whether a given Person can be in a certain Location. Here, it is important to highlight that this property is one of the consequences of authorization policies defined in Section IV).

The top-level class in the Person Ontology is *Person*, which presents two different properties (*hasRole* and *hasAuthLevel* with the *Role* and *Authentication Level* classes, respectively) with the aim of knowing whether the Person has sufficient roles and authentication levels i) to use and interact with a given IoT Device (*isAuthorizedToUse* property with the *IoT Device* class) and ii) to be located in a certain location (isAuthorizedToStay property with the *Location* class). These last two properties are the consequences of the authorization policies described in Section IV-B, which are created at runtime by the framework administrator.

Finally, *IoT Device* is the top-level class of the IoT Devices Ontology, which is categorized into two main subclasses that inherit from the first. In particular, IoT devices have been modeled depending on whether they are user-dependent devices or not; that is, devices with which users interact (e.g., tablets or smart lock devices requiring the user's fingerprint) or devices deployed in the environment infrastructure (with which users do not interact) to report information that will be further analyzed (e.g., surveillance cameras or proximity devices). Furthermore, the *State* class models the situation of IoT Devices in a given moment of time. Among the possible states, we highlight some of them such as active, interacting, authenticated, in standby, or switched on/off. On the other hand, the *Service* class presents two links to define the different services that a given user could use (relationship with Person) and to establish the list of services that are provided by such devices (relationship with the IoT Device class).

IV. Policy-based decision making system

The proposed solution continuously authenticates and authorizes users to stay in certain spaces or use different IoT devices by using semantic rules, which form policies. The proposed architecture uses rules composed of two lists of predicates, the antecedent, and the consequent. If all predicates of the antecedent part take the Boolean value true, all predicates in the consequent part are evaluated. It is important to know that in our semantic rules the predicates in the consequent part establish new relationships between entities of the ontologies, no generating new entities.

Our policies are composed of the following elements: *Type* is the kind of policy; *Target* is the person considered by the policy to be authenticated or authorized; *Location* is the place or environment in which the policy is applied; and *Result* determines the relationship that the Target will have with the IoT Device or Location regarding authentication and authorization grants. Note that Result is the consequent part of the semantic rule, while the rest of fields belong to the antecedent part [15].

Our framework manages two kinds of policies: *Authentication* and *Authorization* policies. Both families are defined by the Framework Administrator to decide the authentication and authorization of the users to stay in a given space or use specific IoT Devices. Below we show an example for each one of these policies.

A. Authentication policies

Authentication policies consider the state of IoT devices, which is influenced by the actions performed by users in the environment, to make decisions about the level of authentication that users have in a given moment. Different levels

