GLOBAL IDENTITY MANAGEMENT IN THE INTERNET OF THINGS

Ausama A. Majeed

Submitted in Partial Fulfilment of the Requirements of the Degree of Doctor of Philosophy in

Computer Science

School of Computing, Science, and Engineering
University of Salford, UK

2018

Table of Content

List of figures

List of Tables

Acknowledgements

First of all, I thank Allah for all his blessings, I would not have completed my PhD without his guidance and success.

I would like to express my special appreciation and thanks to my supervisors Dr. Adil Al-Yasiri and Prof. Nigel Linge, for their support, encouragement, and guidance though out this PhD research journey. Special thanks goes to Dr. Al-Yasiri who I have learned a lot from him, without his help I could not have finished my thesis successfully.

My deepest gratitude and appreciation to my parents, brothers, wife and children for their help and support in difficult times. Your love, understanding and patience sustained me through to the end of my PhD. Without your constant support, this PhD research would have been much difficult to complete.

I would also like to thank the University of Thi-Qar for giving me this opportunity and providing me with a full scholarship to complete my PhD studies. In addition, thanks to the Ministry of Higher Education and Scientific Research of Iraq and the Iraqi cultural attaché in London for their services and support.

Last but not lease, I would like to thank the University of Salford and their staff for all the help and the support provided during my PhD studies.

Husama A. Majeed

List of Abbreviations

IoT	The Internet of Things
IdM	Identity Management System
IdP	Identity Provider
SP	Services Provider
EA	The Effective Actor
co	The communication object
IPv4	The Internet Protocol version 4
IPv6	The Internet Protocol version 6
ITU	International Telecommunications Union
PKI	Public Key Infrastructure
M2M	Machine-to-Machine
$GARI$	Global Actors Relationship Identifier
$GIdM$	Global Identity Management System
$GIdV$	Global Identity Verification Protocol
CoT	Circle of Trust
AR	Actor Relationship
ARS	Actor Relationship Specifier
BAN logic	Burrows–Abadi–Needham logic

Abstract

In the Internet of Things (IoT), objects are seamlessly interconnected by anyone, anywhere, and anytime on behalf of user(s) as an effective actor (EA) for the communication. An actor in the IoT is any identified entity, which needs to be interacting with other entities using the Internet technologies. The service providers (SPs) need to truly establish the EA identity behind the communicated object(s) to offer him/her the right service, which is the vision of the IoT. Theoretically, actors could have different identity attributes and identifiers that are managed by different Identity Management systems ($IdMs$) in every domain they interact with. These $IdMs$ are not always interoperable with each other because they often use different identity attributes and identification systems, which causes that identities are unrecognized across their IdM domains. This can have an impact on the SPs ability to establish the EA identity across their domain, which is a key to realize the IoT. Moreover, the communicated objects identities are widely used as an alternative or secondary identity for their users based on fixed relationship between the user and their devices that can also be used to identify their EAs identities. However, the actor relationships are not always fixed in the IoT; they can be changed or even revoked. This make identifying the actual requester (EA) identity in the IoT a challenge task facing the SPs. Hence, it is important to consider them when identifying the EA of the communicated object in the IoT. This research addresses the SPs difficulty to truly establish the EA identity behind the communicated objects to offer the right services in the IoT environment.

This research proposes a new identification technique to facilitate the establishment of the actual requester's (i.e. the EA) identity behind the communicated object by the SPs in the IoT. This technique requires the existence of four identity parameters for the interacted actors, which are the actor type, Internet connectivity, identifier, and the identity provider (IdP) identifier. Moreover, the actor relationship type between the EA and the communication object(s) that are used to request services or data in the IoT environment has been determined. Thus, a new semantic identifier called a global actor relationship identifier ($GARI$) is formulated to represent the actors identity that are participating in a relationship and the actor relationship type between them. Furthermore, to solve the $IdMs$ interoperability across-domain, a global identity management system ($GIdM$) is proposed to consolidate the $IdMs$ in the IoT environment by using distributed trusted third parties. $GIdM$ includes the design of a new protocol called a global identity verification protocol ($GIdV$). $GIdV$ facilitates the establishment of a dynamic trust relationship and the validation of the EA identity based on the relationship type and a set of identity attributes.

To prove the concept, a testing environment has been built to mimic requesting services or data across-domain in the IoT environment. The simulation testing proves the effectiveness of the developed solutions ($GARI$ and the $GIdM$ system) to establish the EA identity in the IoT environment using the basics scenarios of interaction. Moreover, the comparison with the state of the art identifiers in the IoT shows that the $GARI$ is the only one that presents the interacted actors identity parameters along with their relationship(s) type to use in the IoT environment. Therefore, the $GIdM$ with the $GARI$ is the most suitable IdM that supports the SPs to establish a required trust relationship and verify the EA identity across-domain in the IoT environment based on the actor identity attributes and the relationship(s) type in the IoT environment.

Chapter 1.

Introduction

1.1.Internet of Things: Vision and Concept

The Internet of Things (IoT) has become a technological revolution in the communication and computing fields. IoT has widely attracted researchers from academia and industrial sectors due to its variety of applications [1]. IoT is the environment of integrated "***things***" with electronics, software, sensors, and actuators; which interact with each other via the Internet. They are capable of monitoring, collecting, sharing, analysing, and performing an action, whilst also offering smart services in real-time without any human intervention [2][3].

The term was created by Kevin Ashton to represent connecting physical objects via ubiquitous sensors and a real-time service platform to improve the lives of human beings [4]. In recent years, the IoT has been used to represent the global interconnection of heterogeneous entities. The IoT environment implies physical and logical entities that seamlessly interact to build an information network. Thus, advanced and smart services are provided to its services requesters [5]–[7]. IoT expands the existing interaction patterns to imply human-2-human, human-2-thing, and thing-2-thing.

The standard IoT architecture contains three standard layers which are perception, network, and application layers [8]. The primary role of the perception layer is to sense the context and collect data, to identify the interacted entities, and to perform actions using specific equipment such as sensors, actuators, RFID tags and readers. The network layer function is to transfer the collected data in the perception layer to the application layer, which implies the use of the wire/wireless networks and the Internet. Finally, the application layer contains intelligent solution and application to analyse the received data and do the required function(s).

The IoT is defined by the European Research Cluster on IoT (IERC) [2], [9], [10] as:

"*A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network.*

In the IoT, "things" are expected to become active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information "sensed" about the environment, while reacting autonomously to the "real/physical world" events and influencing it by running processes that trigger actions and create services with or without direct human intervention."

The above definition, identifies the expected roles and capabilities of the "***things***" in IoT. However, the research community believes that there is no "one size fits all" entities to become "things" in IoT. The IoT implies entities such as people, cars, smart objects, devices, applications/services, places, sensors. Entities are diverse regarding their technical specifications, computing and communication capabilities, and deployment fields. Gartner [11], in 2014, classifies things in IoT into two main categories based on their computing and Internet connectivity.

- Smart: A *smart thing* represents the entity with the most computing resources on-board. It is capable of standing alone and communicating through the Internet to share the information and/or receive control instructions remotely. Such things can be found in asset-intensive fields of utilities such as construction, agriculture, infrastructure and transportation. Appendix A explains the smart thing characteristics.
- Dumb: A *dumb thing* represents an entity that wholly or partially relies on another smart thing to perform the computing and communicate through the Internet. They sense data, transfer it through their communication portal, and implement received commands. This type of thing is widely used in many IoT industrial applications such as building systems, utilities, smart city/home, traffic management, and mobile health-monitoring systems.

From a conceptual point of view; the IoT's systems stand on three pillars [12], [13] as listed below:

1. ***Thing identification*****:** things have to be identified by using some form of identification to distinguish them distinctly in the application context.

2. ***Thing communication*: things have to seamlessly communicate (mainly using wireless technologies) with each other to create an environment of interconnected objects using the Internet networks;
3. ***Thing interaction***: things have to interact with their deployment environment based on their sensing, computing, or actuating capability

The realisation of the IoT requires the existence of these three pillars together in any participating entities. However, some of the entities can be communicated directly, and others indirectly using suitable communication technology. They follow the bases *any time, any object, any service, any place over any network* [14]. The Internet is a communication medium of entities oblivious to the underlining technology being used, for instance the communication between RFID tag and reader, or between a Fitbit and a smartphone. Other entities have to interact with these communicated devices to transmit their valuable data and/or perform some actuating, for instance, an older adult who interacts with his health monitor device, or a smart home system which interacts with a smoke sensor to report the fire to the nearest station. However, all entities have to be identified before allow them to access services or resources in an IoT context [15], [16]. Truly identifying the entity is the first security challenges faces the success of the IoT objective of forming "a better world for human beings" [6], [7]. They need to be globally identified by any context they interact with to get the right services or information. A full list of these challenges is summarised in Appendix B.

Achieving such a smart environment requires dramatic improvements in the existing systems, architectures, and communications protocols [13]. They should be more flexible, adaptive, collaborative, and pervasive, which will attract the research community and companies to realise this goal. Additionally, the IoT service applications should be expanded horizontally to overcome the limitation of current vertical services domains.

1.2.Actors in the IoT

1.2.1. Definitions

The terms *entity, object, thing* and *actor* have been utilised in the IoT literature with different meanings. Their meaning is often mixed up and confused by the reader. Therefore, they are defined based on the IoT's three billers in Section 1.1, which are depicted in Figure 1-1, as follows:

- ***Entity*:** A general term used to describe any identified component in the IoT environment, which has an identity and a set of attributes that describe it. Entities represent a person, a car, a place, an organisation, an application or more that tend to communicate with other entities to send or receive information or control messages.

- ***Object*****:** Any entity that embeds (or attached to) a communication device. The communication device allows entities to communicate with each other and before accessing the Internet. It may use various communication technologies such as Radio Frequency (RF), Near Field Communication (NFC), BlueTooth BT, Wireless Fidelity (WiFi), etc. A person who interacts with a wearable Fitbit or a PC that is not connected to the Internet are examples of the IoT's object.
- ***Thing*****:** An object, which has Internet connectivity. Therefore, the object becomes an active participant in the information network, i.e. a thing, as it is accessible by the Internet and able to share its data with interested parties. The terms "smart object" and "smart thing" are denoting to the same meaning of "thing" [13], [17].
- ***Actor*****:** Represents any entity, object or thing from the IoT environment that interacts with each other to communicate with a (possibly remote) real other object or thing to achieve a goal. The goal could be to monitor, move, manipulate that object, or set/get some interesting information [18], [19].

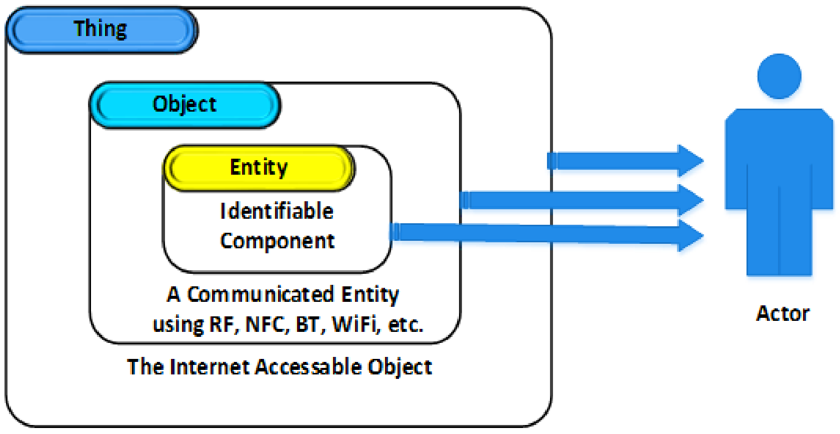

Thing
Object
Entity
Identifiable Component
A Communicated Entity using RF, NFC, BT, WiFi, etc.
The Internet Accessable Object
Actor

Figure 1-1: Entity, Object, Thing, and Actor Demonstration

From the above definitions, all "things" in the IoT are instances of 'entity', but not all entities can become things. For instance, a car driver considers as an entity, but when he/she combined with an active RFID tag, embeds in the car, to interact with the electronic toll system to pay a charge, he/she becomes a thing in the IoT.

1.2.2. Actor's Interaction

From a technical point of view, IoT implies an enormous number of interacted devices (or smart objects) on behalf of other IoT entities, i.e. the effective actors (EA), to perform a task. These communicated devices exist in different application areas like smart building/cities, healthcare, environment monitoring, inventories and more. They form the smart environment that links the EA, the actual service requester, with service providers (SP), which are any entity could offer a service like sensors. In other words, the EA is any physical/logical entity from the IoT environment, such as a person or an application, with a goal of consuming/accessing a service or data offered by other IoT's actor. Figure 1-2 depicts the actor's interaction in the IoT.

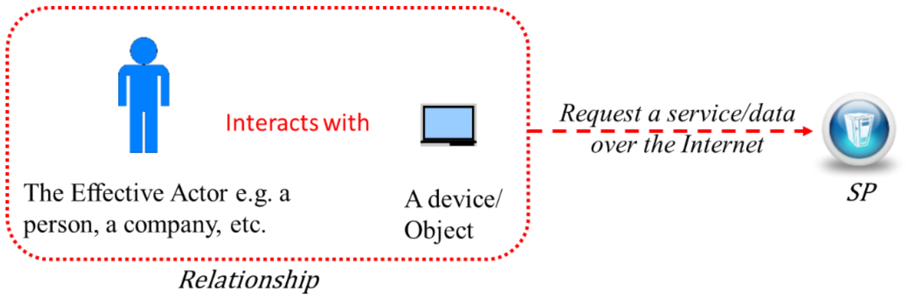

Interacts with
The Effective Actor e.g. a person, a company, etc.
A device/ Object
Relationship
Request a service/data over the Internet
SP

Figure 1-2: Actor's Interaction Demonstration in the IoT

The roles of communication devices/objects is to interact with the EA and with themselves to offer a smart service to the EA from a SP or more. For instance, a RFID reader interactes with a company inventory system to update the statuse of the company assets; an insurance company uses telematics devices to monitor young drivers' behaviour. These interactions can be generally categorized into the two forms as below:

- The interaction between the physical communication devices/objects using the existing communication technologies.
- The interaction between the requester and the communication device/object based on a relationship;

Although the research community understands the interaction between the variety of communication objects well, it inherits the difficulties from the conventional communication networks and brings them to the IoT. The second interaction requires more attention and consideration to realise the IoT environment. In the IoT, the objects are ubiquitously available and offer their service to any interested requester. The requesters could temporarily add such objects to form a virtual space or share their objects with others. This interaction between the requester, i.e. the EA, and the IoT objects requires at least a relationship between two entities. These relationships might not always be static in nature [20]; they could be dynamically established and after a period will change or even vanish.

To understand the actors' interaction, let us consider the following scenario from the actor's interaction point of view. A hospital wheelchair, which has a unique identifier to distinguish it from others is an entity in the IoT. To allow this wheelchair to become part of the IoT as a thing, it requires having Internet connectivity. By attaching a communication device to the wheelchair, it will be able to communicate within its area using a suitable technology. In the case of using a technology that does not have *Internet Connectivity* (*i.e. stack of IP*) such as BT, it is still able to communicate within its domain. In such a case, it will be denoted as an "object". An additional device is used to connect the wheelchair (as an object) with the Internet, which act as an Internet gateway. Next, this object (i.e. the wheelchair with the communication device) has to be accessible by the Internet to call it a "thing" in IoT. By linking it to a patient's smartphone, the wheelchair becomes a "thing" in the IoT and can now send data through the information network.

From this scenario, two actor relationships are recognised: the first relationship is between the wheelchair and the communication device, while the second one is between the communication device and the smartphone being used to access the Internet. These relationships represent interactions between different actors and aim to allow the entity to become a thing in the IoT. Therefore, the wheelchair, communication device, and the smartphone, as an Internet gateway, are represented actors in IoT that have different relationships with each other.

1.3.IoT Enabling Technologies

A rapid development of technologies such as communication technologies, sensors, smartphones, cloud computing, and network virtualisation will enable objects regardless of any limitation in time or location. The simple concept of IoT is to allow any objects in the real world to communicate and collaborate with the digital world. The main technologies which support that concept are:

1. ***Identification technologies:*** Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSN) has a significant role to identify objects in the IoT. RFID tag is small in size, low in costs, and uses RF technology to identify things for a long time uniquely. RFID becomes a crucial factor to realise the IoT because of its long distance reading, re-

written data capability, secure, and fire-resistant. Currently, they are widely used to build the IoT applications. Similarly, Near Field Communication (NFC) is a short-range wireless technology using RFID technology that enables comfortable and convenient communication between devices [21], [22]. The integration of sensors with RFID pushes the IoT towards the implementation of industrial services. Combining IoT with RFID and WSNs facilitate applications development for healthcare, decision-making of complex systems and smart systems as smart transportation or smart rehabilitation systems [23].

2. ***Networks and communications technologies:*** Currently many cross-layer protocols exist for Wireless Networks, Wireless Mesh Networks (WMN), or Ad-Hoc Networks. Protocols convergence is a crucial characteristic to realise IoT due to several reasons. Firstly, things have a wide range of hardware configurations, QoS requirements, functions, and goals. However, nodes in WSN are often characterised as a homogeneity in specifications, communication requirements, and goal. Secondly, IoT relies on the Internet; hence a centralised and hierarchical architecture is inherent, whereas WSN, WMN, and Ad-Hoc are nearly flat in nature. Because devices are diverse in terms of communication, computation, memory, and data storage capabilities; gateways have a key role in organising devices for the communication purposes via the Internet. Since the IoT can be a combination of heterogeneous networks such as WSN, WMN mobile networks and WLAN, the gateway can support the things to make decisions, computations and share data. The existing network protocol like IPv6 can also benefit the IoT.
3. ***Existing Internet technologies and devices*** are considered part of the IoT as smartphones, tablets, laptops, industrial technologies, appliance, and building automation.

1.4.Identity Management Systems (IdM)

IdM defined by ITU-T in [24] as "*A set of functions and capabilities (e.g. administration, management and maintenance, discovery, communication exchanges, correlation and binding, policy enforcement, authentication and assertions) used for:*

- *Assurance of identity information (e.g. identifiers, credentials, attributes);*
- *Assurance of the identity of an entity (e.g. users/subscribers, groups, user devices, organisations, network and service providers, network elements and objects, and virtual objects); and*

- *Enabling business and security applications".*

The IdM framework [25] is composed of the following entities that are depicted in Figure 1-3.

- ***User*** is an entity that intends to access some service or resource;
- ***Identity provider*** ***(IdP)*** is responsible for managing users' identity attributes and delivering it to the service providers;
- ***The service provider (SP)*** is responsible for delivering the requested resource/service to a requester. It relies on the $IdP(s)$ to perform the user identity verification and is usually responsible for the authorisation process.

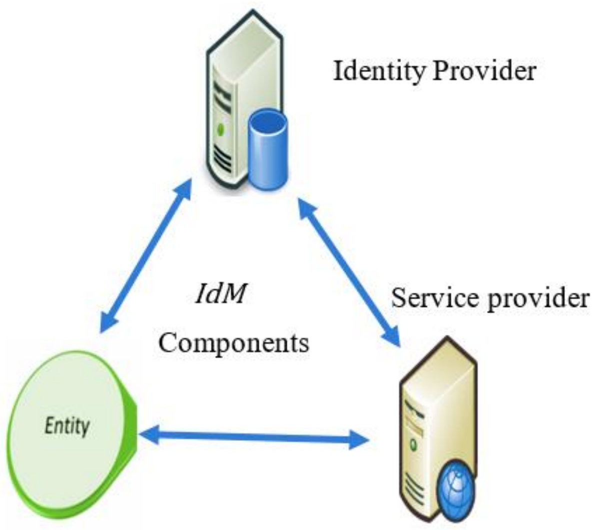


Figure 1-3: The IdM framework

To manage and control dealings with the users, every SP has a legal and permanent contract with an IdP that form a IdM [26]. The role of any IdM system is managing the users identities and their authentication and authorization to serve a single SP or multi SPs through a pre-established Circle of Trust (CoT) [27]. IdM aims to assure that the SP will offer its services to a trusted user (client) based on a pre-established trust relationship with the IdP to increase the enterprise's security and efficiency. The identity is used to distinguish the entity within a particular SP based on its characteristics. However, entities could have several different identity attributes within the same domain or distributed over different unrelated domains [28]–[30]. Practically, entities own diverse identity attributes are distributed over unrelated IdM domains, which are valid and used within that IdM domain for different purposes [31], [32]. Moreover, these systems, i.e. $IdMs$, are not always interoperable/ compatible with each other. This is because they often use varying types of identity attributes, different methods and protocols, and different access control policies.

1.5.Research Motivation

The Internet of Things (IoT) is composed of a vast number of heterogeneous entities, such as people, sensors, places, applications, smart devices, and more. The population of these connected entities is expected by Cisco to be 50 billion in 2020 [33]. These entities have to interconnect and collaborate over the Internet Protocol. Entities, in the IoT vision, should be uniquely identified, located and managed without human intervention. They vary in their computing and communication capabilities. Some of them are communication objects that are capable of directly connecting with others anytime, anywhere, and by anyone through any path/network, while others rely on these communication objects to achieve their goals. Each entity, in IoT, can have one or more identity attributes (e.g. identifier) which are managed by an IdM in every domain they interconnect with. Usually, $IdMs$ use different types of identity certificates and identification methods, which are not always compatible or recognizable by others. Therefore, relying solely on such attributes may not truly represent the actual user across multiple-domains, which is key to the success of the IoT.

Kantara Initiative [34] in its discussion group "IDentities of Things" (IDoT) stated that *"Apart from adapting communication protocols, an overarching identity framework is crucial for a growing IoT. Today we have many separated solutions and niche standards. As a consequence, there is no overall framework for how to recognize and manage identities across different solutions".* That means there is a difficulty to recognise and truly identify the entities across their IdM domains because of the lack of $IdMs$ interoperability [22], [35]–[37]. Such interoperability problems result from the lack of semantic interoperability[38], syntactic interoperability[39], and across-domain systems interoperability [40]. Improving the $IdMs$ collaboration to globally establish the entities identity is crucial in offering its services and in supporting the entities mobility in the IoT environment.

In IoT, communicated object(s) are often related to an end user entity, e.g. a real person or an enterprise. Theoretically, the relationships could be defined among entities (such as people and their related devices or things), between different communicated devices, between entities and applications and services, or between devices and applications and services. A person often uses many objects on a typical day; some of them are owned by that person (or their family), while others are not. One can think of possible scenarios of how to interact with open-access devices (or communication objects in general) to request services or data. Furthermore, according to Gartner in [41], "*The Identity of Things requires a new taxonomy for the participants in Identity and Access Management systems. People, software that makes up systems, applications and services, and devices will all be defined as entities and all entities will have the same requirements to interact*". The IoT will change the current ways of interaction with entities from "owner" and "subscriber" into much broader ways such as interact with open-access objects as discussed in [41]–[43].

The authors of [44] stated that "*At any time t, every IoT object should have an owner, but might have one or more users. The relationship among the IoT object, owner, and users might also change with respect to time in its lifecycle.*" Those objects' identifiers are used to verify their user's identity or as an alternative identity. However, the relationship between a user and its related object(s) is changeable [45], which may cause identity fraud or misidentification because the identification data is becoming out-dated. Current communications of IoT objects lack the means to identify the relationship between the effective actor, as an actual actor behind the communication and the communication object. Furthermore, Gartner [41], said: "*the concept of dynamic relationships is vital to the success of future IdM solutions*". Such relationships have a direct impact on the other identity-related processes like authentication, authorisation, and data access governance [34], [46], [47]. Therefore, defining such relationships has a significant impact on truly identifying the actual actor of the communicated object. This is because there are many to many (m:n) interactions between devices in the IoT environment [42].

Finally, there is a mismatching of aims between the IdM system and IoT. The role of any IdM is managing the users (clients) identities and authentication, authorizing them within the enterprise domain or within its federated domains through a pre-established Circle of Trust (CoT). An IdM system aims to limit the services offered by a SP to trusted users (clients) based on a pre-established trust relationship with the identity provider (IdP). However, the IoT aims to offer the services by a SP to any requester without such domain limitation. Traditional IdM systems focus on managing the real user identity and neglect to consider other end users, such as applications, places, and things. They, in addition, are static IdM in nature because they are unable to dynamically establish a trust relationship with a foreign component [41]–[43], [48], [49]. To solve the above limitations, the researcher has been motivated to do this research.

1.6.Research Problem

The problem targeted by this research is the difficulty of establishing the effective actor (EA) identity behind any communicated object by any visited SP across its home IdM in the IoT environment. Figure 1-4 illustrates that the EA can interacts with multiple communication objects based on an actor relationship to request services or data from SPs in the IoT using the owned identity. However, where each actor could have multiple identifiers supplied by different IdM domains. The SPs should be able to identify the EA to offer him true services or data. This difficulty results from the following:

- Lack of a common means for identifying the EA relationship types between the communication objects or devices.
- Lack of flexibility to establish trust relationship dynamically between unrelated SP and $IdPs$.
- Lack of an $IdMs$ interoperability to globally establish the identity in IoT.
- Lack of a standard identity verification system to establish the EA identity through any of these relationship types.

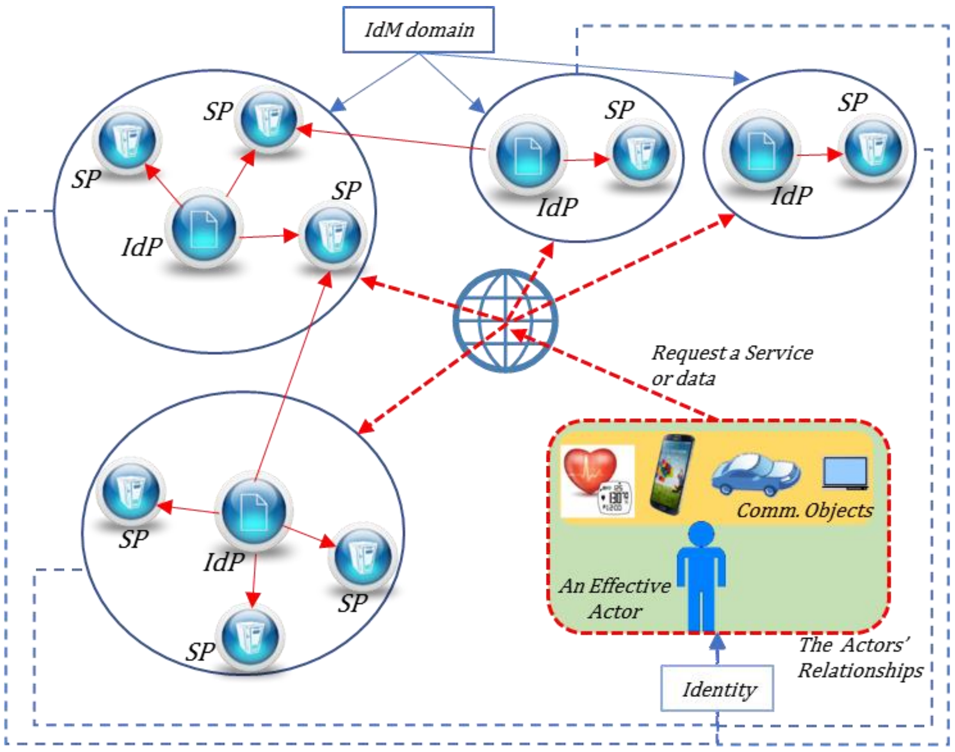


Figure 1-4: The High-Level of the Research Problem

1.7.Research Aim and Objectives

This research aims to propose a new identification technique to establish the EA identity behind any communicated object(s) by Service Providers based on their relationships with other objects in the IoT. To achieve this aim, the objectives of this research are:

1. To determine the requirements to establish the EA identity by SP in the IoT environment.
2. To formulate a semantic identifier to represent the actor relationship in a mathematical model for the IoT.
3. To propose a new IdM system and a protocol to facilitate establishing a dynamic trust relationship between entities (SPs and $IdPs$) and to establish the EA identity by any SP in the IoT.

1.8.Research Process

The scientific research methodology will be used in this research. The research process contains the following steps, which are depicted in Figure 1-5.

1. ***Review the actors' identification techniques in the IoT:***

   At this stage, all the effort has concentrated on investigating previous research about entities' identification in the IoT. This stage shows that different identification schemes are used to identify objects. These schemes are mainly categorised into three types: object identification (e.g. EPC), communication identification (e.g. IP-address), and an application identifier (e.g. URL). Moreover, each real person can have relationships with different objects, which can be used to identify the person in addition to his/her identity attributes. Furthermore, several IdM systems have been developed to cope with the future Internet challenges, while few of them have targeted the IoT environment.

2. ***Define & characterise the research problem:***

   Based on the literature, actors could have different identity attributes and preferences scattered across different administrative domain. Each domain uses a different technique to identify the actor using them. The EA can have many relationships with communication objects, which can be used to identify the actor in different contexts. Due to the mobility characteristic and lack of interoperability between independent $IdMs$, these techniques do not support establishing the EA identity across-domain in the IoT environment. This will be a serious challenge for the IoT goals of offering a better service to the human being. This is because wrongly identifying the EA leads to offering the wrong services.

3. Determine the identity establishment requirements in the IoT

In this step, a scenario driven approach is used for eliciting the requirements. The analysis of the typical IoT scenarios in the literature review where the actor uses various objects to request services across its IdM domain will lead to: (1) formulate a general actors' interaction use-case for requesting service/data in the IoT environment; (2) elicit the requirements to uniquely establish the EA identity behind the communicated object(s) that requested the service.

4. Evaluate the compatibility of the existed $IdMs$ with the requirement & propose a new compatible one:

Based on the requirements that captured from the previous stage, a new identifier is proposed. This identifier will contain the necessary information to identify the EA at any domain in the IoT environment. The identifier will explicitly represent the actors relationship attributes in addition to the actors attributes. Moreover, a new IdM system architecture and a protocol has been proposed to identify the EA. This IdM will improve the interoperability between the unknown entities of the $IdMs$. Finally, the existing network protocol (IPv6) will be used to develop the proposed protocol messages. IPv6 is considered the most suitable protocol for the IoT as stated by the IoT6 research group (http://iot6.eu) and many other researchers such as in [50].

5. *Test and evaluate the proposed IdM*

In order to evaluate the effectiveness of the proposed IdM to establish the EA identity in the IoT environment, the following process steps have been followed:

- The conceptual validation has formally proved the correctness of the proposed protocol logic to establish the mutual trust relationship and verify the EA identity using Burrows-Abadi-Needham (BAN) logic of authentication.
- The formal verification using the ProVerif tool has proved the authenticity criterion of the proposed protocol to establish the EA identity and its security against the simulated attacks.
- The state of the art works have been compared with the developed identifier and system using the elicited requirements as an evaluation factors. The comparisons prove that the proposed identifier and system are fully satisfied the requirements to establish the EA identity in the IoT while the others are partially.
- A testing environment has been built to evaluate the ability of the proposed identifier and system in the IoT environment. The basis testing scenarios of actor's interaction

proves the effectiveness of the proposed solution to identify the actual actor across-domain.

- The protocol overhead are analysed in terms of the communicational and computational costs. It shows that the overhead is in a direct proportional relation with the number of identity verification messages and the number of foreign $IdP(s)$ that are involved in establishing the EA identity.

6. Documenting the results

The last part of the research is documenting and publishing the results. Two conference papers have been published until now while publishing the other are planned shortly. However, they are documented in this thesis.

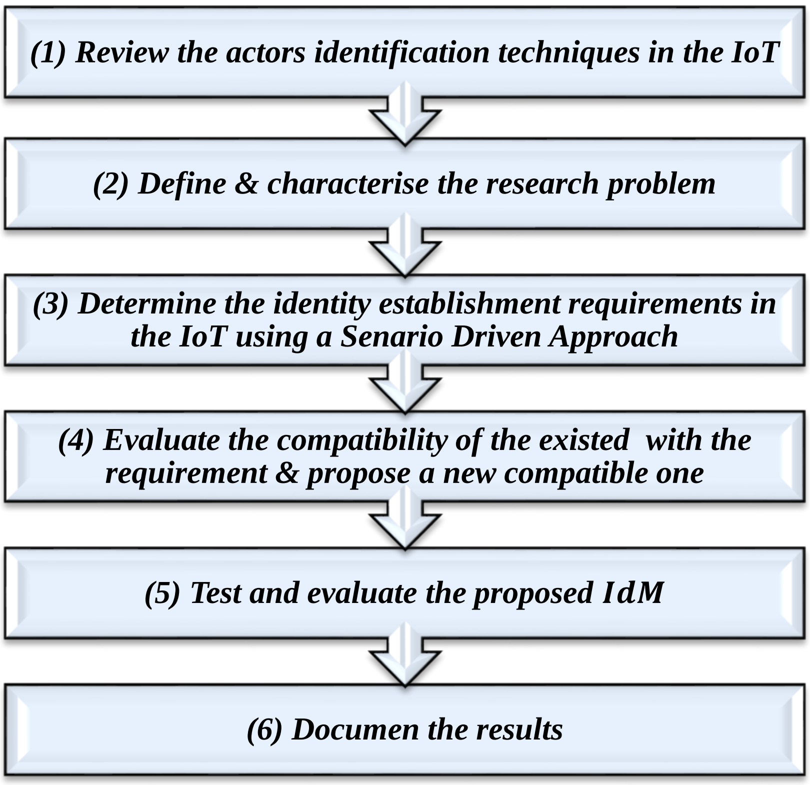

(1) Review the actors identification techniques in the IoT
(2) Define & characterise the research problem
(3) Determine the identity establishment requirements in the IoT using a Senario Driven Approach
(4) Evaluate the compatibility of the existed with the requirement & propose a new compatible one
(5) Test and evaluate the proposed IdM
(6) Documen the results

Figure 1-5: The Research Process Steps

1.9.Research Contribution

This research develops a new identification technique for the IoT environment which implies the development of the global actor relationship identifier $(GARI)$ and the global identity management $(GIdM)$ system to establish the actual requester's (EA) identity in the IoT environment. Four identity parameters have been proposed to use to identify an entity in the

IoT, which are the entity's *identifier*, *type*, *Internet connectivity type, and the identity provider's identifier*. The Internet connectivity of the communication object leads to identify whether a single actor relationship is used by the EA to access the Internet or more relationships are needed, if the communication object is of a passive object type. Moreover, the actor relationship types will be represented in the $GARI$ in addition to the identity attributes of each entity participating in the relationship. The attributes (the Internet connectivity and the actor relationship) have never being used before to identify the entities. Furthermore, the identity establishment process of the $GIdM$ relies on these actor relationship types and the entities attributes to facilitate the establishment of the actual requester (EA) identity by the service providers at any visited domain in the IoT environment.

1.10. Thesis layout

Chapter 2 presents a brief background about the identity and the identity management systems. The existing identification schemes and $IdMs$ are reviewed and evaluated against this research problem. The way forward is drawn at the end of the chapter.

Chapter 3 presents the general use-case for actors' interaction in the IoT environment based on the analyses of the literature. Moreover, the general requirement to establish the EA identity are explained. The actor relationships are defined and represented by a novel mathematical model. The novel global actor relationship identifier is formulated and represented by an example in the chapter.

Chapter 4 presents the architecture of the proposed system, i.e. the global identity management ($GIdM$), with a detailed description of its entities. The proposed protocol messages to establish the EA identity in the $GIdM$ are presented.

Chapter 5 presents the testing environment configuration using the simulation environment NS3. The testing scenarios to test the $GIdM$ is explained. Finally, the testing environment is validated using a scenario.

Chapter 6 explains the evaluation process of $GIdM$. It implies the formal/ informal validation, formal verification, critical evaluation, testing, and analyses the overhead.

Finally, chapter 7 concludes the work and presents the future works.

Chapter 2.

Identity & Identity Management: Background & Literature Review

The concepts for identity and identity management system are not new. They have widely interested people from industrial and academia sectors. However, by launching new information and communication technologies (ICT) like the Internet of Things, they become more important to consider when making effective solutions as it will allows to recognize the user's identity in the digital world to offer a right service. This chapter aims to present a brief background for these concepts. Moreover, the existing solutions in the literature will be reviewed and analysed to check their ability to solve the research problem in the IoT environment.

2.1.Identity

The term "Identity" is used to represent an entity in a particular domain. An entity is something that can be uniquely identified (e.g. a person, device, an organisation) depending on a set of attributes in a specific situation. Every entity has a "whole" identity that encompasses all its distinctive attributes. A subset of these attributes can form different "partial identities" in different domains [51], [52]. In the digital world, the identity is conceived as a "digital identity" which is defined in [53] as "a digital representation of a set of claims made by one party about itself or another data subject".

Identity attributes are categorised into three groups, called "tiers of identity" [54], [55]. Tier1, marked "My Identity", includes attributes and traits derived from the entity such as special interests and favourite activities. "Shared Identity" is the mark of tier2, which contains the attributes that are assigned to the entity by others to identify him temporarily within a specific domain such as driver's license and credit card. However, Tier3 deals with "Abstracted Identity", which is used to establish a group identity such as marketing emails or spam.

According to the ITU-T recommendation, (Y.2720) [24], an identity is represented by three different types of data: identifier, credentials, and attributes.

- *Identifiers*: "A series of digits, characters, and symbols or any other form of data used to identify a subject. Some examples are user account names, passport numbers, mobile phone numbers, employee numbers, and URI."
- *Credentials:* "A set of data providing evidence for claims about parts of our entire identities. A credential can be generated based on one or more credentials. Some examples are passwords, digital certificates, fingerprints, Kerberos tickets, and SAML assertions."
- *Attributes*: "A set of data that describes the characteristics of a subject. The data includes the fundamental information for identifying a subject, his/her preferences, and the information generated because of his/her activities. Some examples are given: family names, domiciles, ages, genders, roles, titles, affiliations, activity records, and reputations."

In the literature, there is no single definition of the "identity". For example, the authors of [56] consider identity as a tool used by an entity to deliver information itself to the system. Pfitzmann and Hansen [57] defined identity as: *"An identity of an individual person may comprise many partial identities of which each represents the person in a specific context or role. A partial identity is a subset of attribute values of a complete identity, where a complete identity is the union of all attributes values of all identities of this person."* Their definition considers only a person as a subject of identity. However, the authors Bishop, cited in [58], [59], and [60], explained that identity implies a wide range of subjects, not just people. *"Subjects of identities can be software agents (e.g., Web services and user client software) and hardware devices (e.g., PCs, mobile phones, and network equipment)"*. Similarly, ITU in its X.1250 recommendation [61] defined identity as "*the representation of an entity (or group of entities) in the form of one or more information elements which allow the entity(s) to be uniquely recognised within a context to the extent that is necessary (for the relevant applications)*". It stated clearly that an "entity" can be "*a physical person, an animal, a juridical person, an organization, an active or passive thing, a device, software application, service, etc., or a group* *of these entities. Moreover, entities include access points, subscribers, users, network elements, networks, software applications, services and devices, interfaces, etc. in telecommunication context."* The ITU's definition is the most acceptable one in the IoT environment as it complies with the IoT pillars, stated in Section 1.1, and involved a wide range of entities.

2.2. Common Identity Establishment Factors

Identity establishment describes the process of verifying a users' identity based on some identification factors. The most common factors are often described as *"something you know (the knowledge factor), something you have (the possession factor) and something you are (the inherence factor)"* [62]. In addition to these factors, the context-aware factor is considered as a fourth identification factor [63]. These factors are applied to both objects and users but with different complexity.

2.2.1. User Identification Factors:

They are briefly outlined as below:

1. ***Knowledge factor:*** Information that a user must be able to provide in order to log in. Usernames or IDs, passwords, PINs and the answers to secret questions all fall into this category.
2. ***Possession factor:*** Anything a user must have in their possession in order to log in, such as a security token, a one-time password (OTP) token, a key fob, an employee ID card or a phone's SIM card number.
3. ***Inherence factor:*** Any biological traits the user has that are confirmed for login. This category includes the scope of biometric authentication methods such as retina scans, iris scans, fingerprint scans, finger vein scans, facial recognition, voice recognition, hand geometry, and even earlobe geometry.
4. ***Context-aware factor****:* the user's current location and time are often suggested as a fourth factor. For instance, verifying the user location using GPS devices combined with the time, enabling reasonable confirmation of the login location at a specific time.

2.2.2. Devices Identification Factors

Device identification is similar to user identification but less complicated using two factors [64] as follows:

1. ***Possession factor***. A device identity will be proved using secrets stored in the device, which are used to enable user identification as described above. A device may use a secret to perceive another type of information that is recognised as a reliable proof by

the authenticator. This secret, e.g. X.509 certificate, is effectively issued for user authentication and often used automatically to login or periodic presence verification without ensuring user presence at the exact time. Thus, secrets stored in a device should be treated as a means to authenticate devices and not users.

2. ***Context-aware factor***. A device identity will be determined using its behaviour or its characteristic such as geographic or frequency of transmitted signals. However, credentials that are considered context-based rather than identity-based are not suitable for detection of the device identity.

2.3. The state of the Art of Identifiers in the IoT

The IoT refers to the use of Internet technologies to interconnect uniquely identifiable objects. Furthermore, IoT applications rely on the interoperation of information and services that are offered by various Internet-connected objects. The objects include both physical devices (e.g. sensors) and virtual/logical entities (e.g. applications). Therefore, utilising effective identification technologies and techniques by these objects to identify the actual requester are the key to achieve successful IoT applications and services.

2.3.1. A Taxonomy of Identification Schemes in IoT

Identification schemes are generally categorised by CATR & IERC [65] according to its purpose, into three categories:

- ***Object Identification scheme***, which is utilized to uniquely identify physical or virtual objects, such as RFID OID (Object Identifier), EPCglobal (Electronic Product Codes), Handle/DOI (Digital Object Identifiers), UUID (Universally Unique Identifiers), MAC, and more.
- ***Communication Identification scheme***, which is utilized to uniquely identify objects, such as sensors and devices, when communicating with other objects. For examples, IP addresses (e.g., IPv4 or IPv6) and E.164 addresses.
- ***Application Identification scheme***, which is utilized to uniquely identify applications and services used in the IoT applications. A host name, URLs (implies URIs and URNs) can be used to represent Application-level identification.

Table 2-1: Features and Limitations of Identification Schemes [66]

Identification scheme	Features	Limitations
RFID OID	• Code structure can support legacy systems	• Various OID structure • No generic resolver system • Lack of an ISO standard budget • Serve in centralized system
EPCglobal	• Following code structures to unique objects identification • Base of all GS1[1]barcode • End-to-end code for service system	• Limited use within GS1 domain • At thing level, limited and lack of proof data carrier options • Increased cost of using the system by fewer retailers • Privacy issues
Short-OID	• A special type of RFID OID • Support OID encoding • Shows carrier data	• No proper resolver system • Lack of items differentiation because of using common root • Similarity with RFID OID
Near Field Communication Forum	• Significant technology integrated with smartphones	• Based on specific air protocol • Low integration of data captured with other tags • Similarity with 2D barcodes
Handle and DOI	• Identifying e-resources • Support increasing number of domains	• Infrastructure overload for additional application • Lack of carrier data • Inapplicable to physical objects
Ubiquitous Code	• Implemented in Japan • Resolve system using TRON engine	• Linking different Ucodes through a Relational DB is required to get a specific information • Significant differences with EPCglobal and ISO RFID in terms of item information.
URL as an Identifier	• Browser based identity established • Short form existed like Tiny/URL	• DNS is required to access the page, unless using "absolute URL" • Long in length • Security issues • Not powerful for data capture
IP Address as an Identifier	• M2M communication • Remote monitoring • Valid for most communication devices	• Not support all IoT entities • Not suitable for tiny objects • Scalability issue

Additional information might be added to the identification scheme to explain the thing relationship with other objects (e.g., server hostnames are associated with the NICs that they

[1] "GS1 is a not-for-profit organisation that develops and maintains global standards for business communication."

comprise) and/or their locations. Table 2-1. summaries the IoT's identification schemes features and limitations. Many researchers have concluded that it is nearly impossible to have a single identification scheme for all the objects in the world. The researchers in [22], [66], [67] consider this conclusion to be a result of the lack of a required infrastructure to support objects mobility and the difficulty of managing and introducing considerable technical costs. The EU-China IoT Advisory Group, in [65], and Jung et al., in [68], consider the interoperability and internetworking across different domains to be the cause of that. Thus, they are more likely to coexist together in the IoT. However, this coexistence of diverse identification schemes will raise interoperability issues for truly identifying entities in such heterogeneous communities of entities that needs to be solved by finding an effective way to sort them.

2.3.2. Proposed Identifiers by Research Projects in the IoT

There are several proposals to develop an identifier to use in the IoT environment by a research community. These can be summarised as follows.

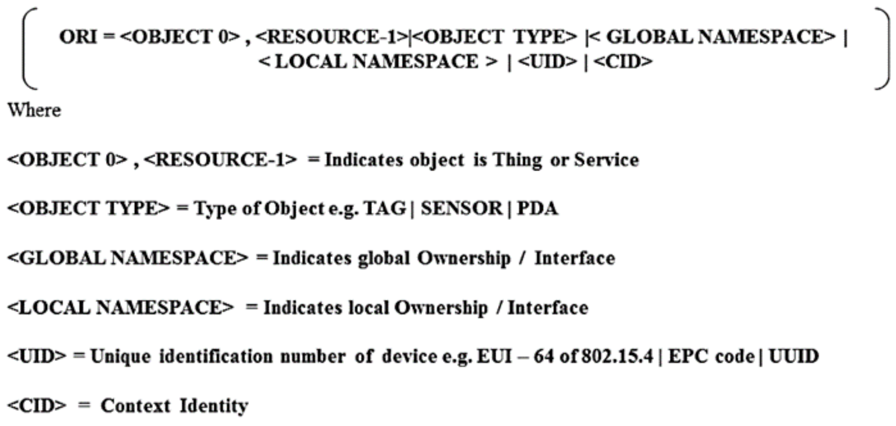
Liu, et al. in [69] proposed an identifier format used to control the sensor nodes remotely in the IoT. Their identifier was composed of a domain identifier, device type and the device identifier using a URL style using 64-bits to formulate their identifier using the format

"dev: //domain-series/devtype/legacy-name"

They focused on object identification without considering the owner (or user) identity of that device nor its relationship with an enterprise (or a real person).

Batalla & Krawiec, in [70], proposed an object identifier, which was composed of a chain of all the names, separated by a dot starting from the root; but again it lacked a mention of the users. This identifier was proposed for sensory environments and focused on controlling fixed devices remotely such as controlling a smart home appliances. For example, to communicate with a light on in the first room, a control message could be send using the format (**.floor001.room0001.lightctr**) followed by the control command.

Mahalle et al., [43], [71] stated that an entity's identification can be defined by using a collection of three parameters which are: *type, identifier,* and *namespace* in which that identifier is assigned to the entity. However, the proposal ignores an important parameter which is the Internet connectivity characteristic of the entity. This is because they limit their work to entities with computing capabilities. That means their identification ignores a large community of tiny and low capability objects, which fill the IoT environment. Accordingly, they proposed an identifier format for objects and resources in IoT, which is composed of a set of permanent or temporary attributes that represent each end-point identification. Object mobility was considered through using a global namespace and local namespace parameters.



ORI = <OBJECT 0> , <RESOURCE-1>|<OBJECT TYPE> |< GLOBAL NAMESPACE> | < LOCAL NAMESPACE > | <UID> | <CID>

Where

<OBJECT 0> , <RESOURCE-1> = Indicates object is Thing or Service

<OBJECT TYPE> = Type of Object e.g. TAG | SENSOR | PDA

<GLOBAL NAMESPACE> = Indicates global Ownership / Interface

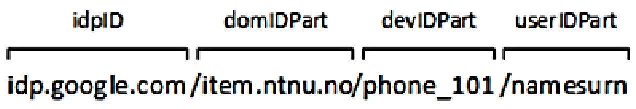
<LOCAL NAMESPACE> = Indicates local Ownership / Interface

<UID> = Unique identification number of device e.g. EUI – 64 of 802.15.4 | EPC code | UUID

<CID> = Context Identity

However, user representation is missing again and in turn the relationship between the user and the object is also missing. The research is limited to the Internet protocol (IP) connected devices without considering other communication technologies that use intermediary devices to connect to the Internet.

Butkus, in [7], [64], proposed an identifier format composed of a set of identities based on a URL format. It contained IdP identifier, domain identifier, device identifier, and a user identifier as follows:



This identifier is used to identify the owner of the devices, and the researchers assumed that both were registered within the same IdP. However, they consider only a fixed relationship type between the user and his/her personal communication. Moreover, they only considered devices with computing resources and neglected other devices with low computing capabilities. Again, the research was limited to devices connected with the Internet Protocol and ignored other communication technologies.

Zdravkova [72] proposed an identifier format for the IoT, which was composed of the following parameters: device type, global ownersip/interface, domain identifier, user identifier, and a device identifier as follows:

“dtype|gloInt|unidomID|unidevID|uniuID”.

The identifier used a device type to specify the type of entity that is identified by this identifier; this entity could be a person or device. However, the relationship between user and device was missing again. The domain identifier was used for both the user and the device without considering that they could be different.

Almost all of the above proposals encompass the *communication object identifier* and the IdP (or *domain namespace*) identifier as identity attributes, whereas some of them use a *user identifier*. Moreover, all proposals lack any information related to the *user type* of the communication object. In addition, none has considered the *user-device relationships* and its change impact on truly identifying the actual requester of a service or data by the service providers. Similarly, all proposals ignored the *Internet Connectivity* characteristic of the entities, assuming all devices able to access the Internet. Hence, a relationship between a tiny object and a gateway (or an intermediary device) is also missing. Therefore, it can conclude that the existing identifiers are insufficient to identify the actual requester, i.e. EA, based on a relationship with a communication object in the IoT.

To sum up, a new identifier is required to represent the actual requester actor properly to any service provider in the IoT environment. It has to achieve two goals: firstly, to identify the effective actor that initiated the communication (e.g. a person) which may not be the entity that is connected to the Internet, and secondly to allow dynamic relationships between such entities over the IoT due to the nomadic nature of the IoT entities that can freely join and leave different *SPs* to get their services.

2.4.Identity Management Systems Models

IdM can be classified into three main models. These models represent the relationships between an identity provider (IdP) and services provider (SP) and the mechanism used to manage the identities of a system [73]–[75].

1. ***Isolated IdM-model*:** In such a model, there is no co-operation between parties to support user authentication. The SP trusts only itself, and also plays the role of the IdP. This model is usually used in online services and resources [25]. The disadvantage of this model is inadequate usability since it causes identity overload and password fatigue for users with many different SPs.

2. ***Centralized IdM-model***: In this model, a single IdP will be used to provide identity services to the participating SPs within a closed domain. However, this model is not suitable to implement in an open environment where SPs are not governed by a common policy and authority.

3. ***Federated IdM-model***: In this model, a frictionless IdM solution is presented by forming a federation and making authentication a distributed task. Each party within a given group trusts some or all of the parties within this group. This means that every party within a group agrees to trust user identities assigned by other members of the group. The main disadvantage is that it creates legal and technical complexity.

In addition to the above models, a dynamic federated model is an evolved model of the federated IdM. It is a business model that aims to allow SPs and $IdPs$ that are unknown to create a federation without a previous agreement between parties [76]. The main issue apposing this model is building the trust between parties on the fly [77]–[81]. The absence of trust is a serious problem because an entity could join a federation without the existence of any previous agreement. The works [77]–[80] discussed the use of the security assertion markup language (SAML) to build the federation. However, SAML does not support creating a federation dynamically, although some changes proposed by the researcher in [80]. OpenId connect is proposed to create a dynamic federation by [81]. All these works assume that SPs and $IdPs$ follow a single standard in their applications.

2.5.Identity Management Models in the Internet of things

In the IoT environment, there are three main IdM models to manage users and devices identities. These paradigms are classified by Mahalle and Railkar in [43] as follows:

1. ***User-Centric IdM:*** This model allows the end-users to control their own digital identities. The users are in the middle of transactions between $IdPs$ and SPs. Such a model is effective to manage the identities across the IdM administrative domain, as the user is in charge of managing the identity information. In such cases, a global identifier or arrangement to support the interoperability is required.

2. ***Device-Centric IdM:*** The concept of device/things identities is still not widely used. It is mainly used for users authentication and to identify things in repositories (e.g. RFID tags, MAC-address, etc.). The user is required to first be identified by the device (e.g.

mobile device or PC), which in turn used its identity to request a service on the user's behalf. In the future, the interaction between users and things in the surrounding environment will be different and this requires new things identities. Can one imagine how a person can use open-access devices for a temporary period? How can the person trust this device? How can the device access his/her personal details? The answer to these questions can be found in the identities proposed by the device.