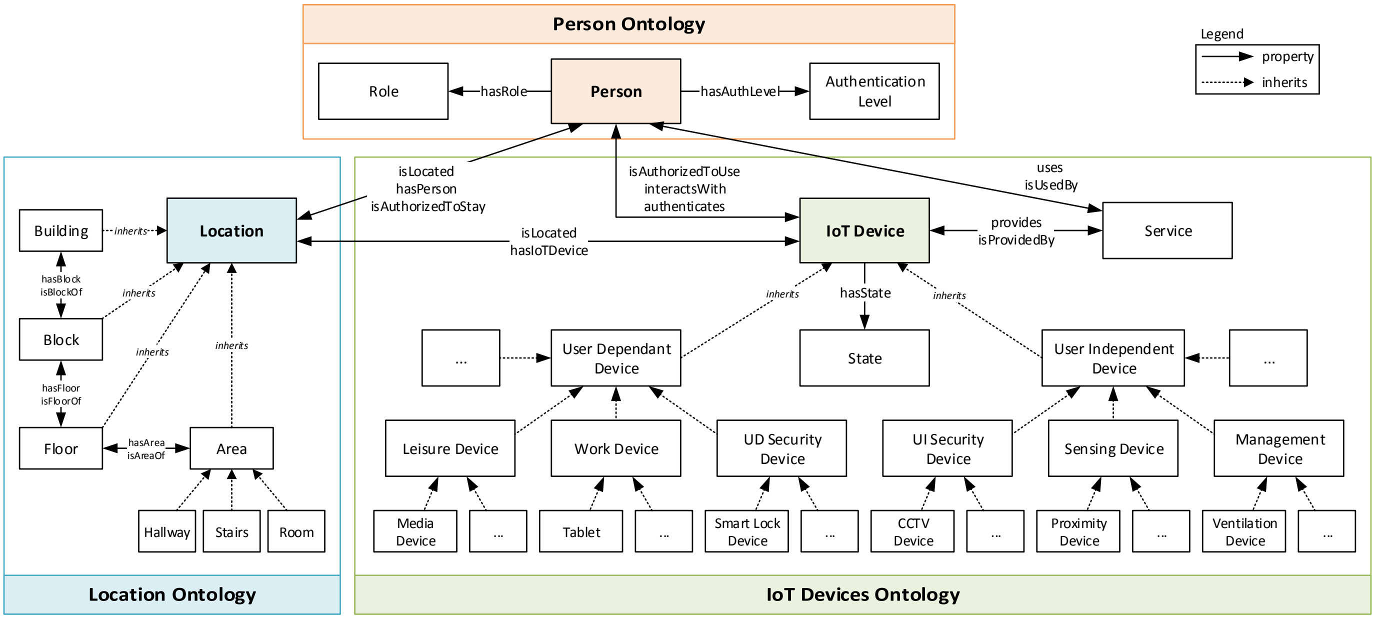

Fig. 2. Set of ontologies making up the continuous authentication framework: Location, Person, and IoT Devices ontologies.

of authentication are generated by these policies according to the user's behavior patterns. These patterns are created, in a transparent way for the users, by the IoT devices allocated in the user's environment or context. By default, the proposed architecture establishes the lowest level of authentication in absence of policies. The antecedent of these policies considers different elements belonging to the three proposed ontologies (Person, Location, and IoT Devices) and the consequent part generates relationships between entities belonging to the Person ontology. As an example, the next policy indicates that persons will be continuously authenticated with the parent level when they have associated the role *Parent*, they consume services provided by a *tablet* located in the *SmartHome* and the security device of the home (an alarm or CCTV smart devices, for example) has already authenticated them.

$$\begin{aligned}
&IoTDevice(?secDevice) \wedge\\
&hasIoTDevice(\#SmartHome, ?secDevice) \wedge\\
&hasState(?secDevice, \#Authenticated) \wedge\\
&Person(?person) \wedge\\
&authenticates(?secDevice, ?person) \wedge\\
&isLocated(?tablet, \#SmartHome) \wedge\\
&hasState(?tablet, \#Active) \wedge\\
&provides(?tablet, ?service) \wedge\\
&isUsedBy(?service, ?person) \wedge\\
&hasRole(?person, \#Parent) \ \rightarrow\\
&hasAuthLevel(?person, \#ParentLevel)
\end{aligned}$$

B. Authorization policies

Authorization policies take into account the level of authentication, provided by Authentication policies, to allow users to stay in certain locations or use specific devices located in the users' environment. By default, the proposed architecture denies the authorization in the absence of rules. As an example, the next policy authorizes persons to stay in the *SmartOffice* and use its *IoT Devices* when they are located in that room with the *Workerlevel* of authentication.

$$\begin{aligned}
&Person(?person) \wedge\\
&hasAuthLevel(?person, \#WorkerLevel) \wedge\\
&isLocated(?person, \#SmartOffice) \wedge\\
&hasIoTDevice(\#SmartOffice, ?ioTDevice) \ \rightarrow\\
&isAuthorizedToUse(?person, ?ioTDevice) \wedge\\
&isAuthorizedToStay(?person, \#SmartOffice)
\end{aligned}$$

V. IoTCAF Architecture

This section shows the proposed architecture, which is able to provide users with a continuous unintrusive authentication and authorization solution according to their interaction with heterogeneous IoT devices. To reach the transparent authentication and authorization processes, IoTCAF consists of several modules organized in three different layers according to their functionalities. Fig. 3 illustrates the *Data*, *Management*, and *Service* Layers composing the presented architecture.

a) Data Layer: This lower level includes all the *IoT Devices* belonging to the smart-environment. Each device outputs a stream of events according to the device purpose. Such events might be perceived either as distinct events generated upon interaction between the user and the device (e.g., a RFID reader) or as a continuous flow of information (e.g., a CCTV camera). As modeled by the ontology specified in Section III-B, there exist two main categories of IoT devices, according to which the events' stream can be either sporadic or continuous. These events are filtered and all the information that might be used for authentication purposes is sent to the *Collector* module in the Management Layer.