3. ***Hybrid IdM***: This model considers dealing with both user and device identities concurrently. In the cloud computing area, both user and devices/services identities are required by the cloud IdM. The vital point of managing hybrid identities in federated IoT is the delegation of $IdPs$ for those identities.

2.6.State of the Art of IdMs in the IoT

Managing the identities of various actors in the IoT is still a widespread difficulty faced by IoT applications today. This Section explores the current standards and research efforts to manage the identities in the IoT.

2.6.1. IdM Initiatives

In literature, there are several IdM systems developed to cope with the future Internet challenges. They follow different architectures and standards to give the intended solutions. However, they need to be harmonized to face the interoperability challenges in the IoT environment [59]. Current $IdMs$ are categorised in [56] into three main groups ***Projects and Architectures,*** e.g. (PRIME, DAIDALOS and SWIFT); ***Alliances***, that represent a collaboration to establish IdM standards, e.g. (Liberty Alliance project and FIDIS); and ***Consolidated Specifications***, that aims to describe the IdM frameworks, e.g. (SAML, Kerberos, Higgins, OpenID, Shibboleth, STORK, PICOS and CardSpace). However, the most deployed IdMs in the IoT environment, as stated by [43], [72], [76], [82], [83], are: Liberty Alliance, OpenID, OAuth, Shibboleth, Higgins, PICOS and STORK.

The Liberty Alliance [84] is a project that implies collaborated companies and organisations, established in 2001, that aims to establish IdM standards, recognizable identity federation, cross-domain authentication, and session management. The implementation process is mainly supported by the SAML[85] standards to promote *ID-FF* (the Identity Federation Framework) and *ID-WSF* (the Identity Web Service Framework). In which, a single federated identity issued by an IdP to the user is used to access services from any SP within the same CoT. The SPs offer their services based on a pre-established trust relationship with the IdP [56]. Although it supports a level of user privacy and network identity security through using pseudonyms, it does not support any form of identity authentication and serious security issues could be found in open and untrusted environments. Moreover, it does not consider the actor relationship between the user and the communication device or defining a strength identityin this architecture [86]. The Kantara Initiative [87] is the next evolution, which since 2008 has targeted the collaboration required to solve the identity issues. In addition to its working groups, its global community includes “CA Technologies”, “Experian”, “ForgeRock”, “Digi.me”, “Internet Society”, “Nomura Research Institute”, and “SecureKey”.Shibboleth [88] is a federated IdM from the Internet2 consortium. It targets sharing resources between research and academic institutions based on SAML2 and web redirection. Although, it has many sharing points with the Liberty alliance, its framework targets a much smaller application area at universities. Shibboleth presents a common interface between the academic institutes in terms of authentications systems. In Shibboleth, the IdP optionally uses proof-of-rightful-possession. Thus, users might not be supported with a proof of rightfully processioning the token. This could raise a security problem. Again, the actor relationship is missing. Moreover, it does not support multiple authentication by related $IdPs$, nor even user’s attribute aggregation, nor single-singe-out [89].

OpenID [90] is a decentralised framework for user-centric IdM. OpenID facilitates accessing services from different SPs by the Internet users based on a single digital identity. The idea behind that is the user has to have account or more with the OpenID IdP, which in turn supports him/her with a global identifier following the URL format; then, the user uses this identifier to request services from any SP compatible with the OpenID [91]. The user controls the selection of a suitable identifier, if it has multiple ones, for the intended SP. “Google”, “IBM”, “PayPal”, “Microsoft”, “VeriSign”, and “Yahoo!” are main contributors for the OpenID. For instance, a user could request to access some documents from web-services using his/her Gmail identifications. However, it does not consider the actor relationship and could suffer from across-domain $IdMs$ interoperability problem in an open environment such as the IoT. Moreover, SP relies on the OpenID for the user authentication, which means the SP does not have an authentication method to verify the requester’s identity. Besides, privacy issues result from using URL identification.

The Eclipse Higgins [92] is an open source software that promotes extensible, platform free, identification protocol free, software architecture to current and future applications to support users security and control over their identity. It is interoperable with all identity protocols such as “WS-Trust”, “OpenID”, “SAML”, “XDI”, “LDAP”, etc. It promotes building a IdM application in different contexts and improves the interoperability between $IdMs$ through defining a new layer “context”. Thus, the digital identities are linked across different $IdMs$. It truly represents a user-centric IdM with federated identity. User’s pseudonym is applicable in Higgins. However, the actor relationship is also missing.

OAuth 2.0 framework focus on defining users authorisation protocol rather than authentication. It is used by the “resource owner” to authorise a third-party client, on behalf of the owner, to access/perform an action on the resource in a “resource server” [93] without sharing his/her credentials with the third-party. It relies on HTTPS (“Secure Hypertext transfer protocol”), which uses the TLS (“Transport Layer Security”). It supports authorisation in web-based applications, on-board applications, mobile phones and home appliances [94]. A typical example of this framework is sharing a user’s private photos or videos from google drive with a friend without sharing the owner identifications. Again, the actor relationship is missing.

PICOS is short for “Privacy and Identity Management for Community Services”. It is a European project aimed enhancing identity privacy and managing the trust feature in community services with mobile communication SPs [95]. It allows users to manage and control their partial identities through a GUI tool to create communities of users. This helps with creating “private rooms”, which represent a restricted area where the user can share his/her partial identity with selected users to offer services or share resources. It could be considered as a privacy-enhanced Facebook to serve a group of users like anglers, taxi drivers, football fans, or game players. It uses a “blurring” concept, to hide the mobile users’ identities and their actual locations from others within a predefined area. Users could be notified about their identities disclosure situations [96]. However, there is no use for the devices identities nor its relationships with the user to identify and authenticate the user in these social roams.

STORK is short of “Secure idenTity acrOss boRders linked”. It is a user-centric IdM framework co-funded by the European Union. Its aim is to authenticate citizens and employees by any State of the EU using the eID. The STORK platform role allows the SPs to get the user identity with his/her consent [97]. Again, the device identity, user relationship with the device are missing.

2.6.2. $IdMs$ in IoT Research Projects

In the research community, there are few systems that target developing IdM in the IoT, which are summarized as below.

Mahalle proposed an identity management layer with a set of processes for IoT in his dissertation [71]. The author relies on the context to define a separate context identity (CID). Moreover, he supports a context awareness by applying a namespace dependent identifier to the communicated device. The key milestones in the proposed framework are "Context management", "identity binding", "identity mapping" and "lifecycle management", which use credentials and identities as an input. However, the proposed solution ignores the user identity and his relationship with the device.

Chibelushi, et al. [83], proposed a user-centric IdM framework for healthcare in IoT. They claim that all the healthcare devices use ad-hoc network in their communications. They target the devices and users identification when sharing devices and create a seamless interaction in IoT environment. The system clearly separates between the user's identity and the device's identity, which allows monitoring of the moving devices. However, the proposed framework does not address the identification in device-to-device communication nor across-domain.

Butkus [64], proposed a user-centric IdM within IoT's gateway architecture. The proposed IdM supports a federated model and incorporates three component users, SP, and IdP. The proposed IdM targets people and devices to interact and collaborate based on users' identities and the "relationships between users". However, the authors do not describe those relationships clearly in the solution.

Zdravkova [72], proposed a user-centric IdM within a cloud-based IoT architecture by using an identity agent in the computing devices. The researcher focuses on using the identification of a single thing (device) with an SP to identify the other things belonging to the user (called Single-Thing-Sign-On). The proposed IdM uses the relationships between a human user and the things without clearly defining those relationships. Although the authentication component exists in the model, the authentication mechanism is missing.

Abreu et al. [98] proposed a user-centric IdM within the "Advanced Metering Infrastructure" in the ICT. The researchers focus on security and privacy of the operator/engineer identity when remotely accessing the smart meters. The operator/engineer identity is represented by a token which is issued by the company's IdP. A RTU ("Remote Terminal Unit") is used as a broker between the smart meter and the requester which is in charge of validating the requester identity within the authorization server [91]. Again, they did not consider the device identity, its relationship with the requester, nor the establishment of a dynamic trust relationship between the communicated parties.

Bernabe et. al. [99] proposed a privacy-preservation IdM using a hybrid model. They focus on integrating user authentication and access control methods with the claim-based machine to machine environment. Users/objects, as a main actor in the IoT environment, delegate their partial identity to get the Identity Mixer (Idemix) credential to maintain the privacy. Furthermore, the classic IdM, i.e. FIWARE (Keyrock), are used to support the SSO or identity federation feature and to support actor identification using SCIM ("System for Cross-domain Identity Management") standard [100]. By doing this, they adapt the classic IdM with IoT features. However, the impact of the relationship between the user and the communication object on the identification are missing again.

To sum up, the above IdM solutions are designed to work in the IoT environment. However, the above discussion show the pros and cons for each of them. None of them supports the establishment of a dynamic trust relationship between unknown SP and $IdP(s)$ and a relationship-based identity establishment. Therefore, a new IdM system to support attribute sharing is required to overcome these limitations in the current IdM solutions.

2.6.3. Identity Verification Approaches

The IdM systems have to be integrated with an interoperable authentication scheme. This section summarizes the existing authentication approaches.

Kerberos is a well-known identity authentication protocol that is designed based on the Needham-Schroeder symmetric key protocol [101]. Kerberos is used to authenticate users in different service providers, which are managed by the related administrative domains [102]. In Kerberos, a ticket granting server (TGS) issues a ticket granting tickets (TGTs) for their clients as an identity authentication to use to request services within a domain or CoT (the realm in Kerberos terms). However, Kerberos is not suitable for an open environment like the IoT for the following reasons: (1) the TGT is unknown across-domain or CoT; (2) the difficulty of store shared keys of all SPs by each TGS; (3) the lack of offering a dynamic mutual authentication between the independent SPs, and $IdPs$; and (4) the lack of generating the TGTs in a real-time because of the use of stored ticket on the device to identify the end user to trusted SPs instead of creating a new one at the request time.

Chin et al. [103] presented an user-centric IdM framework for IoT to authenticate the user based on a random challenge code and a multichannel authentication. However, the across-domain authentication and securing the communication channel are not considered. Witkovski et al. [104] proposed across-domain authentication using asymmetric key encryption. However, the user's relationship with the communication device is missing. Cagnazzo et al. [105] used the QR-code to authenticate the smart objects in the dynamic environment like IoT. However, the object relationship with a real user and a formal validation are missing.

Salman et al. [106] proposed an identity-based authentication scheme for heterogeneous IoT. The scheme is built on the use of gateways to link heterogeneous things with a central data repository. However, such a model suffers from the scalability issue because the IoT contains a sheer amount of entities. Moreover, the entities interactions and its role in the authentication is neglected. Sharaf-Dabbagh et al. [107] proposed an authentication approach for devices based on their unique fingerprints. The approach identifies a unique fingerprint for each device. The fingerprint consists of multiple factors such as a location, a physical state of object, or a transmitter state. Again, they focus on authenticating objects in the IoT rather than the user behind the objects. Liu et al. [108] proposed an authentication and access control approach for things and users in IoT. The ECC (Elliptic Curve Cryptography) for IPv6 is used to secure the protocol. However, they did not consider the interaction between the user and the things. Mahalle et al. [109] presented an identity authentication scheme and capability-based access control (IACAC). The ECC is also used to secure a one-way authentication and a mutual authentication scheme. Again, the user interaction with the communication object is ignored. Ranjan and Hussain [110] proposed a terminal M2M authentication in the IoT. They proved the feasibility of the PKI (public key infrastructure) and the digital signature to authenticate the terminals using a randomly generated key. However, a mutual authentication for the terminals is missing. A scheme for user conditional privacy-preservation authentication and access linkability (CPAL) is proposed by Lai et al. [111] to support a roaming service in the IoT. A trust linking server is used to authenticate the service requester anonymously across-domain based on a master linking key. However, the user relationship with the communication devices is missing again. Rafidha Rehman and Veni [112] proposed an infrastructure to authenticate sensor-enabled mobile devices in IoT based on a ZKP (Zero-Knowledge Prof) and an accumulated hashing. Chaturvedi et al. [113] also proposed a multifactor authentication scheme for a remote user in IoT. The authentication factors are a smart card issued by a server, login password, and biometrics. Again, this scheme neglected a broad range of tiny devices that full the IoT. Other multi-factor user authentication schemes in IoT is presented in [114] and [115].

They use the biometrics data as a user identification. However, these schemes lack generality for IoT scenarios.

In summary, the literature review shows the identity verification and authentication in the IoT environment is an attracted area of research. However, none of the existing proposals supports the following features: (1) establishing a dynamic trust relationship between the independent SP and $IdP(s)$ to support across-domain authentication, and (2) establishing a requester identity based on their relationship with the communication object(s). Therefore, there is a need for an effective identity verification protocol that considers these features.

2.7. Summary & the Way Forward

This chapter presented: firstly, a brief background of the identity and the identity management system; and secondly, reviewing the existed solutions in the literature to check its ability to solve the research problem. By reviewing the state of the art of identifier proposals in the IoT, conducted in Section 2.3, it is found that they were ignoring the actor's relationship type between the actual requester (EA) and the communication device(s). Moreover, the Internet connectivity criterion of the interacted entities, which is essential to recognise their role in the communication. Therefore, they are not able to truly represent the actul requester behind the communication device(s) that have interacted with to the service provider. It is concluded that there is a need for a general identifier format to represent the actual requester actor (EA) properly to any service provider in the IoT environment. This identifier has to contain all the required attributes in a semantic format to be recognisable across its IdM domain. Formulating this identifier will be discussed in detail in chapter 3.

With regards to the IdM, the state of the art in IoT review, presented in Section 2.6, concludes that the IdM solution which supports establishing the identity of the requester based on their relationship with the communiation object by SPs at any domain in IoT is missing. To solve this issue, a new IdM is required that supports SPs to seamlessly interoperate with external $IdPs$ based on the establishment of a dynamic trust relationship to identify the actor's identity. Moreover, a new identification system which relies on an effective identity verification protocol is required. The new IdM design will be discussed in detail in chapter 4.

Chapter 3.

A Global Actor Relationship Identifier ($GARI$) for IoT

This chapter analyses the IoT typical scenarios to get a set of requirements to establish the EA identity. These scenarios have been chosen to show the domains interaction and the actor interactions. Later, the scenarios will be consolidated to generate a general actor interaction use-case in the IoT. Moreover, the requirements will be merged to generate a set of requirements that should be fulfilled by a new IdM to establish the effective actor identity in an IoT environment. Furthermore, the design of a general identifier format that supports establishing the EA identity by any SP in the IoT environment is discussed. It starts by arguing the actors' relationship. Next, the global actor relationship will be modelled using a mathematical model. Finally, the global actor relationship identifier ($GARI$) will be formulated to represent the actor relationship types and the actor identity attributes. $GARI$ is designed to fulfil the requirements that were captured by analyzing the typical IoT scenarios.

3.1.The General Use-Case for Actors Interaction in IoT

To offer a right service, the SPs have to identify who is the actual requester, i.e. the EA, behind the interacted communication object rather that the object identity. Thus, it is important to identify the actors interaction criteria to consider as the basis for eliciting the identity establishment requirements, proposing a suitable solution, and finally, evaluating the proposed solution.

By reviewing the common IoT design architectures in literature, which are summarised in Appendix C, and map them with the existed IoT applications, it is recognised that the interaction criteria can be categorised into three types as follows: domain interactions, actor interactions, and the interaction mode. The domain interactions describe the domains collaboration that might manage by different $IdMs$ to identify the requester identity and offer the right service. It implies the interactions within a domain/CoT, across the domain/CoT, and hybrid. A CoT (circle of trust) is a collaboration of a number of enterprises to share their resources and services with other members. The actor interactions describe the relationship between the EA and the communication object(s). It implies permanent, semi-permanent, and open-access interaction. Both Actors and domains could have a single or multiple interactions mode to perform the required tasks.

From a service provider point of view, the general use-case for actor's interaction implies the interaction of different entities, which are defined as follows:

- ***The effective Actor*** *(EA)*, which is any entity from the IoT environment that intends to consume or produce a service or data.
- ***The communication object*** (co), which is a device that communicates on behalf of the actual actor (EA) based on its relationship with the EA. The co can use a varity of communication technologies to perform the required task;
- ***The service provider*** (SP), which represents any entity of interest that has valuable data, actions, or services to other interested parties.
- ***The Identity Provider*** (IdP), which represents a specific entity which is responsible for managing EA and co identity attributes and delivering its services to the SP.

These entities represent the building blocks of the IdM systems. They interact with each other following the interaction criteria, illustrated in Figure 3-1, in order to perform their tasks. These criteria are explained in the following paragraphs.

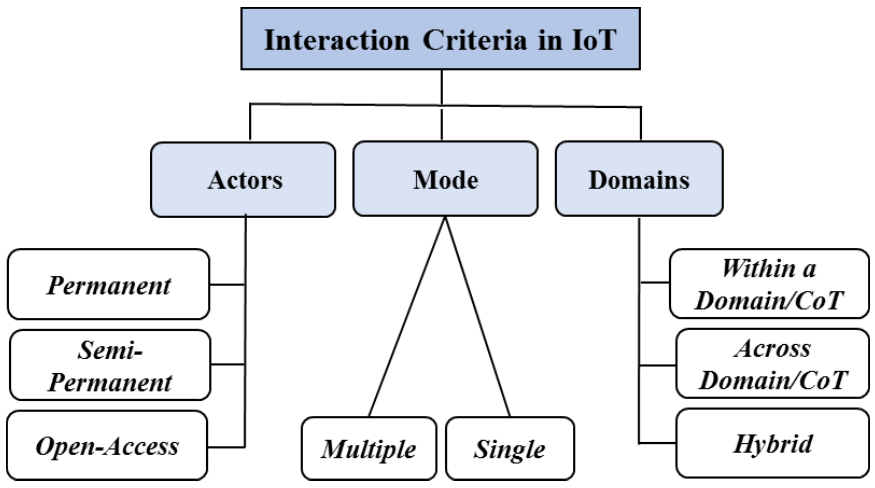


Figure 3-1: The Interactions Criteria in IoT

The co contacts the SP on behalf of another interested entity, i.e. the $EA,$ to request a service or share data. However, the co characteristic is either active or passive in terms of Internet connectivity. Thus, there are two mode of interactions to allow the EA access the required services or data as follows.

- ***Through a single interaction with an active co***. The Co can access the Internet directly, thus, this interaction is enough to complete the task.
- ***Through multiple interactions with multiple cos.*** The first interaction is between the EA and $co1$. If the Internet connectivity of the $co1$ is passive, the $co1$ cannot access the Internet. A second interaction is required between $co1$ and $co2$ to link the EA to the Internet. Again, if the Internet connectivity of the $co2$ is passive, so further interactions are required until the requirement of the active co is satisfied.

Each interaction represents a relationship between two actors. These actors' interaction could have different forms in terms of actor-based interaction as follows:

- ***Permanent***, which means only one EA could use the co. Such relationships have to be established and recognised by the related $IdPs$. Thus, each member of the relationship could be used to identify the other.
- ***Semi-Permanent,*** which means a group of EAs who are permitted to use a group of co(s). Such relationships, also, have to be established and recognised by the related $IdPs$. Thus, each member of the relationship could be used to identify the other.
- ***Open-Access***, which means that co could be accessed by any EA without a pre-established relationship. Thus, none of the relationship' members could be used to identify the other because the related $IdPs$ do not have a fixed record of this relationship type.

Moreover, the $co(s)$ are seamlessly interconnected by anyone, anywhere, and anytime on behalf of their $EA(s)$. The "anywhere" means they are collaborated within or across the IdM realm. However, there is no guarantee that all of the $SP, co(s),$ and the EA belong to the same service domain. Consequently, there is no guarantee that they are managed by the same IdM, or even within a CoT. Therefore, the actors interaction could follow one the following forms in terms of domain-based interaction:

- ***Within a domain/CoT***, in which that EA, co, and SP are managed by a single IdM or by different $IdMs$, but within a single CoT. Therefore, the trust relationship between the SP and the $IdP(s)$ already exists.
- ***Hybrid domains,*** in which some of the domains that manage the EA, co, are trusted by the SP. That means the SP is required to trust those unknown $IdM(s)$ prior to considering them to establish the EA identity.
- ***Across a domain/CoT***, in which the EA, co, and $SP(s)$ are managed by different $IdMs$. However, the trust relationship between the SP and the IdP(s) does not exist. To complete the tasks, the SP has to dynamically establish the trust relationship(s) with the IdP(s).

3.2.The IdM Requirements to Establish an Effective Actor Identity

As discussed in chapter 1, the IoT provides an environment for different actor types, such as people, sensors, devices and objects, to interact. They are registered with one or more service domain IdP; each supplies the actors with an identifier based on their roles. In other words, an actor could have as many identifiers as its roles in the domain. To establish an EA identity in a large-scale environment, such as the IoT, there is a need to elicit a set of requirements to consider as a basis to design an effective solution. For this purpose, a scenario-driven approach will be followed because it is a commonly used approach to elicit the design requirements for the identity management systems [116]–[118]. The typical IoT scenarios have been chosen to show the actors interaction criteria, discussed in Section 3.1. Analysing typical IoT's scenarios leads to identify a set of requirements to establish the identity. The analysis details of these scenarios available in Appendix D. Final requirements to establish the EA identity in the IoT are resulted from merging those identified in all scenarios. Table 3-1 illustrates the $IdMs$ requirements to establish the EA identity in IoT.

Table 3-1: The IdM Requirements to Establish the EA Identity in IoT

Requirement	Description
Req. 1	***Decoupling identities of related actors.*** The SP should be able to differentiate between the EA identifier and the communication object/device identifier. As these entities are related actors, this requires representing them in a semantic format.

Req. 2	***Identifying the home IdP for the actor.*** Each actor's identifier should be paired with its native IdM registration domain identifier. This is due to two IoT's facts: (1) services in the IoT could be requested within one domain (intra-domain) or across multiple domains (inter-domain); (2) the entities' nomadic nature with the aim of consuming services offered by any SP anywhere. Thus, the SP (or the visited domain IdP) must be aware of the domain that manages the identifier to be involved in the EA identity establishment process.
Req. 3	***Identifying actor's attributes.*** The SP should establish the EA identity before provisioning the request. Generally, it is important for the SP to recognise the following: ➢ How does the EA interact with the communication object(s) to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmits the data/ request. ➢ What is the EA type (i.e. Person, Device, System or Application) that maps each actor to its permitted role in the domain? ➢ What is the Internet connectivity type (i.e. passive or active) of the communication that permits the actor to take its specified role in the domain? ➢ Dose a transitive relationship exist? The SP should recognise the transitive relationship type between the EA and the communication object(s) that transmitted the data/ request.
Req. 4	***Actors' identity delegation.*** The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship representation
Req. 5	***The IdP awareness of actor relationships.*** The communication object(s)/device(s) should be aware of their relationship with the EA actor, on whose behalf they communicate. This relationship should be registered within the actor domain $IdP(s)$. It should also be identifiable, recognisable and provable by the SP.
Req. 6	***The establishment of a dynamic trust relationship.*** The SP should be able to establish a dynamic trust relationship with the IdP of unrelated domains in order to involve it in the identity verification.

Req. 7	***Relationship-based identity establishment.*** The SP should establish the EA identity based on its identifier and the actor relationship instead of the physical address, such as the IP address. This is because the physical address like the IP address refers to the communication object location on the network rather than its end user.
Req. 8	***Effective protocol to share the actor's attributes.*** A new authentication protocol is required which should allow SPs to establish the EA identity based on its relationship(s) with the communication device(s) and the actor's characteristics.

3.3.Notations

This section presents the IdM Notations and their meaning that will be used to explain the proposed identification system for the IoT environment. Table 3-2, shows the utilised notations and its description.

Table 3-2: The Notations and their Meanings

Notation	Meaning
$\boldsymbol{a_x}$	The actor x
$\boldsymbol{co}$	The communication object actor
$\boldsymbol{AR_{a,b}}$	The actor relationship between the actors a and b.
$\boldsymbol{AR^a_{a,b}}$	The actor relationship between the actors a and b at the IdP_a.
$\boldsymbol{TR}$	A transitive actor relationship
$\boldsymbol{IdP_x}$	Home IdP that manages the identity of actor a_x
$\boldsymbol{SP_v}$	The visited SP intending to verify the identity of actor A_x
$\boldsymbol{ID_x, ID_{IdP_x}, ID_{SP_v}}$	The identifier of A_x, home IdP of the actor x, and the visited SP respectively.
$\boldsymbol{ARS_{a,b}}$	The $Actor\_relationship\_Specifier$ for $AR_{a,b}$
$\boldsymbol{T_m}$	Timestamp of the message m,
$\boldsymbol{TL_{TDR}}$	Trusted List of TDR
$\boldsymbol{L2TA(SP)}$	Local List of Trusted Agents at SP
$\boldsymbol{TDR_{IdP_x}, TDR_{SP_v}}$	A Trusted Domains Registry of IdP_x, SP_v respectively.
$\boldsymbol{K^+_{IdP_x}, K^+_{SP_v}}$	The public key of IdP_x, SP_v respectively
$\boldsymbol{K^-_{TDR_{IdP_x}}, K^-_{TDR_{SP_v}}}$	The private key of TDR_{IdP_x}, TDR_{SP_v} respectively.
$\boldsymbol{Adrs_{IdP_x}, Adrs_{SP_v}}$	IP address of IdP_x, SP_v respectively.
$\boldsymbol{N^a_{AR_{a,b}}, N^b_{AR_{a,b}}}$	The nonce of $AR_{a,b}$ at IdP_a, IdP_b respectively.
$\boldsymbol{N_Z}$	Nonce of Z.

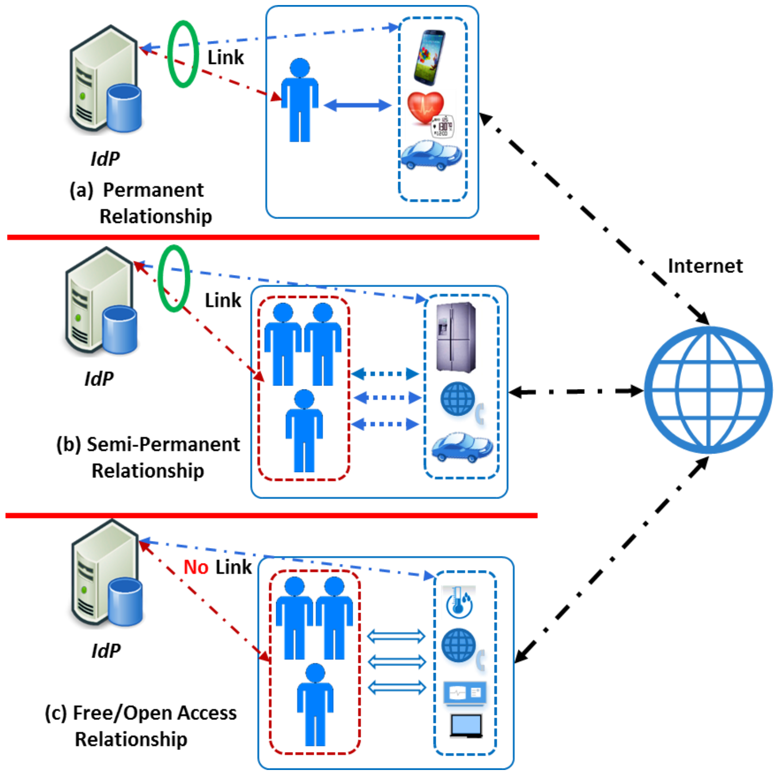
$\boldsymbol{TDRQ(ID_{IdP_x})}$	A request message to a TDR to verify IdP_x
$\boldsymbol{TDRA(ID_{IdP_x})}$	An answer message to a TDR to verify IdP_x
$\boldsymbol{IVR_{AR_x}}$	Identity Verification Request message with respect to AR_x.
$\boldsymbol{IVA_{AR_x}}$	Identity Verification Answer message with respect to AR_x.
$\boldsymbol{V}$	The Identity verification result

3.4. Actors' Relationship Types

In the IoT, communication objects collaborate/interact with each other to serve interested parties that could be a user, a company, etc. Offering the right service requires identifying the actual actor/user (the EA) correctly by the SPs in the IoT. This interaction can be found between people and their related devices or things, between different communicated devices, between people and applications/services, or between devices and applications/services. As discussed in Section 3.1, identifying these relationships has a bearing on truly identifying the actual actor of the communicating device(s), as it will lead to offering the right service to a true requester. Figure 3-2 illustrates the types of actors' relationships in IoT, which are defined as follows:

1. ***Permanent relationship*:** In this relationship type, the communicated objects are collaborated to offer services to only one Actor. It represents the classic way of interaction between EAs and their owned communication object(s) to access services via the Internet. The communication device identity is widely used as a secondary identity for the EA. Such a relationship could be found with patient monitoring devices, personal equipment, etc.
2. ***Semi-Permanent relationship*:** The interaction between actors is not always fixed, it could be changeable. This type is typically seen in the cases where a group of devices and objects are authorized to serve or be used by multiple EAs. The interaction lifetime is varied; it can be held for a long or limited period. Thus, the communication objects identity can help to identify the EA. The long period cases can be found, for instance, in university staff and students use cases who are authorised to use the university PCs to participate in online conferences. A short lifetime relationship can be seen in the cases where another actor is permitted to use an object for a predefined period. Such relationships have to be pre-established with the actors before requesting the service.
3. ***Free / open-access relationship*:** In this, the EA could interact with the communication object and access the Internet without requiring a pre-registration, as seen in the other types. In such cases, the object identity cannot be considered to identify the EA identity because it represents a gateway/broker for the EA to access the Internet. Using an airport's

public personal computer or stores self-check out machines are examples of this relationship.



Link
IdP
(a) Permanent Relationship
Internet
Link
IdP
(b) Semi-Permanent Relationship
No Link
IdP
(c) Free/Open Access Relationship

Figure 3-2: Actor Relationship Types

In the first type of relationship, i.e. a permanent relationship, each of the relationship participants have to be able to identify the other party. In other words, the identity of each participant has to be linked to the other by precisely registering it with their home $IdPs$. For instance, a patient medical record with a medical centre would be able to identify the health monitoring device that is attached to the patient and vice versa.

Similarly, in the second relationship type, i.e. the semi-permanent, a group of actors has multiple relationships with a group of devices/object through many-2-many relationships. However, the communication device/object identity is not sufficient to attribute its actual user, i.e. EA, hence it could help as a secondary identity for the EA. In other words, the identities of EAs a group of authorized communication devices/objects are linked and managed by the domain IdP. In such interactions, the EAs identity represents a primary identity, while the devices/objects identities represent a secondary identity for the EA. Consequently, each relationship participants would be able to identify the second participant identity.

Finally, the free relationship type would not help to identify the relationship participants. This is because it is established without updating the participants' record. Therefore, it could not be used to identify the identity of the participants.

In the IoT vision, things can interact with others to get services or data regardless of their communication technology. In another words, the EA could interact with the communication object(s) and access the Internet following one of these relationships. To establish the EA identity in the IoT environment by any SP, the SP should be able to determine who the EA is? Who is communicated on his/her/it behalf? What is the relationship between them? Which domain IdP could participate in the identification and authentication process?

3.5.Modelling Actor Relationships

As discussed above, the relationships between IoT actors have an essential role in identifying the effective actor of the communicated one. These relationships could be represented as follows.

3.5.1. Definitions

Definition 1. IoT Actor

Let A_{IoT} represents the set of all Actors in the IoT environment.

$$A_{IoT} = \{a_1, a_2, \dots, a_n\} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad (3.1)$$

Where,

$$\forall\ a_l \in A_{IoT}, a_l = Person \mid Device \mid Application\ or\ Service;$$

$$l = 1, 2, \dots, n;\ n = total\ number\ of\ actors.$$

That Actor (a_l) could be a person, a device, an application or a service that interacts with other objects to perform a required task.

Definition 2. Primary Actor

An Actor could be classified into *Primary* or *Secondary* according to the purpose of the communication in IoT. A *Primary Actor (A_P)* represents a subset of A_{IoT} that tends to initiate or consume services with no Internet connectivity. A_P could be defined as follows:

$$A_P \subset A_{IoT} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad (3.2)$$

Where,

$$\forall\ a_i \in A_P, a_i = entity \mid object;\ i = 1,2, \dots, m;$$

$$m = total\ number\ of\ primary\ actors$$

Definition 3. Secondary Actor

A *Secondary Actor* (A_S) represents a subset of A_{IoT} composed of communication objects (co) being used by an actor (a_i) to perform a required task. Members of A_S could be either object or thing, such as a tag reader, an IoT gateway, a mobile device, a PC, etc.

$$A_S \subset A_{IoT} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.3)$$

$Where,$

$$\forall\, co_j \in A_S, co_j = object \mid thing\ ;\ j = 1,2,\ldots,p;$$

$$p = total\ number\ of\ secondary\ actors$$

3.5.2. Actor Relationship

A *communication object* (co) can be categorised according to its *Internet Connectivity* (IC) into two types of A_S. The first type is *Active Object* (O_A), which is a (co) with the ability to connect to the Internet (implements the Internet Protocol IP stack), such as a smartphone. The second type is a *Passive Object* (O_P), which is a (co) that does not have Internet connectivity and relies on another O_A member to access the Internet. Typical examples of such objects are a tag (e.g. RFID, BT, or NFC), a body sensor node, etc. These O_A and O_P could be defined as follows:

$$O_A = \{co_t : Co_t \in A_S \wedge co_t\ have\ stack\ of\ IP\} \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.4)$$

$$O_P = \{co_u : co_u \in A_S \wedge co_u\ does\ not\ have\ stack\ of\ IP\} \ldots\ldots\ldots \quad (3.5)$$

The *Internet Connectivity* (IC) of A_S members could be defined based on (3.4) and (3.5) as follows:

$$IC(co_k) = \begin{cases} Active, & co_k \in O_A \\ Passive, & co_k \in O_P \end{cases} \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.6)$$

Where,

$$\forall\, co_k \in A_S; k = 1,2,\ldots,p$$

To identify the effective actor of any communicated object, in the IoT, the interaction between them is required to be explicitly represented using a relationship. Let an actor relationship, denoted by "AR", represents an interaction of two IoT Actors. The first actor is

$(a_i \in A_P)$ that interacts with the second actor $(co_j \in A_S)$ to allow (a_i) to fulfil a required task. The "$AR_{i,j}$" could be defined as follows:

$$\forall\, a_i \in A_P, \exists\, co_j \in A_S$$
$$AR_{i,j} = \text{Uses}\,(a_i\,, co_j) \quad \text{(3.7)}$$

The $IC(co_j)$ type plays an important role to access the Internet, as previously discussed. Depending on the $IC(co_j)$ we have two cases:

- The *first one* is where the $IC(co_j)$ type is *active*; this means the (co_j) is able to link (a_i) to the Internet directly. Therefore, $AR_{i,j}$, as defined in (3.7), is able to link (a_i) to the Internet to become part of the IoT environment.
- The *second case* is where the $IC(co_j)$ is *passive*, which means the (co_j) is unable to link (a_i) to the Internet directly. Therefore, (co_j) is still required to interact with another *secondary actor*, e.g. $(co_r \in A_S)$, to access the Internet. If such a relationship existing between $(co_j \text{ and } co_r)$ and $IC(co_r)$ is *active*, then the (a_i) can link to the Internet through a transitive relationship between $(a_i \text{ and } co_r)$. Otherwise, another relationship is required with an active (co) is still required. Moreover, the *Transitive Actor Relationship* (TR) will show whether a relationship between the actors $(a_i \text{ and } Co_r)$, i.e.$(AR_{i,r})$, exists or not.

  Let us assume there does exist a $(co_r : co_r \in O_A)$, the $(AR_{j,r})$ relationship between $(co_j \in O_P)$ and (co_r), it could be defined using the AR relationship in (3.7) as follows:

$$\text{Let } co_j \in O_P, co_r \in O_A$$
$$AR_{j,r} = Uses\,(co_j\,, co_r) \quad \text{(3.8)}$$

The relationship in (3.8) represents the interaction between a pair of secondary actors where one belongs to O_P and the other belongs to O_A .

We can now generalise the relation in (3.7) by including the special case of cyclic relationships resulted from (3.8) based on the $IC(co)$ in (3.6) in a ***general actor relationship*** for the IoT that is composed of *n* Actors as follows:

Let n = the number of actors, n > 1

$\forall\, a_i \in A_{IoT}\,, 1 \leq i \leq n-1$

$$AR_{i,i+1} = \begin{cases} Uses\,(a_i, a_{i+1}), & \mathrm{n} = 2, a_{i+1} \in O_A \\ 0, & n = 2, a_{i+1} \in O_P \\ Uses\,\left(a_i, AR_{i+1,i+2}\right), & Ohterwise \end{cases} \ldots\ldots \quad (3.9)$$

3.6. Global Actor Relationship Identifier Format

Representing the identity of an "actor" in IoT requires an identifier that contains sufficient information to identify it at any visited domain across its registration one. As discussed in Section 2.3.2, the identity parameters proposed by Mahalle [43], [71], are insufficient to identify neither tiny actors nor actors across their namespace (IdM domain). To resolve this limitation, the identity of an actor is extended to four parameters instead of three by considering the actor's Internet connectivity, partially satisfying ***Req.3*** previously discussed in Section 3.1. In addition, a minor modification of *namespace* parameter to be IdP name is required to facilitate the identity verification process across-domain, to satisfy ***Req.2***.

A new identifier format is developed based on the proposed identity parameters to build the actor identity in the IoT environment. These parameters are the actor ***type, Internet connectivity, identifier*** and ***identity provider identifier*** of the domain that assigned this identifier. Although it seems obvious, it is important to note that actor with active Internet connectivity can only be of a device actor type as it represents the communication device charachteristic. Thus, the Identity of an Actor is represented as follows:

$\forall\, a_l \in A_{IoT}$

$$Identity(a_l) = \left\{T(a_l), IC(a_l), ID_{a_l}, ID_{IdP_{al}}\right\} \ldots\ldots\ldots\ldots \quad (3.10)$$

Where,

$T(a_l)$ Represents the *actor's type*, as defined in (3.1);

$IC(a_l)$ Represents the *actor's ability to access the Internet*, as defined in (3.5);

ID_{a_l} Represents the *identifier* that is assigned to (a_l) by the IdP;

$ID_{IdP_{al}}$ Represents *the domain's identity provider* in which the identifier is assigned to (a_l);

To fulfil ***Req.3*** previously discussed in Section 3.2, a *Global Actors' Relationship Identifier* ($GARI$) has been formulated. $GARI$ has to represent the general actor relationship, which is defined in (3.9), in a way that it is able to show the actor identity parameters defined in (3.10). Thus, the following ($GARI$) format is proposed, that is composed of three main parts as follows:

- *Actors_Relation_Specifier* (ARS), which is used to specify the characteristics of the relationship participants. These are firstly, the type of (a_i) as it is defined in (3.1). Secondly, $IC(a_j)$ to determine the way of contacting (a_i). Thirdly, (TR) to specify the existence of a transitive actor relationship when $IC(a_j)$ is *passive*, as discussed in (3.8). Finally, the relationship type, as discussed earlier in 3.4, which will allow the SP to decide whether the IdP_{a_j} will query to verify the (a_i) identity or not.
- $Identification(a_i)$, it is used to specify the identifier of (a_i), the IdP_{a_i} that assigned this identifier, and secret nonce(s) (N) of the relationship(s) in the IdP_{a_i}, i.e. $N^{a_i}_{AR_{i,j}}$. $N^{a_i}_{AR_{i,j}}$ is fresh and known only to the IdP_{a_i} to prove the relationship originality. The relationship nonce will be discussed in more details in the next chapter.
- $Identification(a_j)$ could be represented in two forms according to the $IC(a_j)$ type in the first part. The first form is similar to the second part to represent the identification of (a_j) when the $IC(a_j)$ type is *active*. Whilst, the second form is to represent the additional actor relationship (if existent) when the $IC(a_j)$ type is *passive.*

The $GARI$ format is defined as follows:

$$GARI = \{ARS, Identification(a_i), Identification(a_j)\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.11)$$

Where,

$a_i \in A_P \subset A_{IoT}; a_j \in A_S \subset A_{IoT}$;

$$ARS = \{T(a_i), IC(a_j), TR, T(AR_{i,j})\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.11.1)$$

$$Identification(a_i) = \left\{ID_{IdP_{ai}} : ID_{a_i} : N^{a_i}_{AR_{i,j}}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.11.2)$$

$$Identification(a_j) = \begin{cases} ID_{IdP_{aj}} : ID_{a_j} : N^{a_j}_{AR_{i,j}} & , \quad a_j \in O_A \\ GARI & , \quad a_j \in O_P \end{cases} \quad \ldots\ldots\ldots\ldots\ldots \quad (3.11.3)$$

$GARI$ contains all the required information that will facilitate identifying the EA by the SP as the end point of service request. Thus, the SPs confidence of offering their services to the right requester will be improved by involving more $IdPs$ in the requester identification process based on the relationship type.

3.7. $GARI$ Representation

As discussed in the previous section, $GARI$ represents actors' interaction between at least two actors as shown by (3.11). These actors are generally classified into a *primary actor*, to denote the EA, and *secondary actor(s)*, to denote the communication device(s). The IoT's actor defined by (3.1) will be represented using a 2-bits binary number as depicted in Table 3-3 in $GARI$ representation.

Table 3-3: IoT's Actors Type Representation Values

Actor (a_l)	$T(a_l)$
Person	00
Device	10
Application/ Service	11

Similarly, these actors' Internet connectivity characteristic is represented using 1-bit binary number as long, as depicted by Table 3-4. It is worthy to note that only a device actor type could have the active Internet connectivity characteristic.

Table 3-4 Actor Internet Connectivity Type Representation Values

Actor (a_l)	$IC(a_l)$
Active	1
Passive	0

The actor's relationship types between two actors ($\text{e.g.} AR_{i,j}$) could be represented in the proposed format in a 2-bits binary numbers as depicted in Table 3-5.

Table 3-5: Actors Relationship Type Representation Values

Relationship type	**Permanent** $(AR_{i,j})$	**Temporary**	
		Semi-Permanent $(AR_{i,j})$	**Free/Open Access** $(AR_{i,j})$
$T(AR_{i,j})$	11	10	00

Back to $GARI$ representation of the actors (a_i) and (a_j) in the relationship $(AR_{i,j})$, it is composed of three main parts. The first part specifies the characteristics of the relationship participants, i.e. the ARS, which is depicted by equation (3.11.1). These characteristics are firstly represented by binary values based on Table 3-3, Table 3-4, and Table 3-5, then concatenates these values in 6-bits binary number. Finally, the resulted binary number converts to a hexadecimal number to secure these values at transfer time on the Internet. This computation is depicted in Figure 3-3.

Table 3-6: The Reserved Symbols for $GARI$

Symbol	Description
(:)	Used to separate the actor identification attributes
(#)	Used to represent that the primary actor and the secondary actor whose participation in a relationship are managed by a single IdP
(∗)	Used to represent that the IdP who manages the actor identity is declared in the next relationship declaration.
(&)	Used to separate the $GARI$ parts

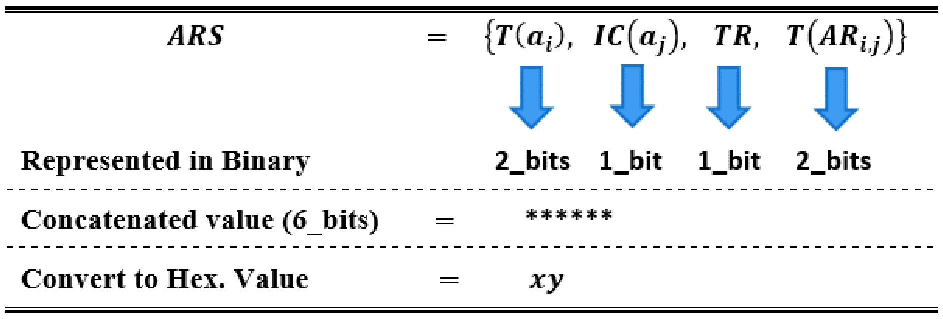

$ARS = \{T(a_i),\ IC(a_j),\ TR,\ T(AR_{i,j})\}$
Represented in Binary
2_bits
1_bit
1_bit
2_bits
Concatenated value (6_bits) = ******
Convert to Hex. Value = xy

Figure 3-3: The $Actors\_Relation\_Specifier$ Computation

The second part of $GARI$ is the primary actor (a_i) identification attributes. It is proposed to represent these attributes i.e. the home $ID_{IdP_{ai}}$ identifier and the actor identifier ID_{a_i} by linking them using the (:) symbol, as depicted in (3.11.2). The third part of the actors' relationship identifier represents the secondary actor attributes, i.e. $Identification(a_j)$. This part could be represented by two different formats based on the $IC(a_j)$ as stated in (3.11.3). The first type follows the same format of the second part, while the second has to represent a new relationship between the (a_j) and $(a_k\ ;\ a_k \in A_S)$, if exists, in the passive type of $IC(a_j)$. The second relationship, i.e. $(AR_{j,k})$, will be represented following the same $GARI$ format to cope with the *general actor relationship* in (3.9).

It is worth to note that these IoT's actors' identities are possible to be assigned by the same IdP. To maintain the $GARI$ size, the unnecessary relisting of the IdP identifier has to be avoided, thus the following notations are proposed.

- If the $ID_{IdP_{ai}}$ and $ID_{IdP_{aj}}$ refers to a single IdP for the actors in a single relationship, e.g. $(a_i), (a_j)$ participating in $(AR_{i,j})$, the primary actor IdP, i.e. IdP_{a_i}, is proposed to use as the principle IdP for the relationship verification. In addition, the sharp symbol $(\#)$ is used to replace the IdP_{a_j} in $Identification(a_j)$. This will inform the SP that the actors are managed by a single IdP
- If there are three actors participating in two relationships, e.g. $(a_i), (a_j)$ participating in $(AR_{i,j})$ and $(a_j), (a_k)$ participating in $(AR_{j,k})$, it is obvious that the (a_j) is shared between the relationships, which will be represented normally in a $GARI$ to represent $(AR_{j,k})$, as defined in (3.11.3). The IdP_{a_i} in $(AR_{i,j})$ will be replaced by the asterisks symbol $(*)$ in the case IdP_{a_i} and IdP_{a_j} is a single provider.

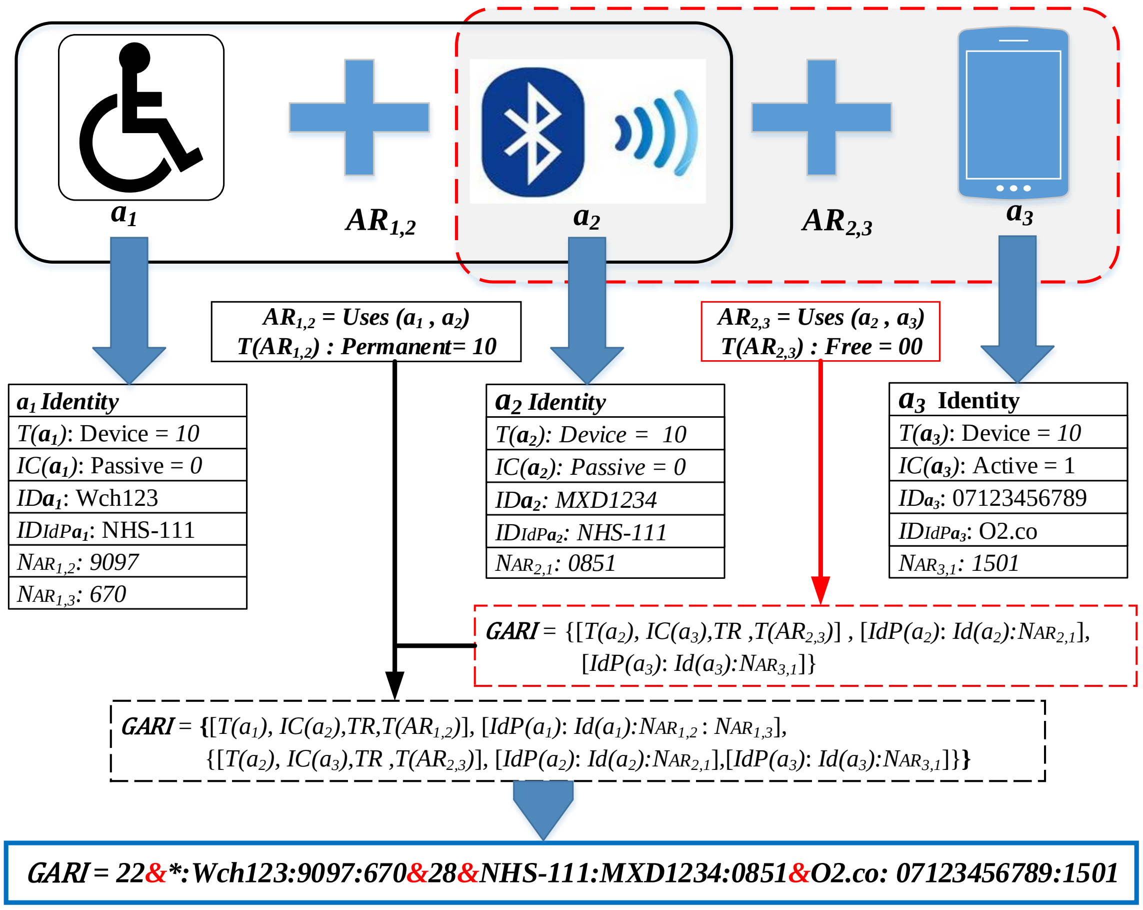
To link these three parts to form the $GARI$, the ampersand symbol "&" is proposed to separate these parameters from each other. Table 3-6 summarises the reserved symbols for $GARI$ representation. The composed $GARI$ will be used when interconnected SPs to identify the effective actor in the IoT environment.

3.7.1. Example

To illustrate the actors' relationship in the $GARI$, let us consider the same scenario that is described in Section 1.2. In this scenario, illustrated in Figure 3-4, there are three actors (a primary actor and two secondary actors) participating in two relationships. The first relationship $(AR_{1,2})$ is between the wheelchair as a primary actor and the BT communication device attached to it. However, $AR_{1,2}$ is unable to access the Internet as $IC(a_2)$ is *passive*. Thus, the second relationship is needed to link the wheelchair to the Internet. The second relationship $(AR_{2,3})$ is between the BT device and the smartphone with WiFi technology to access the Internet. Let us assume that the TR does not exist between (a_1) and (a_3), thus it is considered as a free type, i.e. $(TR = 0)$ in $(AR_{1,2})$ representation. Moreover, the *NHS-111* is the only IdP that could be used to identify the effective actor because of its permanent relationship type and inexistence of a transitive relationship to use the IdP_{a_3}. It is replaced by $(*)$ in $(AR_{1,2})$ because it is the same provider that supplied the identifiers ID_{a_1} and ID_{a_2} and it is defined in the $(AR_{2,3})$.

This way, $GARI$ helps the receiver to identify the effective actor along with all the communication devices that participated in the request. Moreover, the receiver of the $GARI$

identifier can recognise who the EA of this communication is, what his/her/it relationship with the communication devices is, which IdP can support the identity establishment process, and the relationships nonce to use to verify the relationship within the selected $IdPs$.



a_1
$AR_{1,2}$
a_2
$AR_{2,3}$
a_3
$AR_{1,2}$ = Uses (a_1 , a_2)
$T(AR_{1,2})$: Permanent= 10
$AR_{2,3}$ = Uses (a_2 , a_3)
$T(AR_{2,3})$: Free = 00
a_1 Identity
$T(a_1)$: Device = 10
$IC(a_1)$: Passive = 0
ID_{a_1}: Wch123
ID_{IdPa_1}: NHS-111
$NA_{R1,2}$: 9097
$NA_{R1,3}$: 670
a_2 Identity
$T(a_2)$: Device = 10
$IC(a_2)$: Passive = 0
ID_{a_2}: MXD1234
ID_{IdPa_2}: NHS-111
$NA_{R2,1}$: 0851
a_3 Identity
$T(a_3)$: Device = 10
$IC(a_3)$: Active = 1
ID_{a_3}: 07123456789
ID_{IdPa_3}: O2.co
$NA_{R3,1}$: 1501
GARI = {[T(a_2), IC(a_3),TR ,T($AR_{2,3}$)] , [IdP(a_2): Id(a_2):$NA_{R2,1}$], [IdP(a_3): Id(a_3):$NA_{R3,1}$]}
GARI = {[T(a_1), IC(a_2),TR,T($AR_{1,2}$)], [IdP(a_1): Id(a_1):$NA_{R1,2}$: $NA_{R1,3}$], {[T(a_2), IC(a_3),TR ,T($AR_{2,3}$)], [IdP(a_2): Id(a_2):$NA_{R2,1}$],[IdP(a_3): Id(a_3):$NA_{R3,1}$]}}
GARI = 22&*:Wch123:9097:670&28&NHS-111:MXD1234:0851&O2.co: 07123456789:1501

Figure 3-4: An example of $GARI$ Composing

3.8. Summary

The IoT is a technology revolution that will change the relationships between interconnected entities. Identifying these relationships has a direct impact on the identification of the effective actor of the communicated object. Moreover, the Internet connectivity of the communication object leads identifying its ways to access the Internet as it might require establishing an additional relationship when the object is passive. This will allow a broad range of tiny and passive objects to be part of the IoT and recognise them globally by following these relationships. Although previous work has used multiple parameters to identify these entities, such parameters are insufficient to fully describe how entities collaborate to establish a connection to the Internet.