A *Location Middleware* module is taking care of locating all the devices belonging to the framework. For example, a device such as a CCTV camera might not be able to provide a localization service, so it has to be located through other

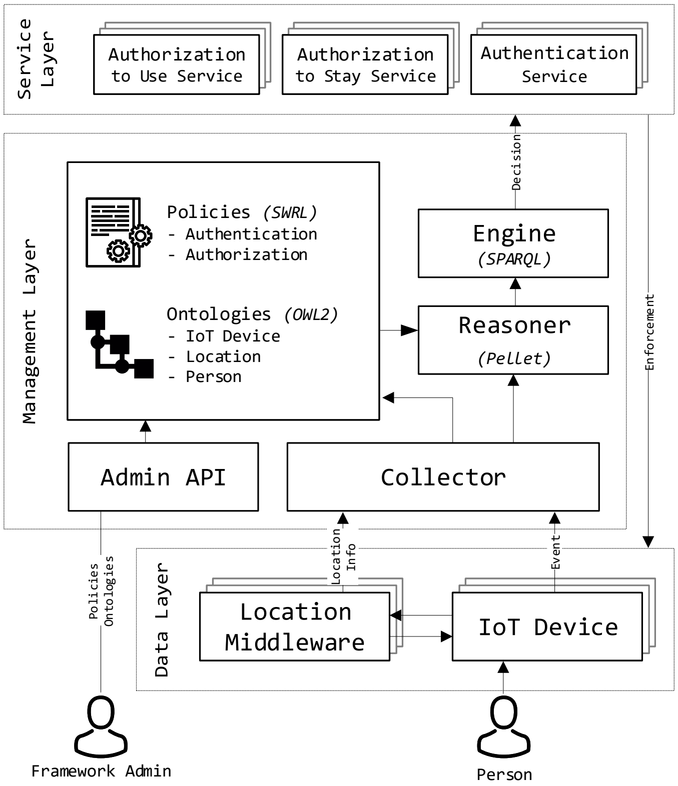

Fig. 3. Overview of the IoTCAF multilayered architecture.

information, such as the unique identifier and the installation location. As for the *IoT Devices*, the location data are sent to the *Collector* that will interpret them according to the previously mentioned ontology (Section III-B).

b) Management Layer: This layer represents the intelligent core of this framework. It provides several features, among them the possibility of modeling according to a standard format (the ontology) all the events generated by the *Data* layer. This function is required in order to be able to provide both the authentication and the authorization services responsible for controlling the IoT devices. In this context, these capabilities enable the framework to authenticate users according to their behavior and interaction with the devices, without losing the capabilities of providing layered security. It is clear, in fact, that the layer receives different inputs from multiple devices that are collected, integrated, and analyzed. Two main sources of knowledge for the *Reasoner* module are confined, namely the *Policies* and the *Ontologies* databases. The former contains the definition of all the policies structured in two groups, one for the authentication (Section IV-A) and one for the authorization (Section IV-B); the latter includes the three ontologies of IoT Device, Location, and Person as described in Section III-B. However, the control flow starts with the location information and events' streams that are collected by the *Collector* module, which takes care of interpret and save them in the aforementioned ontologies database.

The information is then used by the *Reasoner*, whose implementation details are highlighted in Section VI, which ultimately triggers a decision and a reaction in the *Engine*. The decision is taken according to the policies specified in the aforementioned database applied over the collected events modeled upon the class-specific ontology. The decision is hence enforced by retro-fitting the reaction for the *Data* layer. Finally, an administrative API is available for human administrators in order to be able to configure (i.e., add, edit or delete policies and ontologies) and tune (i.e., frequency of update, etc.) the system.

c) Service Layer: This layer contains the set of services provided by our solution to interact with different IoT devices and authenticate or authorize users to use specific devices or stay in given places. Here it is important to highlight that some devices such as boundary authentication gateways do not require these services because they are used firstly to identify users. So, our solution has a specific authentication and authorization service for each IoT device. To be able to manage this process, some devices expose APIs that permit the authentication and authorization of the users that are interacting with them. Multiple services can be included later on for increasing the maneuverability and the reactiveness of the system to external inputs.

d) Actors: The main actors included in this architecture are the *Framework Admin*, whose tasks include the configuration and the maintenance of the policies and ontology database, and the *Person* who may interact with the devices. The benefits for persons are clear, since avoiding intrusive authentication and authorization procedure greatly simplifies user experience without jeopardizing the environment security. In fact, independently from their active interaction, the framework is built so to be able to recognize and profile them, in order to provide real and continuous authentication.

VI. Deployment and Experimental results

We have deployed IoTCAF to validate its proper functioning and measure its throughput and scalability. In this context, the representation of the information (ontologies and policies) and the decision-making process (Reasoner and Engine) are based on Semantic Web techniques, where Location, Person, and IoT Device ontologies, shown in Section III, are defined in OWL 2 (Web Ontology Language) [16] and have been generated with the Protégé tool [17]. We have chosen OWL 2 rather than other languages like RDF, RDFS, or DAML+OIL because OWL 2 is more expressive than the rest. It was specifically designed as an ontology language, being an open standard, and the main ontology language used nowadays in Semantic Web. On the other hand, semantic rules defining the policies of Section IV are expressed in SWRL (Semantic Web Rule Language) [18]. SWRL includes a type of axiom, called *Horn clause logic*, of the form $if...$ *then...*, being the most widely used solution in Semantic Web today.

The proposed architecture makes decisions about the authentication and authorization of users according to the previous ontologies and semantic rules. For that, a semantic reasoner, implemented by the Reasoner component of Fig. 3, infers new knowledge that decides whether a given user is authenticated, authorized, or not. We use Pellet [19] as semantic reasoner, which receives ontological models with the information shaped by the ontologies and policies. Finally, the Engine component of the proposed architecture is in charge of applying periodical queries, performed in SPARQL [20], to the inferred model and gets the result about the authentication and authorization of users.

Once the decision-making process has been explained, we conducted several experiments with the aim of measuring its throughput and scalability. These experiments were intended to deal with two questions:

- Is the decision-making process time acceptable?
- How it scales with different IoT devices and authentication/authorization policies?

As experimental setting, the proposed framework and the conducted tests were carried out in a dedicated PC with an Intel Core i7-3770 3.40 GHz, 16 GB of RAM, and an Ubuntu 16.04 LTS as operating system. The results shown in this section have been obtained by executing the experiments 100 times and computing their arithmetic mean.

A way to measure the performance of the decision-making process is doing executions with different level of complexity. This complexity is related to the number of individuals present in the ontologies and the number of semantic rules making up the policies. Increasing the number of individuals and semantic rules will provoke an increment on the number of statements, and thus on the complexity of executions. The number of individuals contained in our ontologies is referred as *population*. This was randomly generated, but in a controlled way to achieve a real distribution of the elements composing the environment. TABLE I depicts the number of elements used in our environment and their percentages.

TABLE I
INDIVIDUAL DISTRIBUTION OF POPULATION.

Element	**Amount**	**Percentage**	**Element**	**Amount**	**Percentage**
Buildings	1	0.1%	Persons	4	0.1%
Floors	4	0.2%	Roles	10	0.3%
Areas	20	0.6%	IoTDevices	1,000	31.0%
Sections	80	2.5%	Others	100	3.1%
Positions	2,000	62.1%	**Total**	**3219**	**100%**

Another important aspect, related to the second question highlighted in this section, is to evaluate how the scalability of the decision-making process is. With this goal, we defined an initial population of 30,000 individuals, which is increased with other 30,000 individuals in each step. Table II shows the complexity of the proposed ontologies (relationships between the individuals and the statements generated by the semantic reasoner). As observed, the number of statements is proportionally increased according to the number of individuals.

Fig. 4 depicts the time, measured in milliseconds (ms), used by the semantic reasoner to validate the ontology considering different population groups (shown in Table II).

TABLE II
NUMBER OF INDIVIDUALS AND STATEMENTS PER POPULATION.