This chapter shows using the scenario-based approach to analyse the typical IoT scenarios in order to determine the actor's interaction criteria, formulate the general actor interaction use-case, and capture the general requirements to establish the effective actor identity in the IoT environment. Moreover, the chapter argued that the identity of entities in IoT could be sufficiently established based on the existence of four parameters: *type, Internet connectivity, entity identifier, and the IdP identifier*. Furthermore, the actor relationship types, in the IoT, have been defined and modelled and then represented in a new semantic identifier format (denoted as $GARI$), to solve this issue. This identifier will be used to represent the effective actor identity when requesting services or data from any SP. The evaluation of $GARI$ efficiency to support establishing the identity will be discussed in chapter 6.

Chapter 4.

Global Identity Management System for the IoT

Today, there are several IdM solutions which have been used in the literature for the IoT environment. However, there is no evidence of a dominant IdM that satisfies the requirements to establish the requester, i.e. EA, identity by any SP at the IoT as discussed in Section 2.6. Therefore, this chapter discusses the proposal of a global identity management system ($GIdM$) for the IoT. This system could facilitate establishing the identity of a service requester across-domain. Yet, it will consolidates the IdM systems to establish the identity of a requester across-domain by proposing a novel global identity verification protocol ($GIdV$) to: firstly, dynamically establish trust relationship(s) between the SP and foreign $IdP(s)$ by using a trusted 3rd party domains registration; and secondly, verify the EA identity.

This chapter covers the following. Firstly, an overview of the proposed $GIdM$ architecture for the IoT is discussed. Secondly, The $GIdM$ components are explained. Thirdly, the $GIdV$ processes are discussed. Finally, the $GIdV$ messages are designed.

4.1.A General Overview of the $GIdM$ for IoT

As previously mentioned, the IdM aims to limit the access of services offered by a SP to a trusted user. This requires a trust relationship between the SP and the IdP to establish the user identity. The IdP could serve a single SP or multiple SPs. Applying this idea in the IoT environment requires extending the IdM models, in Section 2.4, to be flexible and effective to verify the EA identity of nomadic objects that might belong to different $IdMs$. This is because of the difficulty to use those IdM models to establish trust relationships between all SPs and $IdPs$ in the IoT. The solution is by improving these IdMs interoperability to facilitate establishing the identity across-domain. For this purpose, the hybrid IdM model will be followed to propose the $GIdM$ system to globally verify the identity in the IoT.

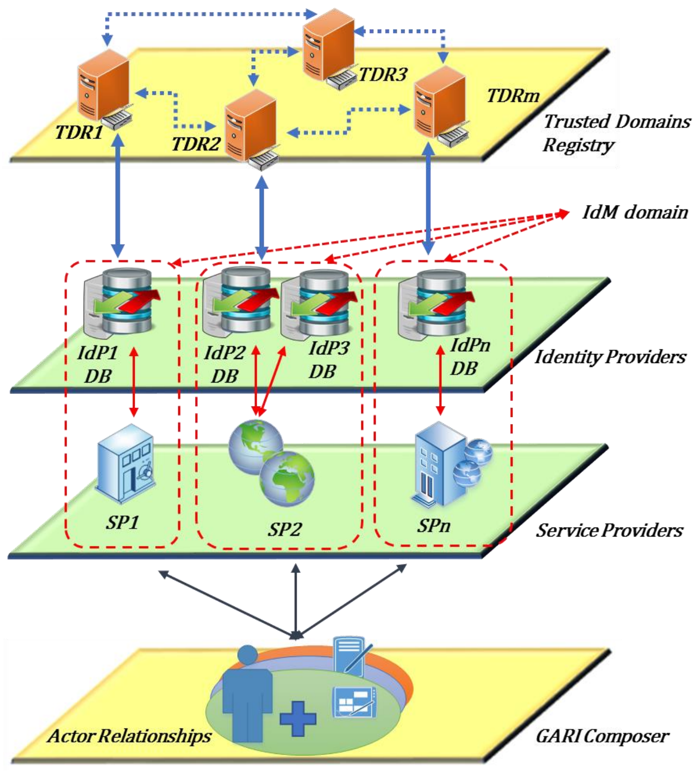


Figure 4-1: The $GIdM$ Architecture

The $GIdM$ architecture contains four main layers, as depicted in Figure 4-1. The first layer from the bottom is the $GARI\ Composer\ layer$ that will be used to compose the proposed identifier to represent the actor relationship to an SP in the IoT environment. The next layer is the $service\ providers\ layer$, which contains the SPs from different $IdMs$. The next layer is the $service\ providers\ layer$, which contains the SPs from different $IdMs$. Each SP could have a trust relationship with an IdP (or even more) to control the access of their services by trusted requesters within the IdM boarder. SPs are responsible for establishing the requester identity by using an identity verification method. Once the requester identity is successfully established, the services will be offered. The third layer is the identity providers layer which contains all the $IdPs$. Each IdP can have a trust relationship with an SP or more. Each trust relationship between the SP and IdP represents a subset of the IdM domain that managed the user identities. Entities within a domain are allowed to use identifiers issued by the IdP responsible for that domain to request a service from SPs within that domain. However, in the IoT, such a trust relationship between an independent SP and the actors' home $IdPs$ might not exist in advance as they can belong to unrelated domains, as seen in the general actor interaction use-case discussed in Section 3.1 above. Thus, an additional layer called Trusted Domains Registry (TDR) is added on top of these layers.

The idea behind using the $TDRs$ is to maintain a dynamic trust relationship between the $IdMs$ entities, i.e. SPs and $IdPs$, across their domain boarder. Each TDR implies a list of trusted SPs and $IdPs$ with their public keys. Hence, the SPs could establish the required trust relationship dynamically with foreign $IdPs$ relying on the data of these $TDRs$. Moreover, each TDR has to reply to its registered entities queries and other $TDRs$ queries. $TDRs$ should maintain their trust relationships with other $TDRs$ to be used later to establish a trust relationship between the interconnected entities, i.e. a SP and an IdP, that might be trusted by different $TDRs$. However, the entities, i.e. SPs and $IdPs$, trust and reputation measurements are another interesting field of study, like in [31], [76], [119], that are currently out of this research's scope.

To establish the EA identity using the proposed $GIdM$, an effective identity verification protocol is required to be followed. The protocol has to meet the requirements discussed in Section 3.2 and be used by all the participating entities in the identity verification. Moreover, the $GARI$ Composer layer will be used to represent the actor relationship at any SP in the IoT environment that was discussed in chapter 3.

4.2. Global Identity Management Components

This Section describes the main components of the $GIdM$ (see Figure 4-1) and their role in the identity establishment processes.

4.2.1. The IdP as a Database

It is clear that the IdP is the repository of identity information for the SP in $IdMs$. Based on the IdM model of design, it might serve a single SP or multiple SPs through pre-established trust relationship(s). It is responsible for the actors identity verification on behalf of the SP. To cope with the research requirements, the $IdPs$ database design in the $GIdM$ should be modified to host extra information. As proposed by the equation (3.10), in Section 3.6, all actors' identification attributes will be stored in the $ActorInfo$ table by the IdP. The identity attributes of an actor are an identifier, a type, and the Internet connectivity type. Table 4-1 illustrate the $ActorInfo$ table at the IdP databases.

Table 4-1: $ActorInfo$ Table Design for the $IdP's$ Database

$ActorInfo$	
ActorId	Char
ActorType	Binary
InternetConnectivity	Binary

Moreover, all identity providers have to have the actors' relationship attributes, which denoted as $Relations$. These attributes are the relation identifier (RId), the identifier of first actors in the relation (Id_a_1), the identifier of second actor in the relation (Id_a_2), the relationship type, and the relationship nonce, as illustrate in Table 4-2.

Table 4-2: $Relations$ Table Design for the $IdP's$ Database

$Relations$	
$\boldsymbol{RId}$	Int.
$\boldsymbol{Id\_a_1}$	Char
$\boldsymbol{Id\_a_2}$	Char
relationType	Binary
Nonce	Int.

In terms of nonce size, it is proposed to use 4-bytes to represent the relationship nonce. The nonce (N) will be created randomly by the IdP_x at the $GARI$ composition time and it is known only to the IdP_x. The IdP_x then computes a one-way hash function $h(N)$ using (SHA1 or MD5) algorithm to secure the nonce and avoid a collision that may be produced with another relationship with the same IdP_x. The probability of generating the same (N) value is $\left(P(N) = \left({}^{1}/_{2^8}\right)^4 = {}^{1}/_{2^{32}}\right)$. The collision probability of getting the same value for the hash function of the nonce (N), i.e. $P(h(N))$, results from multiplying the probability of the hash function $P(h)$ by the probability of generating the same nonce $P(N)$ [120], which is represented by the function in (4.1). Table 4-3 shows the probability of one-way hash algorithms for the relations nonce at the IdP_x.

$$P(h(N)) = P(h) * P(N) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \quad (4.1)$$

Table 4-3: The Probability of Occurrence $h(N)$ Collision per IdP_x

Hash Algorithm $h(N)$	**Output Size**	**Collision Probability**
SHA1	160 bits	$P = {}^1/_{2^{160}} * {}^1/_{2^{32}} = {}^1/_{2^{192}}$
MD5	128 bits	$P = {}^1/_{2^{128}} * {}^1/_{2^{32}} = {}^1/_{2^{160}}$

4.2.2. The Trusted Domains Registry (TDR) as a Database

As discussed above, the $TDRs$ are used as trusted 3rd parties to support establishing trust relationships between the unrelated entities SP and IdP across-domain or $CoTs$. Each TDR has an updated information in its Trusted List (TL_{TDR}) of the trusted $IdMs$ entities, i.e. SPs and $IdPs$. This information implies the entity identifier and the entity pubic key as depicted in Table 4-4. SPs and $IdPs$ have to maintain their information in the TL_{TDR}, thus the TDR could respond to the queries with true and up to date data. The TDR accepts queries from its trusted entities and other $TDRs$ partners. If a TDR does not have the information of an entity in its (TL_{TDR}) then it transmits the query to other partners $TDRs$. It is proposed that the messages will follow the DNS messages in terms of the messages repetition, time interval, and hop limits. Thus, the hop limit of the query message is set to 32 to avoid unwanted delay. The process detail of the TDR will be explained in detail in Section 4.3 below, while the query and answer messages design will be explained in Section 4.4.

Table 4-4: A Trusted List of a TDR (TL_{TDR}) Design

TL_{TDR}	
Key	Int.
Entity_ID	Char
Entity_PK	Int.

4.2.3. The $GARI$ Composer

The preliminary step required to request a service from any SP_v is compose the identifier to be used by objects. It is assumed that each actor, denoted as A_x, has been registered with an IdP_x and will be able to present its identity attributes when needed. Based on that the $GARI$ composition process to represent ($AR_{a,b}$) contains the following steps:

1. The (a), as EA, delegates its identity to the smart object or the gateway of tiny objects, (b). (b) in turn forwards the $Identity(a)$ to the IdP_b.

2. IdP_b checks whether it has a record or more with the identity of (a); if so it checks $T(AR_{a,b})$ in its records by checking if $\{ID_a, ID_b, T(AR_{a,b})\}$ at $DB(IdP_b)$. If the checking is valid, then IdP_b generates a random nonce for this relationship $(N^b_{AR_{a,b}})$, performs the one-way hash operation to get the value $h(N^b_{AR_{a,b}})$, and adds it to their record. Moreover, IdP_b sends a checking request message to IdP_a to check whether the $T(AR_{a,b})$ exists in $DB(IdP_a)$. If $\{ID_a, ID_b, T(AR_{a,b})\}$ at $DB(IdP_a)$ is valid then, similarly, IdP_a generates a random nonce for this relationship $(N^a_{AR_{a,b}})$, computes the value $h(N^a_{AR_{a,b}})$, updates its records and sends the relationship details back to IdP_b. Otherwise, if either IdP_a or IdP_b does not recognise the relationship, then $T(AR_{a,b})$ is considered as a free relationship type and the value $h(N^a_{AR_{a,b}})$ represented by “0”.
3. IdP_b receives the relationship details from IdP_a, and checks whether $T(AR_{a,b})$ at $DB(IdP_a)$ and $T(AR_{a,b})$ at $DB(IdP_b)$ is equal. If the checking is valid, then it computes the three components of the $GARI$ identifier:
   - $ARS_{a,b} = \left(T(a) \parallel IC(b) \parallel TR \parallel T(AR_{a,b})\right)$. $Where\ TR = 0$ by default
   - $identification(a) = \left(ID_{IdP_a} \parallel ID_a \parallel h(N^a_{AR_{a,b}})\right)$.
   - Based on $IC(b)$, $GARI$ is defined as follows:
     - if $IC(b)$ is active, then the $identification(b)$ will be computed first to compose the $GARI$ as follows:
       - $identification(b) = \left(ID_{IdP_b} \parallel ID_b \parallel h(N^b_{AR_{a,b}})\right)$.
       - $\boldsymbol{GARI = \{ARS_{a,b} \parallel identification(a) \parallel identification(b)\}}$.
     - Otherwise, the following optional step (3.1) is triggered.

3.1. ***Optionally,*** if $IC(b)$ is passive, and there exits another actor, e.g. (c), then IdP_c receives the identification of $(AR_{a,b})$ and repeats the steps (1 - 3) for $AR_{a,c}$. If $T(AR_{a,c})$ is not of an open-access relationship type then $TR = 1$ in $ARS_{a,b}$. The $GARI$ components will be computed as follows:

- $ARS_{a,b} = \left(T(a) \parallel IC(b) \parallel TR \parallel T(AR_{a,b})\right)$,
- $ARS_{b,c} = \left(T(b) \parallel IC(c) \parallel TR \parallel T(AR_{a,c})\right)$,
- $identification(a) = \left(ID_{IdP_a} \parallel ID_a \parallel h(N^a_{AR_{a,b}}) \parallel h(N^a_{AR_{a,c}})\right)$.
- $identification(b) = \left(ID_{IdP_b} \parallel ID_b \parallel h(N^b_{AR_{a,b}})\right)$.

- $identification(c) = \left(ID_{IdP_c} \parallel ID_c \parallel h\left(N^{c}_{AR_{a,c}}\right)\right).$

The final $GARI$ identifier becomes:

$$\boldsymbol{GARI} = \left\{\boldsymbol{ARS_{a,b}} \parallel \boldsymbol{identification(a)} \parallel \left\{\boldsymbol{ARS_{b,c}} \parallel \boldsymbol{identification\ (b)} \parallel \boldsymbol{identification(c)}\right\}\right\}$$

The following algorithm will be used to implement the above steps to compose the $GARI$ by a smart object or gateway, a mobile device (smartphone, tablet, laptop, etc.) or a cloud service.

Algorithm 1. $GARI$ Composing Algorithm

```
BEGIN
GET the number of actors $(n)$ that will be used in Identifier composition.
$A_n$ ← GET identity $(a_n) = \left\{T(a_n), IC(a_n), ID_{a_n}, ID_{IdP_{a_n}}\right\}$
$AR^{a_n}_{1,n}$ ← GetRelationInfo $\left\{ T\left(AR^{a_n}_{1,n}\right), N^{n}_{AR_{1,n}}\right\}$
$A_1$ ← GET identity $(a_1) = \left\{T(a_1), IC(a_1), ID_{a_1}, ID_{IdP_{a_1}}\right\}$
$AR^{a_1}_{1,n}$ ← GetRelationInfo $\left\{ T\left(AR^{a_1}_{1,n}\right), N^{1}_{AR_{1,n}}\right\}$
$T$ ← CheckRelationType $\left(T\left(AR^{a_1}_{1,n}\right), T\left(AR^{a_n}_{1,n}\right)\right)$
$TR$ ← 0 (false)
$If\ (n\ ==\ 2)$ Then
    $GARI$ ← ComposeGari $\left(a_1, a_n, TR, T, N^{1}_{AR_{1,n}}, N^{n}_{AR_{1,n}}\right)$
ElseIf
    $tmpN = N^{a_1}_{A_{1,n}};$
        $A_i$ ← $a_n$; $AR^{a_i}_{1,i}$ ← $AR^{a_n}_{1,n}$;
    For $(i\ =\ n\ DownTo\ i\ =\ 2\ )$
      Begin
      $A_{i-1}$ ← GET identity $(a_{i-1}) = \left\{T(a_{i-1}), IC(a_{i-1}), ID_{a_{i-1}}, ID_{IdP_{a_{i-1}}}\right\}$
      $AR^{a_{i-1}}_{i-1,i}$ ← GetRelationInfo $\left\{ T\left(AR^{a_{i-1}}_{i-1,i}\right)\right\}$
      $AR^{a_{i-1}}_{1,i-1}$ ← GetRelationInfo $\left\{ T\left(AR^{a_{i-1}}_{1,i-1}\right), N^{i-1}_{AR_{1,i-1}}\right\}$
      $AR^{a_1}_{1,i-1}$ ← GetRelationInfo $\left\{ T\left(AR^{a_1}_{1,i-1}\right), N^{1}_{AR_{1,i-1}}\right\}$
```

```
        $T \leftarrow$ CheckRelationType $\left(AR_{i-1,i}^{a_{i-1}}, AR_{i-1,i}^{a_i}\right)$
        $TR \leftarrow$ IsTransitiveRelation $\left(AR_{1,i}^{a_1}, AR_{1,i}^{a_i}\right)$
        $TmpGARI \leftarrow$ ComposeGari $\left(a_{i-1}, a_i, TR, T, N_{AR_{1,i-1}}^{i-1}, N_{AR_{1,i}}^{i}\right)$
        $A_{i-1} \leftarrow \{ T(a_{i-1}), IC(a_{i-1}), TmpGARI, *\}$
        $tmpN = N_{AR_{1,i-1}}^{1} : tmpN;$
      EndFor
    $TR \leftarrow$ IsTransitiveRelation $\left(AR_{1,i-1}^{a_1}, AR_{1,i-1}^{a_{i-1}}\right)$
   $GARI \leftarrow$ ComposeGariWithTmp $(A_1, A_{i-1}, TR, T, tmpN)$
      Return $GARI$
END
ComposeGari $(A_1, A_2, TR, T, N_1, N_2)$
Begin
    $ARS \leftarrow Ars\ (T(a_n), IC(a_n), TR, T)$
    $TmpID1 \leftarrow ID_{IdP1} + ":" + ID_1 + ":" + N_1$
    If $(ID_{IdP1} == ID_{IdP2})$
        Then $TmpID2 \leftarrow "\#" + ":" + ID_2 + ":" + N_2$
    Else $TmpID2 \leftarrow ID_{IdP1} + ":" + ID_2 + ":" + N_2$
    EndIf
    $Gari \leftarrow ARS$ + “&” + $TmpID1$ + “&” + $TmpID2$
    Return $Gari$
End
ComposeGariWithTmp $(A_1, A_2, TR, T, N_1)$
Begin
   $ARS \leftarrow Ars\ (T(a_n), IC(a_n), TR, T)$
   $TmpID1 \leftarrow ID_{IdP1} + ":" + ID_1 + ":" + N_1$
   If $(ID_{IdP2} == " * ")$
        Then $TmpID2 \leftarrow " * " + ":" + ID_2$
   ElseIf $(ID_{IdP1} == ID_{IdP2})$
        Then $TmpID2 \leftarrow "\#" + ":" + ID_2$
   Else $TmpID2 \leftarrow ID_{IdP1} + ":" + ID_2$
   EndIf
$Gari \leftarrow ARS$ + “&” + $TmpID1$ + “&” + $TmpID2$
```

```
Return ComposeGari ← $Gari$
End
IsTransitiveRelation($AR_1$ , $AR_i$)
  Begin
    If $T(AR_1) and\ T(AR_i)$) are not equal to 0 and not adjoin
      Then $TR$ ← 1 (true)
    Return $TR$
  End
CheckRelationType ($R_1, R_2$)
Begin
    If ($R_1 == R_2$) then
      Return $T(R)$
    ElseIf    Return 0.
End;
$Ars\ (T(a_1), IC(a_2), TR, T)$
Begin
        $T(A)$ ← $T(a_1)$ = {x| x = 00,10,11}
        $IC(A)$ ← $IC(a_2)$, = {x| x = 1,0}
        $R$ ← {x| x =00,10,11}
    $TmpArs$ ← $Concatenate\ (T(A), IC(A), TR, R)$
    Return $Ars$ ← $hex(tmpArs)$
End
```

4.2.4. The SP as an Identity Establishment Unit

The SP is a main component in the $GIdM$ architecture that controls the service requests in the IoT. This is because it could be a standalone entity like smart devices or an IoT gateway on behalf of other tiny objects like sensors. Thus, it has to manage the EA identity establishment process to offer the right service. As explained in Section 2.4, the SP offers its services to a trusted client by the $IdP(s)$ based on a pre-established trust(s) relationship. In other words, the SP has to collaborate with a single IdP or more to establish the requester identity. The SPs have to have a list of all trusted $IdPs$ details in a Local List of Trusted Agents $\big(L2TA(SP)\big)$. The $L2TA(SP)$ will be used to check whether the requester identity is issued by a trusted IdPor not. Moreover, each SP should have an actor relationship table (ART) to store the actor relationship attributes through the identity establishment processes, as depicted in Table 4-5. Finally, it is worth to note that each SP has to register with a TDR and both have up to date public key of the other to use in the communication as discussed earlier in Section 4.2.2. Figure 4-2 illustrates an overview of the requester identity establishment system by SPs in the IoT that explain in detail in the following subsections.

Table 4-5: The ART table at an SP

ART	
$\boldsymbol{RId}$	Int.
$\boldsymbol{Id\_a_1}$	Char
$\boldsymbol{Id\_a_2}$	Char
relationType	Binary.
$\boldsymbol{IdP\_a_1}$	Char
Nonce	Int.
$\boldsymbol{T(a1)}$	Binary
$\boldsymbol{IC(a2)}$	Binary
$\boldsymbol{V}$	Boolean

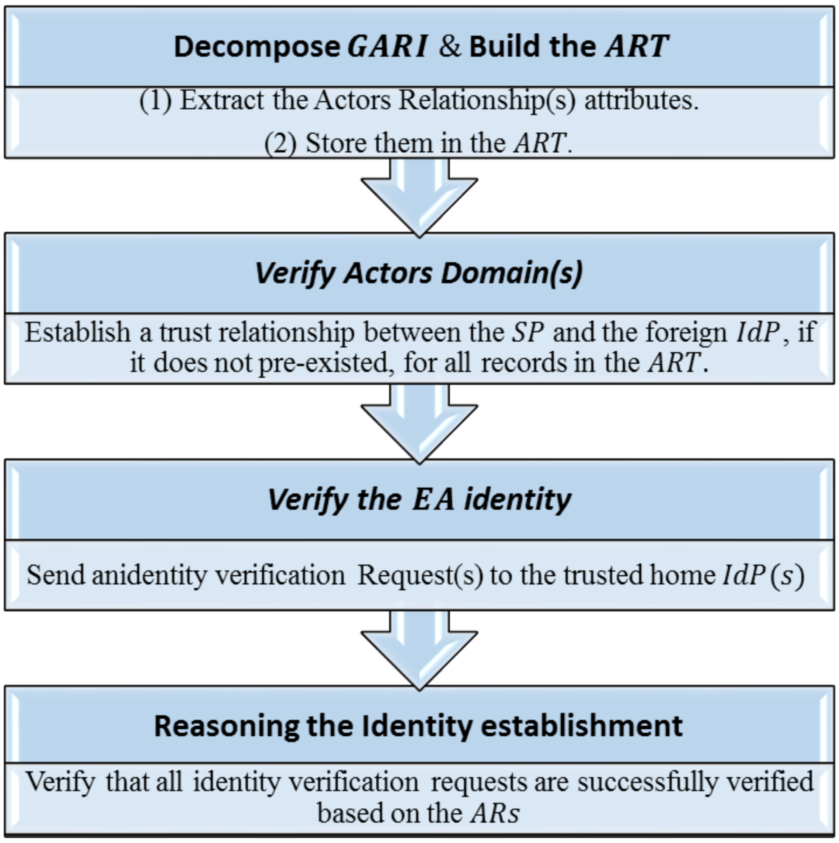

Decompose $GARI$ & Build the ART
(1) Extract the Actors Relationship(s) attributes.
(2) Store them in the ART.
Verify Actors Domain(s)
Establish a trust relationship between the SP and the foreign IdP, if it does not pre-existed, for all records in the ART.
Verify the EA identity
Send anidentity verification Request(s) to the trusted home $IdP(s)$
Reasoning the Identity establishment
Verify that all identity verification requests are successfully verified based on the ARs