Population	**0**	**1**	**2**	**3**	**4**
Individuals	30,000	60,000	90,000	120,000	150,000
Statements	352,532	710,004	1,065,537	1,465,409	1,804,336

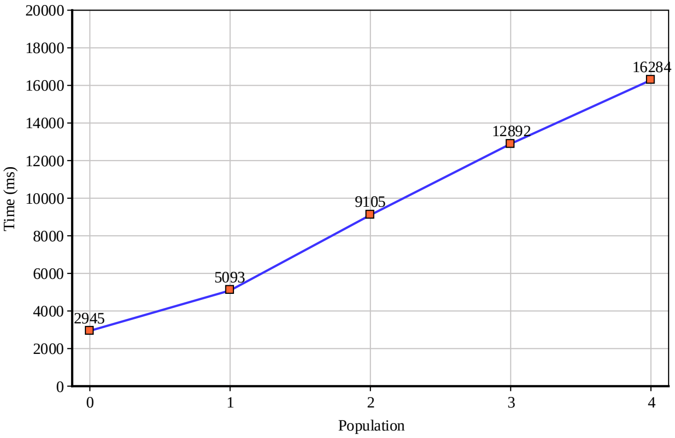

Fig. 4. Consistency checking time.

Comparing the increase of individuals and statements with the time required by the decision-making process, we can observe that the proposed solution can support a very large number of individuals or statements within a reasonable time. Furthermore, the linearity property behind these results allows us to deduce that a better computer system setting would obtain lower reasoning times.

The previous experiment has demonstrated a linear relationship between individual/statements and the reasoning time, but without considering policies. Thus, the main goal behind the next test is to check how policies affect to the scalability of the proposed solution. In this sense, we defined several percentages of policies related with the persons contained in our population groups.

Fig. 5 shows the variation of the time required by decision-making process when the population (see Table II) and the number of policies change. As we can see, policies have a very low impact in our framework. For all populations, the difference between having 10 and 200 policies per user is around a few milliseconds.

As main conclusion of this section, we have demonstrated with the previous experiments that when the number of individuals/statements is linearly increased in our ontology, the decision-making process time also increases linearly. Furthermore, the semantic rules that form the policies do not have an important impact on the decision-making process time.

VII. CONCLUSIONS AND FUTURE WORK

Despite the numerous efforts to develop successful authentication systems, a number of challenges still remain unsolved. In this regard, the paper at hand leverages the multiple benefits of the IoT paradigm to propose IoTCAF,

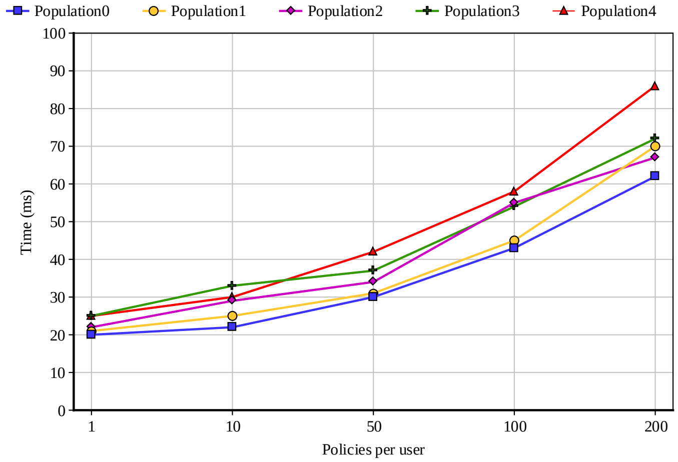

Fig. 5. Reasoning time for different populations and policies.

a novel IoT-enabled continuous authentication framework. In particular, we developed an ontology to formally model IoT scenarios and designed an architecture to deal with the current status of a given IoT scenario, as well as administrator policies to authenticate and authorize users seamlessly and in a non-intrusive fashion. The conducted experiments have demonstrated the suitability of our solution.

As future work we intend to apply machine learning techniques to accurately build behavioral profiles of users and integrate them within IoTCAF.

Acknowledgment

This work has been supported by a Leonardo Grant 2017 for Researchers and Cultural Creators awarded by the BBVA Foundation; by a predoctoral and post-doctoral INCIBE grants within the “Ayudas para la Excelencia de los Equipos de Investigación Avanzada en Ciberseguridad” Program, with codes INCIBEI-2015-27353 and INCIBEI-2015-27352, respectively; by a Séneca Foundation grant within the Human Resources Researching Postdoctoral Program 2018; by a Government of Ireland Postdoctoral Fellowship 2018; and by a Ramón y Cajal research contract (RYC-2015-18210) granted by the MINECO (Spain) and co-funded by the European Social Fund.

References

[1] D. Dasgupta, A. Roy, and A. Nag, *Continuous authentication*, 2017, pp. 235–279.

[2] Y.-H. Chuang, N.-W. Lo, C.-Y. Yang, and S.-W. Tang, “A lightweight continuous authentication protocol for the Internet of Things,” *Sensors*, vol. 18, no. 4, pp. 1–26, Apr. 2018.

[3] H. Khan, A. Atwater, and U. Hengartner, “A comparative evaluation of implicit authentication schemes,” in *Research in Attacks, Intrusions and Defenses*, 2014, pp. 255–275.

[4] S. Li, L. D. Xu, and S. Zhao, “The internet of things: a survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015.

[5] M. Ehatisham-ul Haq, M. A. Azam, J. Loo, K. Shuang, S. Islam, U. Naeem, and Y. Amin, “Authentication of smartphone users based on activity recognition and mobile sensing,” *Sensors*, vol. 17, no. 9, pp. 1–31, 2017.

[6] M. A. Burhanuddin, A. A.-J. Mohammed, R. Ismail, M. E. Hameed, A. N. Kareem, and H. Basiron, “A review on security challenges and features in wireless sensor networks: IoT perspective,” *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 1-7, pp. 17–21, 2018.

[7] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, “Lightweight and secure session-key establishment scheme in smart home environments,” *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, Jan. 2016.

[8] W. M. Kang, S. Y. Moon, and J. H. Park, “An enhanced security framework for home appliances in smart home,” *Human-Centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–12, Mar. 2017.

[9] H. Khemissa and D. Tandjaoui, “A lightweight authentication scheme for e-Health applications in the context of Internet of Things,” in *9th International Conference on Next Generation Mobile Applications, Services and Technologies*, Sep. 2015, pp. 90–95.

[10] Y. Kawamoto, H. Nishiyama, N. Kato, Y. Shimizu, A. Takahara, and T. Jiang, “Effectively collecting data for the location-based authentication in Internet of Things,” *IEEE Systems Journal*, vol. 11, no. 3, pp. 1403–1411, Sep. 2017.

[11] H. Khemissa and D. Tandjaoui, “A novel lightweight authentication scheme for heterogeneous wireless sensor networks in the context of Internet of Things,” in *2016 Wireless Telecommunications Symposium*, Apr. 2016, pp. 1–6.

[12] U. S. Premarathne, “Reliable context-aware multi-attribute continuous authentication framework for secure energy utilization management in smart homes,” *Energy*, vol. 93, pp. 1210–1221, 2015.

[13] G. Peng, G. Zhou, D. T. Nguyen, X. Qi, Q. Yang, and S. Wang, “Continuous authentication with touch behavioral biometrics and voice on wearable glasses,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 404–416, Jun. 2017.

[14] L. Zhou, C. Su, W. Chiu, and K. H. Yeh, “You think, therefore you are: Transparent authentication system with brainwave-oriented bio-features for IoT networks,” *IEEE Transactions on Emerging Topics in Computing*, pp. 1–11, 2017.

[15] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, “Enabling highly dynamic mobile scenarios with Software Defined Networking,” *IEEE Communications Magazine*, vol. 55, no. 4, pp. 108–113, Apr. 2017.

[16] B. Motik, P. F. Patel-Schneider, and B. Parsia (ed.), “OWL 2 web ontology language: Structural specification and functional-style syntax (2nd ed.),” W3C Recommendation, Dec. 2012.

[17] Stanford Center for Biomedical Informatics Research, “Protégé: A free, open source ontology editor and knowledge-base framework,” [Online]. Available: http://protege.stanford.edu.

[18] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A semantic web rule language combining OWL and RuleML,” W3C Member Submission, May 2004.

[19] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical OWL-DL reasoner,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, Jun. 2007.

[20] E. Prud’hommeaux and A. Seaborne (ed.), “SPARQL query language for RDF,” W3C Recommendation, Jan. 2008.