Figure 4-2: An Overview of Identity Establishment System process at SPs

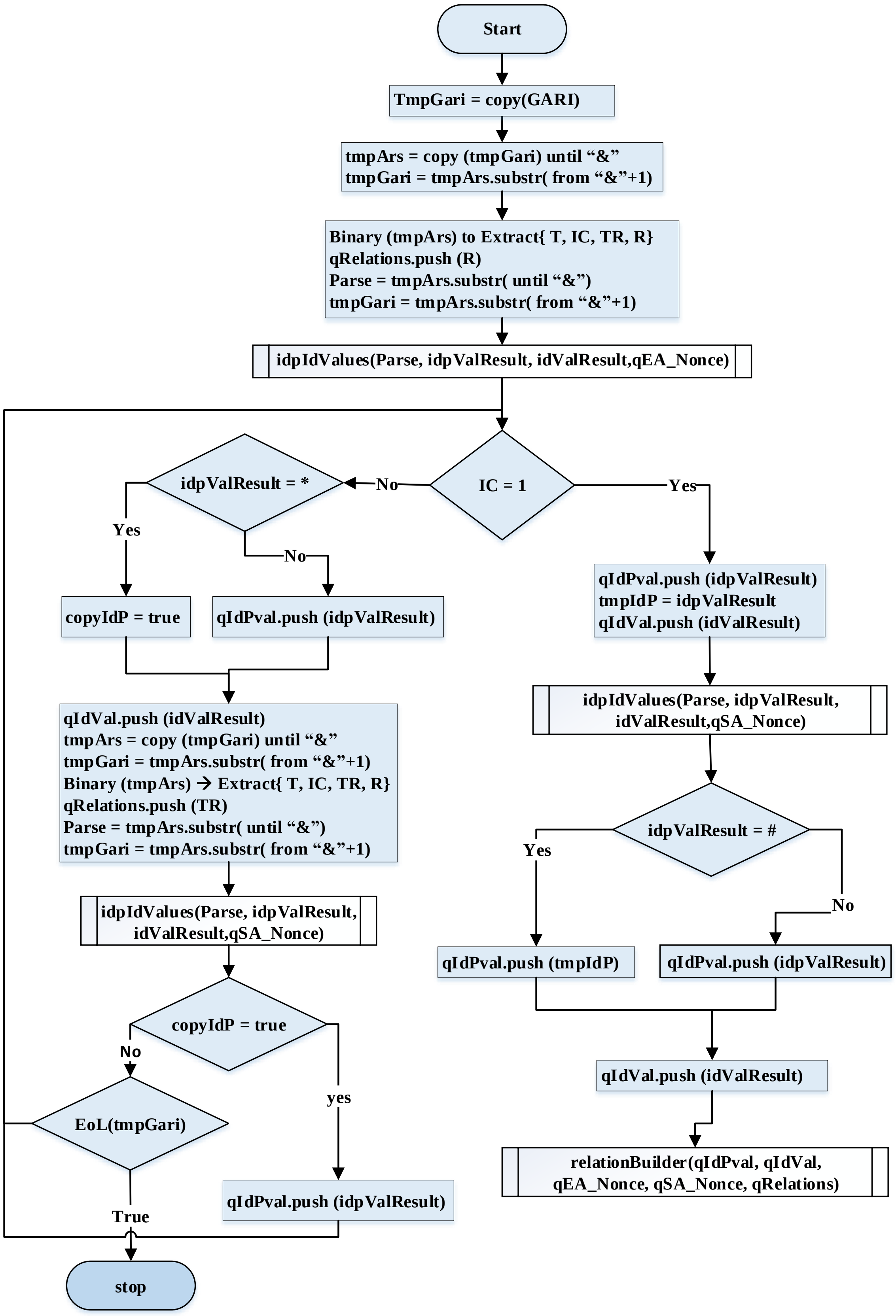

Start
TmpGari = copy(GARI)
tmpArs = copy (tmpGari) until "&"
tmpGari = tmpArs.substr(from "&"+1)
Binary (tmpArs) to Extract{ T, IC, TR, R}
qRelations.push (R)
Parse = tmpArs.substr(until "&")
tmpGari = tmpArs.substr(from "&"+1)
idpIdValues(Parse, idpValResult, idValResult,qEA_Nonce)
IC = 1
No
Yes
idpValResult = *
Yes
No
copyIdP = true
qIdPval.push (idpValResult)
qIdPval.push (idpValResult)
tmpIdP = idpValResult
qIdVal.push (idValResult)
qIdVal.push (idValResult)
tmpArs = copy (tmpGari) until "&"
tmpGari = tmpArs.substr(from "&"+1)
Binary (tmpArs) → Extract{ T, IC, TR, R}
qRelations.push (TR)
Parse = tmpArs.substr(until "&")
tmpGari = tmpArs.substr(from "&"+1)
idpIdValues(Parse, idpValResult, idValResult,qSA_Nonce)
idpIdValues(Parse, idpValResult, idValResult,qSA_Nonce)
idpValResult = #
Yes
No
qIdPval.push (tmpIdP)
qIdPval.push (idpValResult)
qIdVal.push (idValResult)
relationBuilder(qIdPval, qIdVal, qEA_Nonce, qSA_Nonce, qRelations)
copyIdP = true
No
yes
EoL(tmpGari)
qIdPval.push (idpValResult)
True
stop

Figure 4-3: *GARI* Decomposition Flowchart (stage 1)

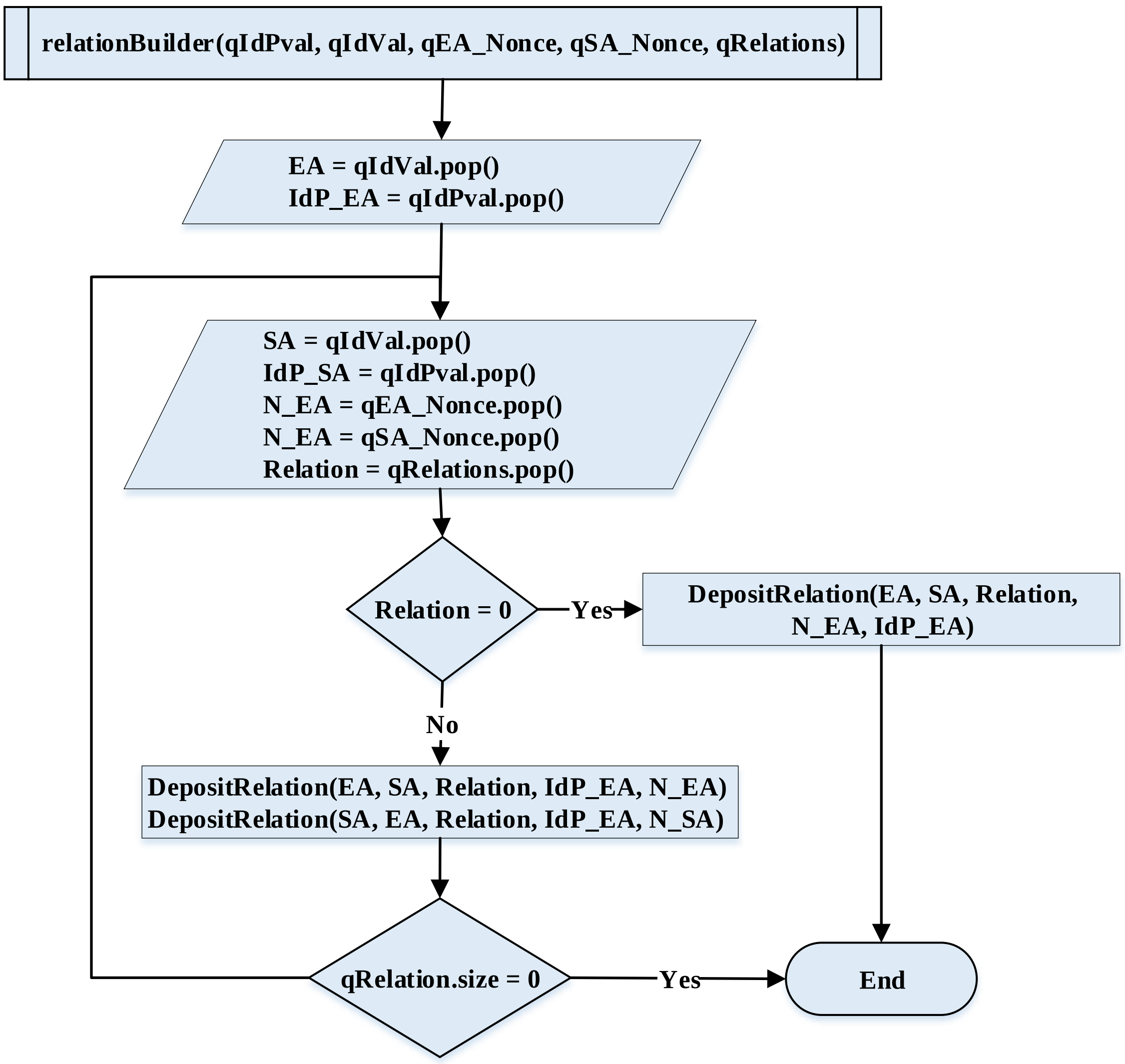

relationBuilder(qIdPval, qIdVal, qEA_Nonce, qSA_Nonce, qRelations)
EA = qIdVal.pop()
IdP_EA = qIdPval.pop()
SA = qIdVal.pop()
IdP_SA = qIdPval.pop()
N_EA = qEA_Nonce.pop()
N_EA = qSA_Nonce.pop()
Relation = qRelations.pop()
Relation = 0
Yes
DepositRelation(EA, SA, Relation, N_EA, IdP_EA)
No
DepositRelation(EA, SA, Relation, IdP_EA, N_EA)
DepositRelation(SA, EA, Relation, IdP_EA, N_SA)
qRelation.size = 0
Yes
End

Figure 4-4:$GARI$ Decomposition Flowchart (stage 2)

4.2.4.1. *Decompose the $GARI$* and Build the ART

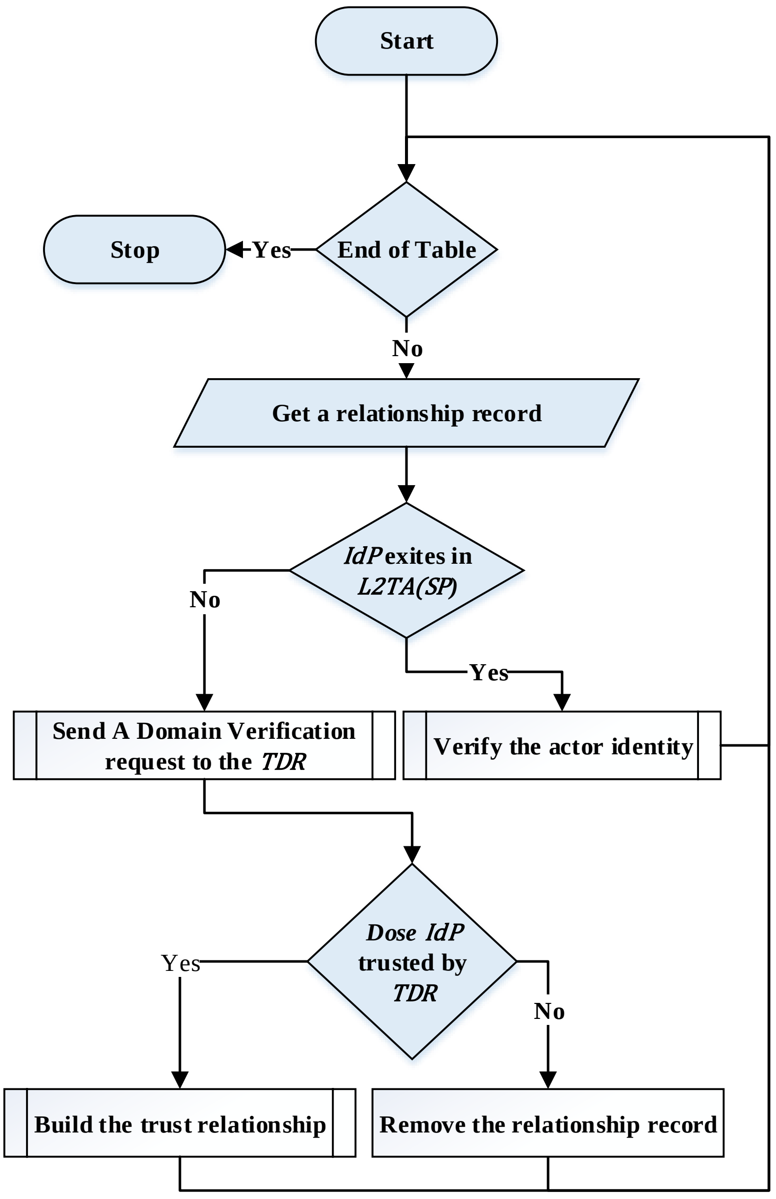
To establish the requester identity, the SP decomposes the received $GARI$ to extract the participated actors identities and their relationship attributes. The resulted data for this stage will be inserted in five queues as follows; a qRelations queue to store the actor relationship types; a qIdVal queue to store the actors identifiers; a qIdPval queue to store the identity provider identifiers for the actors identifiers, a qEA_Nonce queue to store the EA relationship nonce(s) with other actors in the IdP of the EA, and a qSA_Nonce queue to store the SA relationship nonces with the EA at the IdP of the SA. Figure 4-3 illustrates the flowchart to get these data form the received $GARI$.

In the next stage, these relationship(s) attributes are used to build the ART for each request that will be used later for the identity verification process. Based on the type of relationship, each actors relationship will be represented relationship by single or two records as follows.

Each relationship of (semi-)permanent type could be represented by two records in the ART. Each record contains the following attributes in addition to other actor attributes; for instance, the relationship $(AR_{a,b})$ will be represented in the ART as follows:

- $R1 = \left\{ID_a, ID_b, ID_{IdP_a}, T(AR_{a,b}), h\left(N^{a}_{AR_{a,b}}\right), \dots\right\}$
- $R2 = \left\{ID_b, ID_a, ID_{IdP_b}, T(AR_{a,b}), h\left(N^{b}_{AR_{a,b}}\right), \dots\right\}$

However, the relationship of free type will be represented by a single record in the ART which is the $R1$. The ART is implemented as a table within the SQLight3 database as seen by Figure 4-4.



Start
End of Table
Yes
Stop
No
Get a relationship record
IdP exites in L2TA(SP)
No
Yes
Send A Domain Verification request to the TDR
Verify the actor identity
Dose IdP trusted by TDR
Yes
No
Build the trust relationship
Remove the relationship record

Figure 4-5: Verifying the Actors Domain Flowchart

4.2.4.2. Verify Actors Domain(s)

The next step to building the ART is to verify each IdP in the actor relationship listed in the table. Figure 4-5 illustrates the flowchart to verify the actor domain. This process will be done by checking whether the SP has already a trust relationship with the IdP that manages the actor identity in its $L2TA(SP)$. If it exists then move to the next stage; otherwise, the SP will be interconnected with a trusted TDR to verify the foreign IdP to build the required trust relationship. If it is trusted by a TDR then the process to build the trust relationship with the IdP will be started. Otherwise, that record will be removed, which is not implemented in this research because the TDR processes of managing entities trust is not covered.

4.2.4.3. Verify the EA Identity based on $AR(s)$

The SP, next, interconnects with the domains $IdPs$ to verify the actor identity using the relations in the ART.

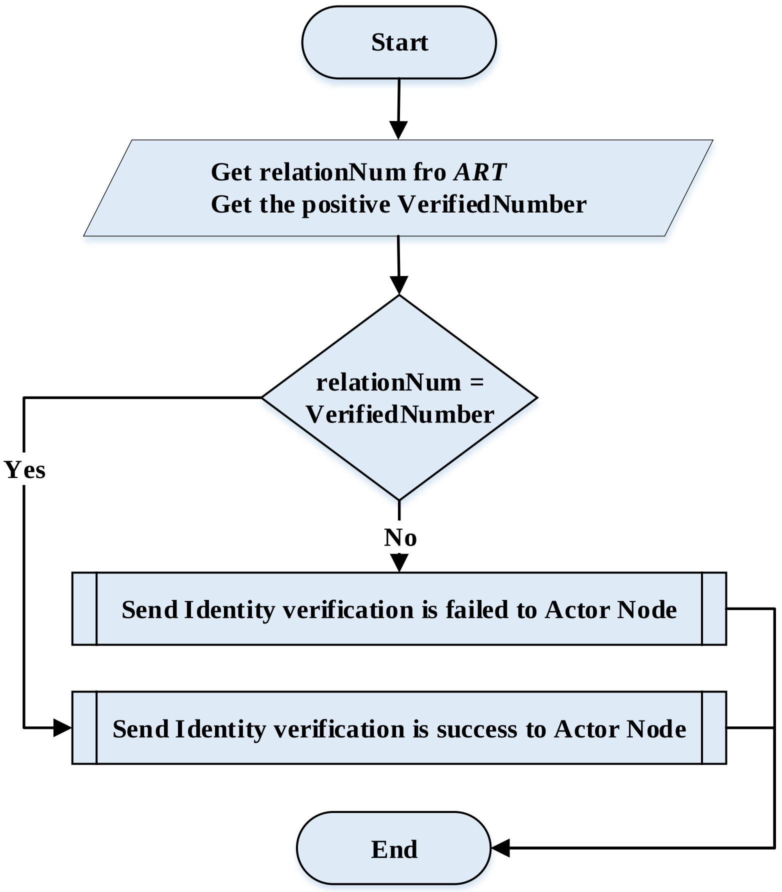

Start
Get relationNum fro ART
Get the positive VerifiedNumber
relationNum = VerifiedNumber
Yes
No
Send Identity verification is failed to Actor Node
Send Identity verification is success to Actor Node
End

Figure 4-6: A Flowchart of Reasoning the Identity Establishment

4.2.4.4. Reasoning the Identity Establishment

The last step in the process of identifying the EA identity is reasoning the identity establishment. In this step, SP checks the replies of all the identity verification requests that were sent to the $IdP(s)$ in previous step within a period of time. If they are verified by those $IdP(s)$, then the identity will be established successfully, otherwise, it is failed. Figure 4-6 illustrates the flowchart of reasoning the identity establishment.

4.3. Global Identity Verification ($GIdV$) Protocol for IoT

As discussed in previous chapters, the $GIdM$ requires an effective identity verification protocol to support the establishment of the EA identity by a visited service provider SP_v at any domain. Thus, the $GIdV$ is designed to be deployed by the $GIdM$ by components to support SP_v in the general IoT use-case, presented in Section 3.1 above, to establish the EA identity. In such scenarios, the SP_v could uses the $GARI$ information, as depicted in chapter 3, to verify the AE identity. The SP_v starts by decomposing the $GARI$ information to get the actors identification attributes and the relationship(s) types between these actors. It is worth to note that two types of actor's interactions could be extracted from the $GARI$. These types are direct interaction and transitive interaction. To explain them, let us assume that the $GARI$ represents a relationships between three actors involved in two relationships, e.g. $AR_{a,b}$ and $AR_{b,c}$. The EA in the $GARI$ is the actor (a), while the communication devices are (b) and (c). The SP_v could decide which $IdP(s)$ could be involved in the identity establishment process based on these types of interaction as explained in detail below.

- ***Direct interaction***: It represents a direct relationship between the EA, i.e. (a), and the commination device (a). Therefore, if $T\left(AR_{a,b}\right)$ is of (semi-)permanent relationship, then both IdP_a and IdP_b will participate in the process. Otherwise, only the IdP_a could be used to establish the AE identity because in such a relationship type, IdP_b does not have a record of the user(s).
- ***Transitive interaction***: It represents the relationship between the EA, i.e. (a), and the communication object (c), which is represented by , i.e. (TR) in the ARS. If $T\left(AR_{a,c}\right)$ is of (semi-)permanent relationship, then the TR exists and both the IdP_a and IdP_c will be involved in the process; otherwise none will be involved.

The protocol uses the following identification factors to verify the actor's identity. *Firstly*, the relationship between two actors. *Secondly,* a one-way hash function of a secret nonce for this relationship in the *home* IdP. To apply these factors, the SP_v requires having a pre-established trust relationship with each domain's IdP that manages the actor identity in the relationship to authenticate it.

Applying the proposed identity verification protocol with the general actor use-case interaction in IoT requires two main phases. *Firstly*, establishing a trust relationship between SP_v and the home IdP of each actor (IdP_x) in the relationship; and sharing the SP_v's public key with IdP_x and vice versa; *secondly*, verifying the AE based on its relationship(s) with the communicated object(s). It is worth to note that the first phase is required only in the case where the SP_v does not have a pre-established trust relationship with the foreign IdP that manages the actor's identity. Otherwise both should be followed in sequence.

The identity verification processes start when the SP_v receives the requester object identification, i.e. the $GARI$. The next step is, SP_v extracts the relationship(s) from $GARI$, and builds the ART for this service request. After that, SP_v checks which phase to be followed, i.e. *phase 1* or *phase 2*. Needham-Schroder-Lowe (NSL) public key infrastructure [121] is proposed to establish the required trust relationships because the efficiency of PKI method has been approved in literature like in [115], [122], [123] for the IoT environment.

The protocol builds the trust relationship dynamically between SP_v and IdP_x based on distributed trusted 3rd parties called Trusted Domains Registries ($TDRs$) and the mutual authentication between them. It is assumed that all domain entities, (SP and IdP) are registered as agents with a TDR. The TDR should answer the requests from their agents about foreign agents. If the TDR does not have this information, it multicasts the request to other $TDRs$ within its multicast domain until the agent's trusted information is found or the request hop limit is expired. It is proposed that the messages will follow the DNS messages in terms of the messages repetition, time interval, and hop limits. When such a trust relationship is established, the SP_v can rely on IdP_x to authenticate the EA identity. The proposed protocol for this phase is composed of two phases.

Phase 1: Build a trust relationship and share secret keys between SP_v and IdP_x

This phase starts when an SP_v receives a service request from an IoT's object. It receives the $GARI$ identifier with the request. To allow the SP_v to identify the EA, SP_v needs to establish trust relationships with each of the IdP_x available in the $GARI$. Figure 4-7, illustrates this phase's steps.

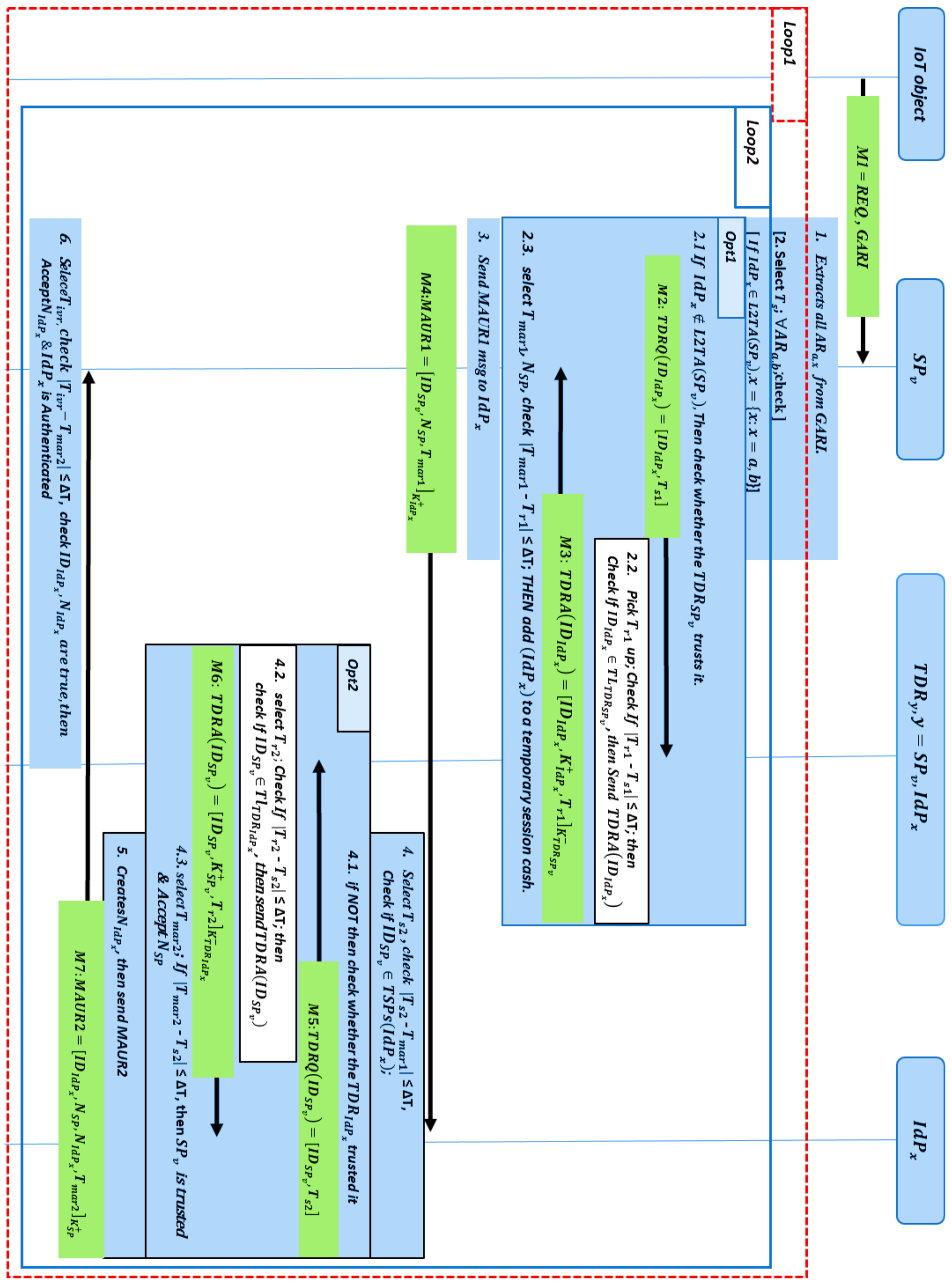

IoT object
SP_v
$TDR_y, y = SP_v, IdP_x$
IdP_x
Loop1
Loop2
Opt1
Opt2
M1 = REQ, GARI
1. Extracts all $AR_{a,x}$ from GARI.
[2. Select T_s; $\forall AR_{(a,b)}$;check]
[If $IdP_x \in L2TA(SP_v), x = \{x: x = a, b\}$]
2.1 If $IdP_x \notin L2TA(SP_v)$, Then check whether the TDR_{SP_v} trusts it.
M2: $TDRQ(ID_{IdP_x}) = [ID_{IdP_x}, T_{s1}]$
2.2. Pick T_{r1} up; Check If $|T_{r1} - T_{s1}| \leq \Delta T$; then Check if $ID_{IdP_x} \in TL_{TDR_{SP_v}}$, then Send $TDRA(ID_{IdP_x})$
M3: $TDRA(ID_{IdP_x}) = [ID_{IdP_x}, K^+_{IdP_x}, T_{r1}]_{K^-_{TDR_{SP_v}}}$
2.3. select T_{mar1}, N_{SP}, check $|T_{mar1} - T_{r1}| \leq \Delta T$; THEN add (IdP_x) to a temporary session cash.
3. Send MAUR1 msg to IdP_x
M4:MAUR1 = $[ID_{SP_v}, N_{SP}, T_{mar1}]_{K^+_{IdP_x}}$
4. Select T_{s2}, check $|T_{s2} - T_{mar1}| \leq \Delta T$, Check if $ID_{SP_v} \in TSPs(IdP_x)$;
4.1. if NOT then check whether the TDR_{IdP_x} trusted it
M5:$TDRQ(ID_{SP_v}) = [ID_{SP_v}, T_{s2}]$
4.2. select T_{r2}; Check If $|T_{r2} - T_{s2}| \leq \Delta T$; then check if $ID_{SP_v} \in TL_{TDR_{IdP_x}}$, then send $TDRA(ID_{SP_v})$
M6: $TDRA(ID_{SP_v}) = [ID_{SP_v}, K^+_{SP_v}, T_{r2}]_{K^-_{TDR_{IdP_x}}}$
4.3. select T_{mar2}; If $|T_{mar2} - T_{s2}| \leq \Delta T$, then SP_v is trusted & Accept N_{SP}
5. Creates N_{IdP_x}, then send MAUR2
M7:MAUR2 = $[ID_{IdP_x}, N_{SP}, N_{IdP_x}, T_{mar2}]_{K^+_{SP}}$
6. SelecT T_{ivr}, check $|T_{ivr} - T_{mar2}| \leq \Delta T$, check ID_{IdP_x}, N_{IdP_x} are true, then Accept N_{IdP_x} & IdP_x is Authenticated

Figure 4-7: Trust Relationship Building and Secret Key Sharing

1. The user requests a service from an SP_v and presents its identity using $GARI$ identifier.
2. The SP_v decomposes the received $GARI$ and extracts the relationship(s) attributes, i.e. the number of actor relationship(s), the relationship type(s), the communication device type(s), the transitive relationship(s), and the identification information for each actor to build the ART. SP_v selects a timestamp T_{s1} to prove the message freshness. After that, for each *relationship* with (a) as the EA, SP_v checks the existence of trust relationship(s) with the IdP_x of each actor (x) for each $AR_{a,b}$, $IdP_x, x = \{x: x = a, b\}$; in its $L2TA(SP_v)$. If $IdP_x \in L2TA(SP_v)$ is not valid, then the following *optional* steps (2.1 – 2.3) are triggered.

2.1. If $IdP_x \notin L2TA(SP_v)$; SP_v inquires with the TDR_{SP_v}, the one that SP_v is registered with, whether it trusts IdP_x by sending the following $TDRQ$ message.

$$TDRQ\left(ID_{IdP_x}\right) = [ID_{IdP_x}, T_{s1}]$$

2.2. TDR_{SP_v} receives the $TDRQ$ message, picks up a timestamp T_{r1} and checks it was received within an acceptable time delay by checking if $|T_{r1}$ - $T_{s1}| \leq \Delta$T. If so, *then* it checks the existence of IdP_x in its trusted list $(TL_{TDR_{SP_v}})$ as follows:

- If $ID_{IdP_x} \in TL_{TDR_{SP_v}}$, *then* IdP_x is trusted.
- Otherwise, it multicasts the query to its partners in TDR layer until the information is found or exceeds the hop's limit.

Finally, TDR_{SP_v} replies SP_v with the answer message ($TDRA$), and binds the necessary information with the timestamp (T_{r1}) in the message and signs it with TDR_{SP_v} private key as follows:

$$TDRA\left(ID_{IdP_x}\right) = [ID_{IdP_x}, K^{+}_{IdP_x}, T_{r1}]_{K^{-}_{TDR_{SP_v}}}$$

2.3. If SP_v receives the $TDRA(IdP_x)$, within a ΔT, $|T_{r1}$ - $T_{s1}| \leq \Delta$T, *then add* (IdP_x) *to a temporary sessions cash.* At this point, the SP_v trusts IdP_x based on a trusted 3rd party, i.e. TDR_{SP_v}, gets its public key$K^{+}_{IdP_x}$ and is ready to start the mutual authentication process with SP_v.

3. SP_v creates a fresh secret (nonce), i.e. N_{SP_v}, and binds it with its identifier, i.e. ID_{SP_v} and the timestamp T_{mar1}. SP_v sends a *mutual authentication request* message *(MAUR1)* to IdP_x encrypted under $K^{+}_{IdP_x}$. The $MAUR1$ contains this information as follows:

$$MAUR1 = \left[ID_{SP_v}, N_{SP}, T_{mar1}\right]_{K^{+}_{IdP_x}}$$

4. IdP_x receives the $MAUR1$, decrypts it to read the information, selects the current timestamp T_{s2} and checks whether it received the message within an acceptable time, i.e. $|T_{s1}$ - $T_{mar1}|$ $\leq \Delta$T. If the verification succeeds, then it checks its trusted SPs list to check whether it deals with a trusted one. If $ID_{SP_v} \in TSPs_{IdP_x}$ is not valid, then the following *optional* steps (4.1 – 4.3) is triggered.

4.1. If $ID_{SP_v} \notin TSPs_{IdP_x}$, it inquiries with the TDR_{IdP_x}, the one IdP_x is registered with, whether it trusts the SP_v as follows:

$$TDRQ(ID_{SP_v}) = [ID_{SP_v}, T_{s2}]$$

4.2. When TDR_{IdP_x} receives the $TDRQ$ message, it picks up the current timestamp T_{r2}, then checks it was received within time, i.e. if $|T_{r2}$ - $T_{s2}| \leq \Delta$T. If so *then,* it checks

- If $ID_{SP_v} \in TDR_{IdP_x}$, *then* SP_v is trusted by TDR_{IdP_x}.
- Otherwise, it multicasts the query to its partners in *TDR layer* until the information is found or exceeds the hop’s limit.

Finally, TDR_{IdP_x} replies to SP_v with the answer message(s) ($TDRA$) and the necessary information and timestamp (T_{r2}). TDR_{IdP_x}signs the $TDRA$ as follows:

$$TDRA(ID_{SP_v}) = [ID_{SP_v}, K^+_{SP_v}, T_{r2}]_{K^-_{TDR_{IdP_x}}}$$

4.3. When IdP_x receives the $TDRA$ message, it picks up the current timestamp T_{mar2}, checks if $|T_{mar2}$ - $T_{r2}| \leq \Delta$T is valid, then IdP_x trusts SP_v. At this point, the IdP_x trusts SP_v based on a trusted 3rd party agent, i.e. TDR_{IdP_x}, and gets the public key of SP_v and ready for the next phase.

5. IdP_x creates a fresh (nonce), i.e. N_{IdP_x}, it binds the secret with the received N_{SP} and its identifier (ID_{IdP_x}) to challenge SP_v. The timestamp T_{mar2} is added as well to compose the second *mutual authentication request* message *(MAUR2)* and sends to SP_v. *MAUR2* is encrypted under k_{SP_v} as follows:

$$MAUR2 = [ID_{IdP_x}, N_{SP}, N_{IdP_x}, T_{mar2}]_{K^+_{SP_v}}$$

6. When SP_v receives the $MAUR2$, it uses its private key to read the message. It starts by checking the delay time, i.e. $|T_c$ - $T_{mar1}| \leq \Delta$T, if so, it maps the received ID_{IdP_x} and N_{SP} with the destination identifier and the secret key used to generate $MAUR1$ or not. If they are mapped then SP_v being confident that it is dealing with the right principles, i.e. IdP_x, accepts using N_{IdP_x} as a shared secret to be used in future messages.

At this stage a mutual authentication is satisfied between the SP_v and IdP_x. A trust relationship is built based on distributed trusted 3rd parties, i.e. $TDR_{SP_v}$and TDR_{IdP_x}. The next stage is sharing a secret key between the SP_v and IdP_x before starting the actor's authentication process.

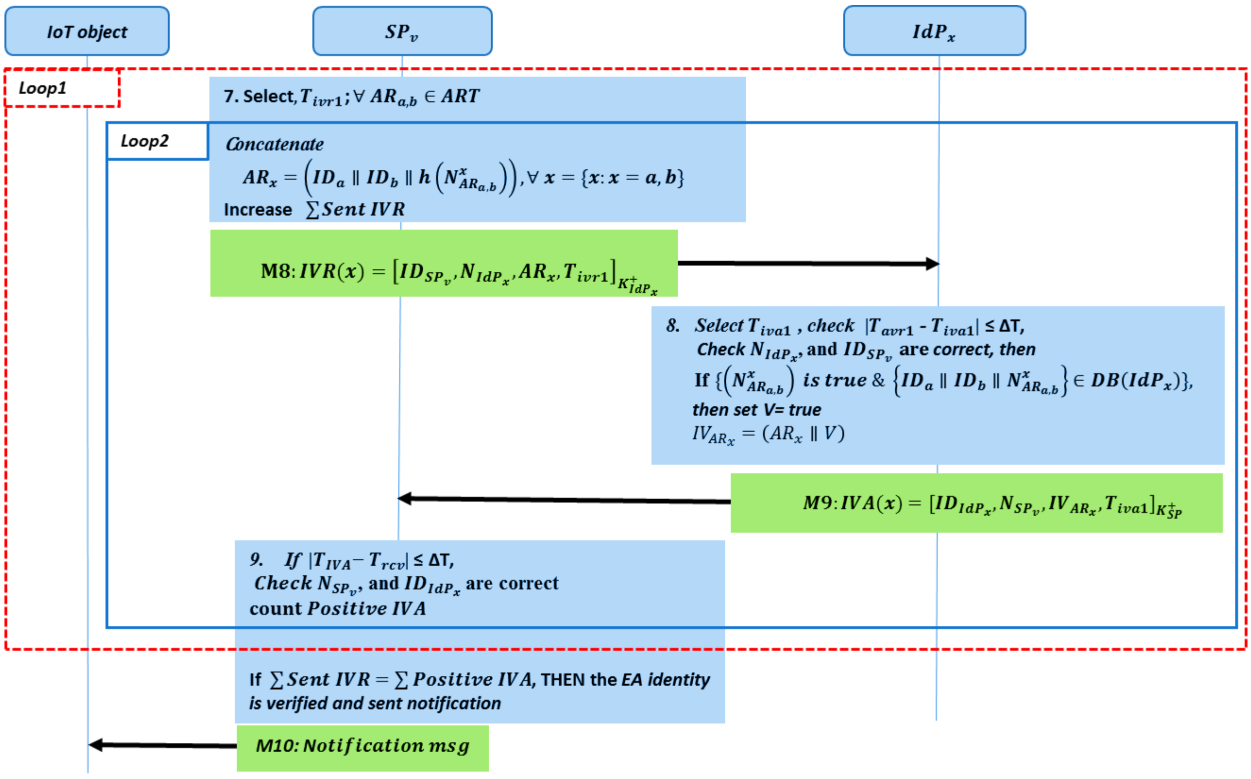

IoT object
SP_v
IdP_x
Loop1
Loop2
7. Select, T_{ivr1}; $\forall AR_{a,b} \in ART$
Concatenate
$AR_x = \left(ID_a \parallel ID_b \parallel h\left(N^x_{AR_{a,b}}\right)\right), \forall x = \{x: x = a, b\}$
Increase $\sum Sent\ IVR$
M8: $IVR(x) = [ID_{SP_v}, N_{IdP_x}, AR_x, T_{ivr1}]_{K^+_{IdP_x}}$
8. Select T_{iva1}, check $|T_{avr1} - T_{iva1}| \leq \Delta T$,
Check N_{IdP_x}, and ID_{SP_v} are correct, then
If $\{\left(N^x_{AR_{a,b}}\right)$ is true & $\left\{ID_a \parallel ID_b \parallel N^x_{AR_{a,b}}\right\} \in DB(IdP_x)\}$,
then set V= true
$IV_{AR_x} = (AR_x \parallel V)$
M9: $IVA(x) = [ID_{IdP_x}, N_{SP_v}, IV_{AR_x}, T_{iva1}]_{K^+_{SP}}$
9. If $|T_{IVA} - T_{rcv}| \leq \Delta T$,
Check N_{SP_v}, and ID_{IdP_x} are correct
count $Positive\ IVA$
If $\sum Sent\ IVR = \sum Positive\ IVA$, THEN the EA identity is verified and sent notification
M10: $Notification\ msg$

Figure 4-8: The EA Identity Verification Phase

Phase 2: Verify the EA Identity

In this stage, SP_v uses the trust relationship(s) and its secret key with IdP_x to perform the EA identity authentication. The SP_v sends the *IVR message* to the trusted IdP_x in the ART, which is involved in the identity verification process. Finally, SP_v compares the number of sending *IVR message* with the number of positive answers. If the comparison is successfully passed then, (a) identity is authenticated as EA of the communication based on its relationship(s) with the IoT object(s). Otherwise, the identity will not be authenticated. Figure 4-8 shows the following steps of this stage.

7. For each $AR_{a,x}$ in the ART, one IVR message inquires being sent to the domain IdP. In other words, for the IdP_x of (x) in $AR_{a,b}$, SP_v concatenates $AR_x = \left(ID_a \parallel ID_b \parallel h\left(N^x_{AR_{a,b}}\right)\right), x = \{x: x = a, b\}$. For each IVR message, the AR_x is concatenated with ID_{SP_v}, one-way hash

function of N_{IdP_x}, and the timestamp T_{ivr1}. The *IVR* message is encrypted under $K^{+}_{IdP_x}$ as follows:

$$IVR = \left[ID_{SP_v}, N_{IdP_x}, AR_x, T_{ivr1}\right]_{K^{+}_{IdP_x}} .$$

Then SP_v sends *IVR* to IdP_x. The total number of sent messages will count as well.

8. When the IdP_x receives the IVR message, it picks the current timestamp T_{iva1} up and uses its private key to decrypt the message. It checks whether the IVR was received within an acceptable time, i.e. $|T_{iva1}$ - $T_{ivr1}| \leq \Delta$T. If the time is verified, then it checks the correctness of the IdP_x secret key, i.e. N_{IdP_x}, and *SP* identifier, i.e. ID_{SP_v}. If all checks are verified successfully, then IdP_x proceeds with the identity verification process by mapping the relationship participant's identifiers and the relationship nonce with those available in the registration records, i.e. $DB(IdP_x)$, as follows:

$$If\ \left\{\left(N^{x}_{AR_{a,b}}\right) is\ true\ \&\ \left\{ID_a \parallel ID_b \parallel N^{x}_{AR_{a,b}}\right\} \in DB(IdP_x)\right\}; then\ (a)\ is\ authenticated.$$

If the checking is verified, then IdP_x sets $V = true$ and concatenates it with AR_x to represent the verification result, i.e. $IV_{AR_x} = (AR_x \parallel V)$. The IdP_x composes the identity verification answer, *IVA*, and encrypts it with K^{+}_{SP}. Finally, IdP_x sends the *IVA* to SP_v.

$$IVA = [ID_{IdP_x}, N_{SP_v}, IV_{AR_x}, T_{iva1}]_{K^{+}_{SP}}.$$

9. SP_v selects a timestamp T_{rcv1}, and decrypts the received messages using its private key. Then it checks that all IVA messages are received within a predefined delay, i.e. $|T_{rcv1}$- $T_{iva1}| \leq \Delta$T. It maps the secret key, N_{SP_v}, and the ID_{IdP_x} with the IdP_x session details. SP_v counts the number of positive replies, *V*. After that it compares with the number of *IVRs*, as follows:

$$If\ \sum sent\ IVR = \sum positive\ IVA,\ Then\ \text{the } EA \text{ identity is established.}$$

A further processes of authorization and access control are applied to acknowledge the request; otherwise, the identity establishment fail and a message send back to the requester.

4.4. Global Identity Verification Protocol ($GIdM$) Messages

This section presents the proposed protocol messages design to be used by in the $GIdM$ to establish the requester, i.e. EA, identity. These messages can be classified into three types based on their role as follows:

- ***GARI exchange***, which is used by the communication devices to exchange the $GARI$ with a SP at a service request time and to receive the identity establishment notification.

- ***Verify the visitor actors' domain registration***, which is used by the SPs and the $TDRs$ to verify the IdP that manages the visitor actor identity, and get the result.
- ***Verify the actor identity by home IdP,*** which is used by the SP to interconnect the actor home $IdP(s)$ to verify the identity based on its relationship(s) and get the result.

These protocols have to be compatible with IPv6 as it is considered as the backbone network protocol for the IoT [124]–[126]. This is because IPv6 has characteristics like neighbours and service discovery that allows an object to be aware of its surroundings and mobility support by allowing objects to keep its IP-address through moving to another domain. More details can be found in Appendix E.

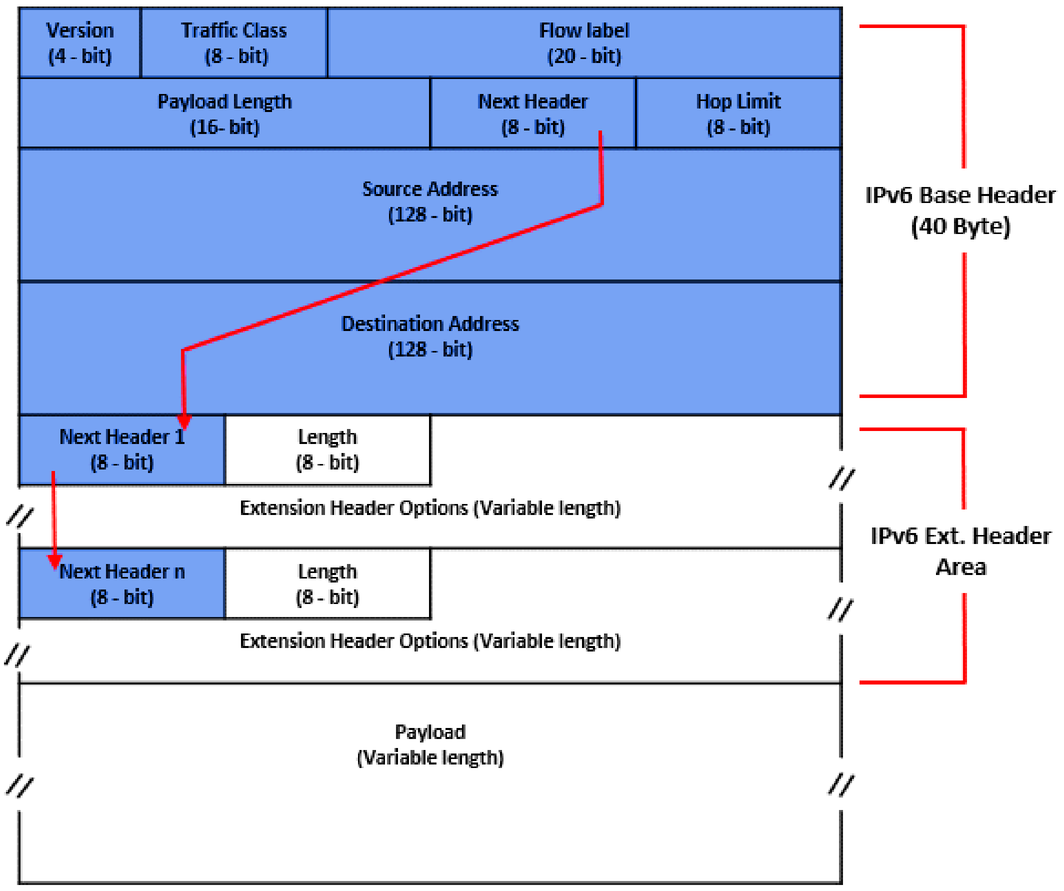


Figure 4-9: IPv6 Packet Format, Adapted from [127]

IPv6 packet headers are composed of base headers with a fixed size (40 byte) and optional extension headers with a fixable length, as depicted in Figure 4-9. Ipv6 extension headers are located between the base headers and the transport layer header in a packet. The main characteristic of them is they are not examined by routers along the traffic path unless it is necessary to forward the packet such as in a hop-by-hop option [127]. Therefore, the IPv6 extension headers area will be used to host the identification information throughout the path to the destination by adding new headers. The new headers will be named as $GARI$ Extension Header ($GARI\ Ext.Hdr.$), Visitor's Registration Domain Header ($VDR\ Ext.Hdr.$), and Identity Verification Header ($IV\ Ext.Hdr.$).

Table 4-6 illustrates the proposed extension headers codes. By considering the guidelines for defining new extension headers in RFC2460 [127] and RFC6564 [128], the headers format will be designed and discussed in the following paragraphs.

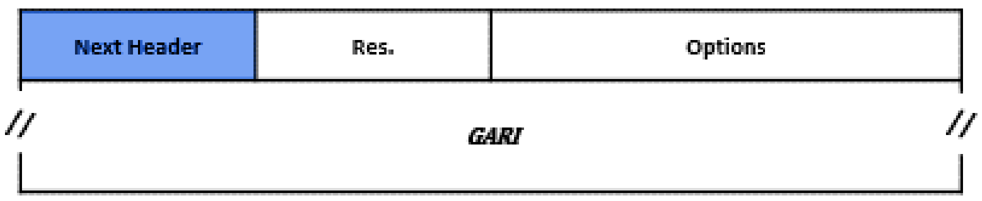
Table 4-6: The Proposed Extension headers Codes

Extensions Header	**Representation Code**
$\boldsymbol{GARI\ Ext.\ Hdr.}$	75
$\boldsymbol{VDR\ Ext.\ Hdr.}$	70
$\boldsymbol{IV\ Ext.\ Hdr.}$	71

4.4.1. The $GARI$ Exchange Message

This message is used by the communicated device for the purpose of transmitting the identifier to the service provider. According to [127], the maximum extension header size is ($2^8 - 1 = 255$ byte, exclude the next header field). However, the $GARI$ has a variable length according to the attributes size of participating actors that may exceed the extension header size. Thus, the ($GARI\ Ext.Hdr.$) format will follows the same format on the ($Fragment\ Ext.Hdr.$). In other words, the header size will be declared by the payload field of the IPv6 header itself. It will be represented in the Next Header Code using the value (75). Figure 4-10 illustrates the proposed format.

This format could be used by the communicated devices and the SP to share the composed identifier, i.e. $GARI$ with the SP and to receive a successful identify establishment notification. The *Options* field will be used to differentiate the meaning of the message. Submitting the $GARI$ for identity establishment by the SP will be represented by the options (Query/Answer = 1 & Answer = 0). The answer message from the SP will be represented in two forms: first (Query/Answer = 0 & Answer = 1) which means the identity successfully established, while (Query/Answer = 0 & Answer = 0) means the identity failed to establish.



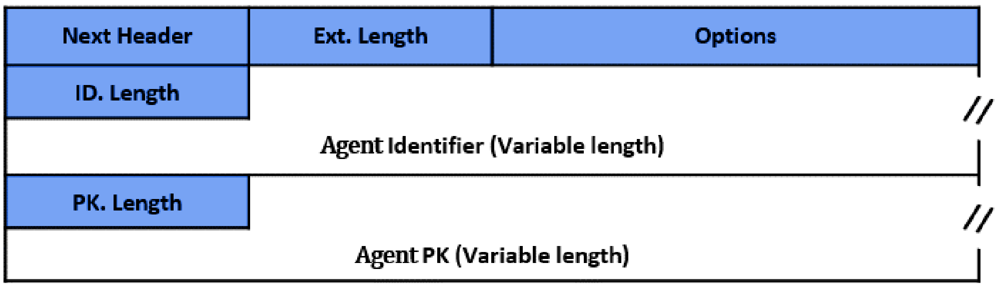
Field	Description
Next Header	Specify the next header code in 8 – bit.
Res.	Reserved field in 8 – bit. By default is set by "0" and is ignored by received node
Options	16 – bits represents the message options as follows: • 1 – bit Query/Answer message type (1: Q; 0: A) • 1 – bit Answer Type (1: Identified; 0: un-identified) • 2 – bit Reserved • 12 – bit Sequence number.
$\boldsymbol{GARI}$	The Identifier that represents the actors' relationship(s).

Figure 4-10: $GARI\ Ext.Hdr.$ Format

4.4.2. Visitor Actors' Domain Registration $\boldsymbol{VDR}$ Verification Messages

These messages are used by SPs and $TDRs$ in order to verify the $IdP(s)$ that issued the actor's identifier. Figure 4-11 illustrates the $(VDR\ Ext.Hdr.)$ format. This format follows the standard notices for defining a new IPv6 extension header stated in RFC2460. It will be represented in the Next Header Code using the value (70) in the IPv6 base header. The SP uses their format to interconnect with its trusted TDR as seen by $TDRQ$ message and its answer $TDRA$ in Section 4.3. Moreover, the $TDRs$ nodes use the same message format to interconnect with each other for the same purpose. VDR messages carry variable length data, hence the final length is variable in size with a limit up to (255) excluding the next header field.

This message format starts with the extension header length in bytes excluding the first byte. The message options are used to represent the type of the message, the result type of this message, and a message sequence number. Next, the format has two variable length fields to carry the IdP identifier and its PKvalue with a delimiter in front of each field. The PKvalue is set to zeros by the requester initially, then updated with the found value if the domain is trusted by the TDR. The verification request message will be represented by using options (Query/Answer = 1 & Answer = 0).The answer message from the TDR will be represented in two forms: first (Query/Answer = 0 & Answer = 1) which means the agent is trusted, while (Query/Answer = 0 & Answer = 0) means the agent is not trusted or unfound. As stated in previous section the VDR messages follow the DNS notation in terms of messages repetition, time interval, and hop limits.


Next Header
Ext. Length
Options
ID. Length
Agent Identifier (Variable length)
PK. Length
Agent PK (Variable length)

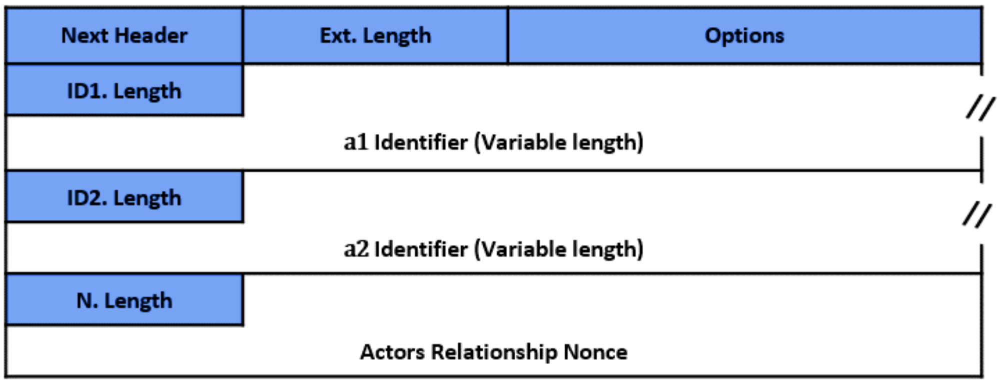
Field	Description
Next Header	Specify the next header code in 8 – bit.
Ext. Length	Specify headers length in bytes, excluding the first byte.
Options	16 – bits represents the message options as follows: • 1 – bit Query/Answer message type (Q: 1; A: 0) • 1 – bit Answer Type (Trusted:1; un-Trusted: 0) • 2 – bit Reserved • 12 – bit Sequence number.
ID. Length	Specify the IdP identifier length in bytes.
IdP Identifier	The required IdP identifier to verify
PK Length	Specify the PK length of the IdP in bytes.
The PK	The required PK of the dP, which is set to zeros if $Q/A = 1$ or Answer is un-Trusted.

Figure 4-11: $VDR\ Ext.Hdr.$ Format

4.4.3. Actor's Identity Verification (IV) Messages

The SPs in $GIdM$ use the IV message format to request to verify the actor identity by its domain IdP. Similar to VDR messages, the IV messages follow the standard IPv6 extension format. It is proposed to use (71) as message code at Next Header code. SP uses the IV format to request the actor's identity verification from the home IdP by transmitting the actors identities, the relationship nonce, and the relationship type using the Identity verification Request (IVR) message. The IdP in turn uses the same format to send back the verification result using Identity verification Answer (IVA) message.

As shown in Figure 4-12, the header extension length is stated by the extension header field, excluding the next header field. The message options area contains the following data: the message type, i.e. Request or Answer; the result type of this message; the actor relationship type; and a message sequence number. Next, the format has three variable length fields to carry the first actor identifier in the relationship, the second actor identifier in the relationship, and the relationship nonce in the targeted IdP with a delimiter in front of each field. The verification request message will be represented by using options (Query/Answer = 1 & Answer = 0). The answer message from the TDR will be represented in two forms: first (Query/Answer = 0 & Answer = 1) which means the agent is trusted, while (Query/Answer = 0 & Answer = 0) means the agent is not trusted or unfound. The sequence number will be used by both the sender and receiver as a reference for the message.


Next Header
Ext. Length
Options
ID1. Length
a1 Identifier (Variable length)
ID2. Length
a2 Identifier (Variable length)
N. Length
Actors Relationship Nonce

Field	**Description**
Next Header	Specify the next header code in 8 – bit.
Ext. Length	Specify headers length in bytes, excluding the first byte.
Options	16 – bits represents the message options as follows: • 1 – bit Request/Answer message type (Q: 1; A: 0) • 1 – bit Answer Type (Verified:1; Un-Verified: 0) • 2 – bit Actor Relationship Type • 12 – bit Sequence number.
ID*1 *Length	Specify the ($a1$) identifier length in bytes.
***a*1 Identifier**	The actor ($a1$) identifier in $AR_{a1,a2}$
ID*2 *Length	Specify the $a2$ identifier length in bytes.
***a*2 Identifier**	The actor $a2$ identifier in $AR_{a1,a2}$
N. Length	Specify the relationship nonce length in bytes.
Actor Relationship Nonce	The actor relationship Nonce $N_{AR_{a1,a2}}$

Figure 4-12: IV Message Format

4.5. Summary

This chapter presented the $GIdM$ system to satisfy the identity establishment requirements in the IoT environment. The new architecture allows the SPs to establish a trust relationship with a foreign IdP based on using distributed trusted 3rd parties denoted as $TDRs$. The TDR is responsible for trusting the agents and sharing its data with a requester agent or another TDR node. However, the agents trust measurement is out of this research scope. The required identity verification and messages design are discussed in detail. The details of the implementation of the $GIdM$ will be presented in the next chapter.

Chapter 5.

Global Identity Management Testing Using a Simulation Environment

As discussed in Section 4.1, the $GIdM$ is composed of an actor node that uses a $GARI$ to request a service or data from an SP, $IdPs$ that provide the actors identities, an SP that uses the identity establishment method, and the TDR nodes to manage the establishment of a dynamic trust relationship between the independent SP and $IdPs$. Moreover, it requires building three additional extension headers for IPv6 to represent the $GIdV$ protocol messages. Due to the difficulty of implementing them in a real environment, a discreet event simulation environment has been chosen instead. In computer networks, the discreet event simulations are widely used by the research community to implement and test their works at all of the computer network layers. This is because of two reasons, first, the simulation model is very well fitted to system consecrations; second, the ease of implementing the discreet events simulation [129]. Hence, the discreet event simulation is suitable for implementing the $GIdM$ in a simple and flexible way to achieve the overall $GIdM$ testing. NS3 is a widely used network simulator in the research community that already has several classes needed to test the proposed solution. Thus, it been chosen to develop the new IPv6 extension headers and build the $GIdM$ testing environment. Appendix F gives a brief explanation of the NS3 simulation.

This chapter gives a detailed explanation of the configuration of $GIdM$ in the NS3 simulation environment. Secondly, the possible testing scenarios of $GIdM$ system is presented. Finally, the validation of testing environment implementation in NS3 is presented.

5.1. $GARI$ Composer Implementation and Verification

The preliminary step to test the $GIdM$ is composing the global actor relationship identifier $GARI$. The $GARI$ composing algorithm that is discussed in Section 4.2.3 will be implemented using the C++ programming language. The actor's information will be gathered from the domains $IdPs$ tables ($ActorInfo$ and $Relations$), which are implemented using the SQLite3 database engine. To compose a $GARI$, the composer subsystem requires the number of actors that will be represented in the $GARI$ and their record number (ID) in the $ActorInfo$ table. These data will be used as input for the algorithm to generate the $GARI$ that will be used by the actor node at the time of requesting a service or data from the SP node. It is worth to note that a hash function for the actor' relationship's nonce will not be applied because privacy is out of this research objective.

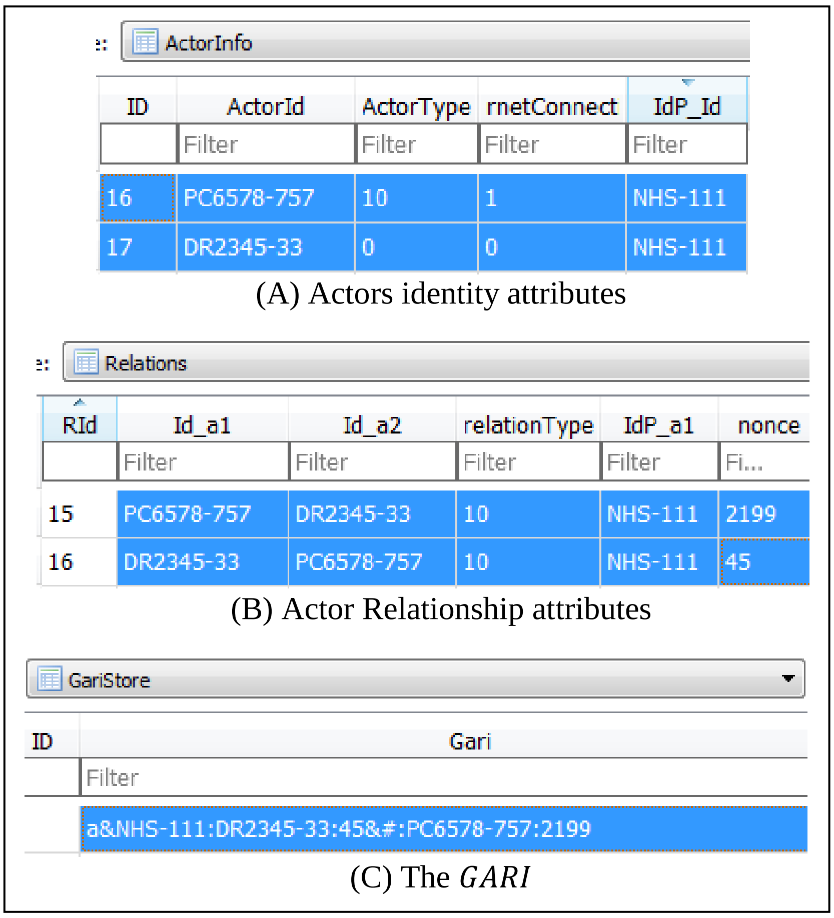
To ensure that the $GARI$ composing algorithm process is implemented correctly, the following verification approach will be used. Firstly, there will be a manual manipulation to compose the $GARI$ identifier using the actor relationship and their identity data. Secondly, the $GARI$ identifier will be composed by the implemented algorithm. Finally, they will be mapped to validate the results. This approach will be followed to validate the representation of different situations by composing two $GARI$ for two cases. The first case is used to validate composing a $GARI$ to represent two actors, an EA and a co, in a semi-permanent relationship where the communication object is of an active type of the Internet connectivity. However, the second case is used to compose a $GARI$ to represent three actors, an EA and two cos, in two relationships. The first co is of a passive type of the Internet connectivity, which required a second co of an active type to access the Internet. Moreover, the first relationship is of a permanent type, while the second is of an open-access. A transitive relationship between the EA and the second co exists as well.

Table 5-1: The $GARI$ Composing for Case 1

Parameter	ARS					**Identification($a1$)**						**Identification($a2$)**				
Sub parameter	$T(a1)$	$IC(a2)$	TR	R	**&**	ID_{IdP}	**:**	ID_1	**:**	$N^1_{AR1,2}$	**&**	ID_{IdP2}	**:**	ID_2	**:**	$N^2_{AR2,1}$
Representation	00	1	0	10		NHS-111		DR2345-33		45		#		PC6578-757		2199
$GARI$	a&NHS-111:DR2345-33:45&#:PC6578-757:2199															

Case 1:

A doctor in a hospital has permission to use a group of computer machines. It is clear that the relationship is of semi-permanent type, i.e. $R = 10$. The effective actor in this relationship is the doctor of person type, which is represented by "00", and his/her Internet connectivity is of passive type $IC(a1) = 0$. On the other hand, the machine is of active Internet connectivity type; thus, there is no transitive relationship. Table 5-1 illustrates the actors and relationship attributes with the manually composed $GARI$. The same identifier is resulted from applying the composing algorithm as can be seen from Figure 5-1. This indicates the correctness of $GARI$ composer implementation.



ID	ActorId	ActorType	rnetConnect	IdP_Id
	Filter	Filter	Filter	Filter
16	PC6578-757	10	1	NHS-111
17	DR2345-33	0	0	NHS-111

(A) Actors identity attributes

RId	Id_a1	Id_a2	relationType	IdP_a1	nonce
	Filter	Filter	Filter	Filter	Fi...
15	PC6578-757	DR2345-33	10	NHS-111	2199
16	DR2345-33	PC6578-757	10	NHS-111	45

(B) Actor Relationship attributes

ID	Gari
	Filter
	a&NHS-111:DR2345-33:45&#:PC6578-757:2199

(C) The $GARI$

Figure 5-1: The $GARI$ Generated by the Composing Algorithm for Case1

Case 2:

A patient has a portable heart-monitoring sensor attached to his/her body that periodically sends data about the patient's health status to his/her consultant. The sensor uses Bluetooth technology to communicate with the patient's smartphone, which in turn sends data through the Internet.

In such scenarios the number of actors is three that are participating in two relationships. Let us assume that the first relationship is of permanent type and the second relationship is of a free relationship. An additional relationship is presented here which is the transitive relationship between the patient and the smartphone. Composing the $GARI$ will be done in two stages. The first stage identifies the relationship between the health sensor and the smartphone as a gateway for the sensor's data as depicted in Table 5-2.

Table 5-2: The $GARI$ Composing for Case 2 (stage1)

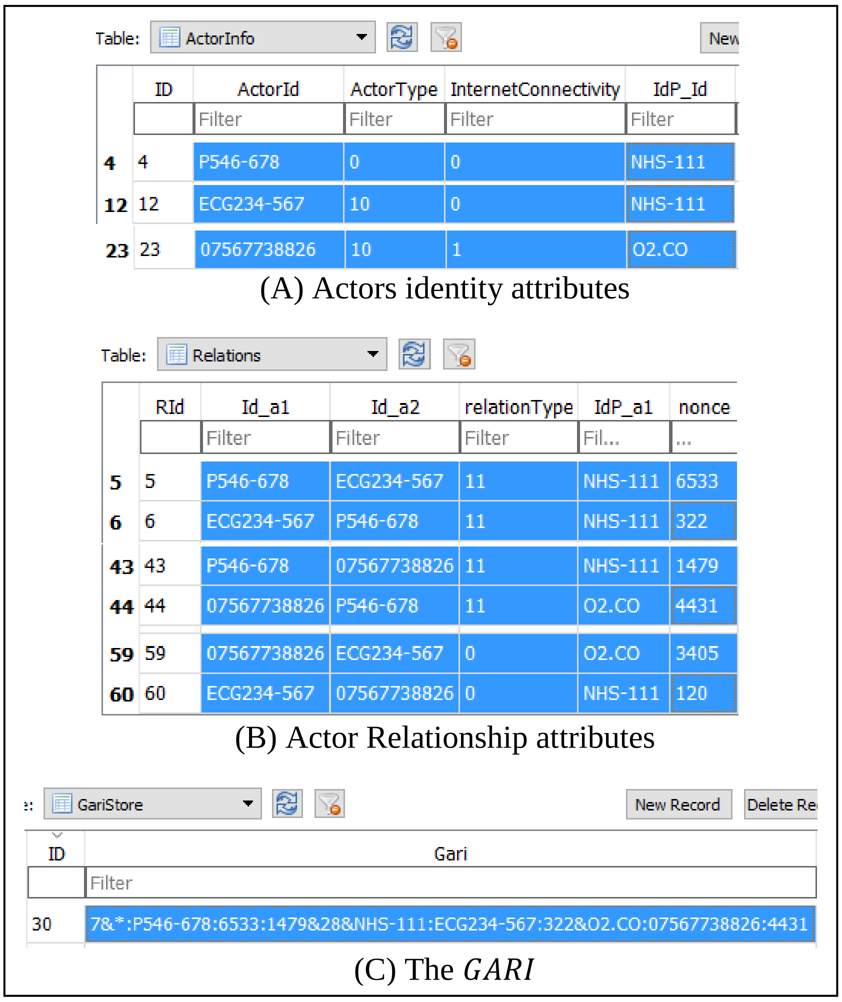
Parameter	ARS				&	**Identification($a2$)**					&	**Identification(a3)**				
Sub parameter	$T(a2)$	$IC(a3)$	TR	R	&	ID_{IdP}		ID_2		$N^2_{AR2,1}$	&	ID_{IdP3}		ID_3		$N^3_{AR3,1}$
Representation	10	1	0	00		NHS-111	:	ECG234-567	:	322		O2.CO	:	073647899931	:	4431
$tmpGARI$	28&NHS-111:ECG234-567:322&O2.CO:07567738826:4431															

Table 5-3: The $GARI$ Composing for Case 2 (stage 2)

Parameter	ARS				&	**Identification($A1$)**					&	**Identification($A2$)**		
Sub parameter	$T(a1)$	$IC(a2)$	TR	R	&	ID_{IdP}		ID_1		$N^1_{AR1,2}:N^1_{AR1,3}$	&	ID_{IdP2}		ID_2
Representation	00	0	1	11		NHS-111	:	P546-678	:	1479		NHS-111	:	$tmpGARI$
$GARI$	7&*:P546-678:6533:1479&28&NHS-111:ECG234-567:322&O2.CO:07567738826:4431													

The second stage is identifying the relationship between the patient and the sensor. The identifier generated from the first stage will be used in place of the secondary actor identifier by the second stage of $GARI$ composing. The transitive relationship will be represented in this ARS. Moreover, the IdP for the effective actor will be replaced by " $*$ " as it is already defined in $tmpGARI$. The final $GARI$ depicts in Table 5-3. The composed $GARI$ is the same as compared as the one generated by the implemented algorithm as depicted by Figure 5-2.

Based on these two cases, it could be concluded that the implemented $GARI$ composing algorithm is correct to compose the identifier following the above explained steps. The generated $GARI$ is unique when compared with the manually manipulated ones.



	ID	ActorId	ActorType	InternetConnectivity	IdP_Id
4	4	P546-678	0	0	NHS-111
12	12	ECG234-567	10	0	NHS-111
23	23	07567738826	10	1	O2.CO

(A) Actors identity attributes

	RId	Id_a1	Id_a2	relationType	IdP_a1	nonce
5	5	P546-678	ECG234-567	11	NHS-111	6533
6	6	ECG234-567	P546-678	11	NHS-111	322
43	43	P546-678	07567738826	11	NHS-111	1479
44	44	07567738826	P546-678	11	O2.CO	4431
59	59	07567738826	ECG234-567	0	O2.CO	3405
60	60	ECG234-567	07567738826	0	NHS-111	120

(B) Actor Relationship attributes

ID	Gari
30	7&*:P546-678:6533:1479&28&NHS-111:ECG234-567:322&O2.CO:07567738826:4431

(C) The $GARI$

Figure 5-2: The $GARI$ Generated by the Composing Algorithm for Caes2

5.2. Configuring the Test Environment

In order to test the ability of the proposed $GIdM$ to establish the effective actor identity in the IoT environment, the general use-case of actor interaction, discussed in Section 3.1 above, will be used as a base to build the testing environment as in the following subsections.

5.2.1. Implementing the Global Identity Verification Protocol

As discussed above, the IPv6 protocol is used as a base to implement the $GIdV$ protocol by designing three new IPv6 extension headers called $(GARI\ Ext.Hdr.)$, $(VDR\ Ext.Hdr.)$, and $(IV\ Ext.Hdr.)$ respectively. The interface card port numbers that are specified to send/receive them are proposed to be 5000, 6000, and 5500 respectively.

The new IPv6 extensions have been defined as subclasses in the core classes of NS3. Thus, they could be called by any node in NS3 to send and receive data using the IPv6 protocol. The following approach will be followed to define them in a set of classes that can be found in the folder “src/internet/model” of the NS3 source code.

1. Add the new IPv6 extension headers code to “ipv6-header.h” class.
2. Define the IPv6 packet functions to write/read data to/from the new IPv6 extension headers as subclasses in the ipv6-extension-header{.h, .cc} classes. These functions are set/get next header, set/get the header length, print, get the serialize size, serialize, and desterilize.
3. Implement the new IPv6 extension as subclasses in the classes ipv6-extension{.h, .cc}. This class is used by the nodes to process the IPv6 extensions. It is used by the “Ipv6L3Protocol::Receive” function to read the extensions.
4. Add the IPv6 extensions to the ipv6ExtensionDemux in file “ipv6-l3-protocol.cc” that is used to implement the IPv6 layer.
5. Rebuild the NS3 programme to check the compatibility of the newly defined extension with the other NS3 classes.

5.2.2. Network Configuration

To simulate the IoT environment to test the proposed solution ($GARI$ and $GIdM$), the general actor’s interaction uses-case, discussed in Section 3.1, will be used as a basis to build the testing environment. The proposed testing environment is composed of five $IdMs$ domains named (O2.CO, CARINSUR.CO, NHS-111, UOMAN.AC.UK, GAMING.CO) that are linked by three routers. Each domain consists of SP, IdP, and $Actor\ node$s. However, for the purpose of testing the suggested model, it is proposed to use a SP and five $IdPs$ and an $Actor\ node$ as a testing network topology. The proposed testing network topology is illustrated in Figure 5-3.

Each domain IdP is represented by a node with a database engine using the SQLite3. Two of them, e.g. IdP_0 and IdP_1, are participating in a CoT with a service provider, which is the SP node. In addition, two trusted Domains registry TDR nodes are linked with these routers to help in establishing a trust relationship between the unrelated SP and IdP. An Actor node is attached to one of the routers, which is responsible for sending the $GARI$ as the requester identifier to the SP node. All nodes have the IPv6 address. The Routing Information Protocol ($RIPng$) has been activated for the Internet traffic simulation. $RIPng$ starts to build the routing tables at each router at the fourth second after the simulation is started. It performs a periodic check for nodes reachability every 30 seconds according to the NS3 official website [130]. This will help to monitor the changes that might happen in the network nodes status. It allows nodes to send packets between each other using open shortest path first algorithm. The simulation environment parameters are illustrated in Table 5-4. Moreover, three new extension headers have been added to the IPv6 to implement the proposed $GIdV$ protocol. Finally, the identity establishment system installed on the SP node to manage the communication over these protocols and to make the final identity establishment decision. The following sections describe the system implementation briefly. Figure 5-4 illustrates main entities methods in the $GIdM$.

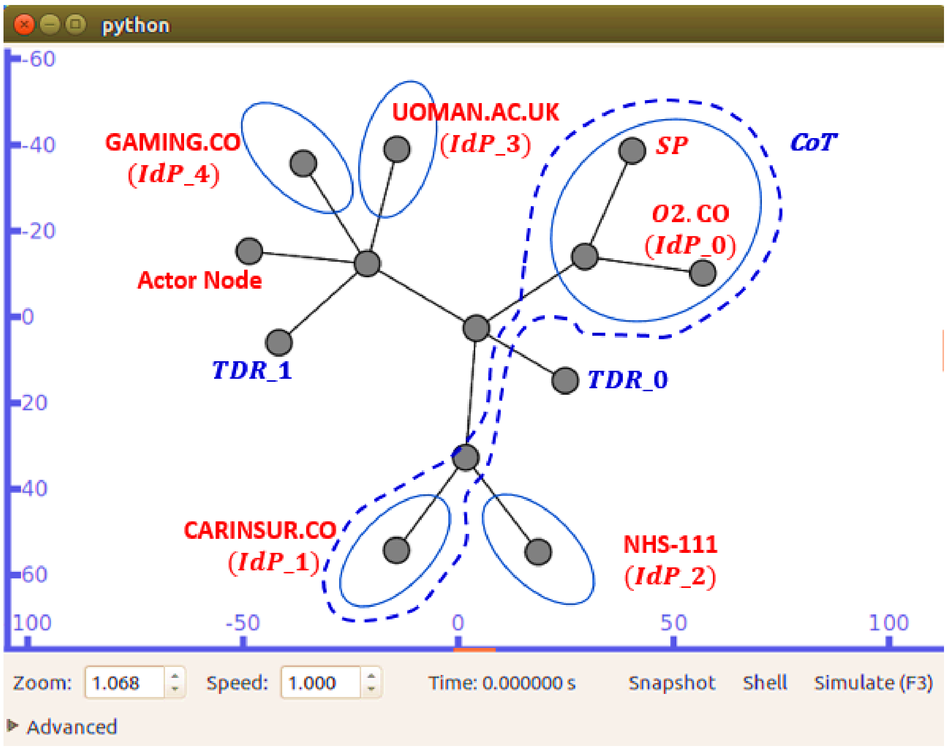


Figure 5-3: The Network Topology

Table 5-4: Simulation Parameters

No.	Parameter	Value
1	No. of IdP node	5
2	No. of SP node	1
3	No. of actor node	1
4	No. of CoT	1
5	No. of TDR node	2
6	Channel	$P2P$
7	Channel attribute: Delay	2 ms
8	Channel attribute: Mtu	1500
9	Channel attribute: DataRate	5000000
10	Routing algorithm	$RIPng$
11	$GIdM$ starting time	Sec 5

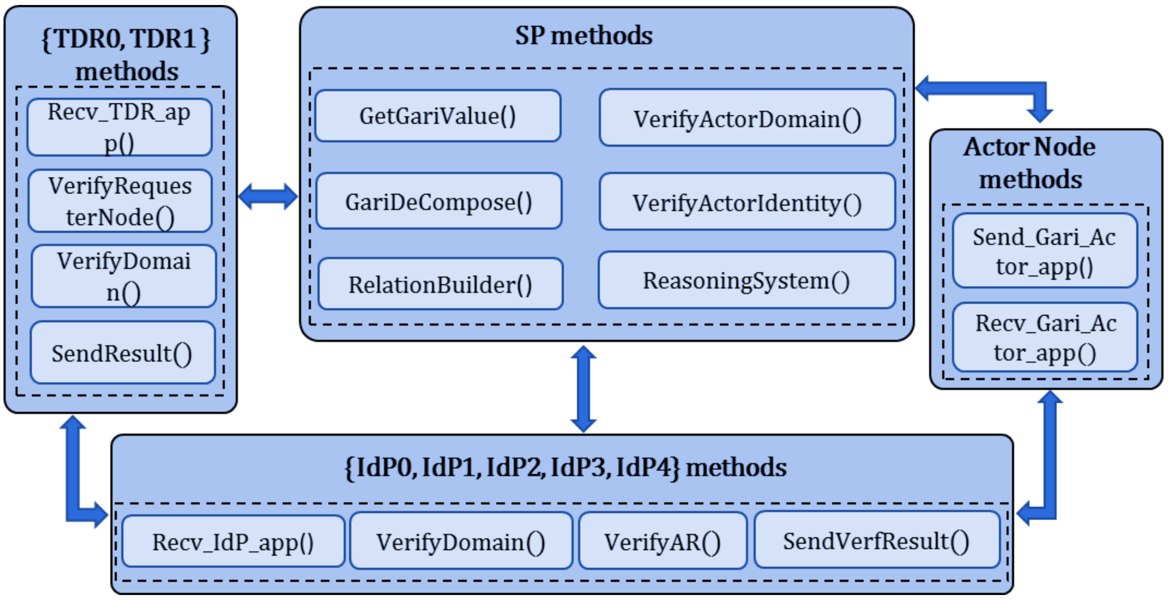
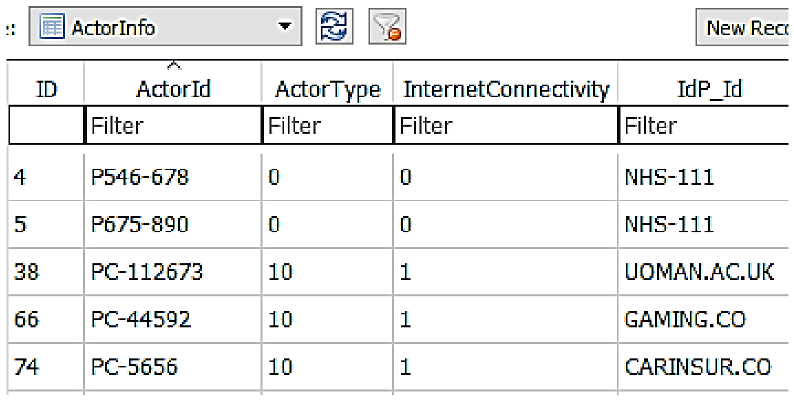
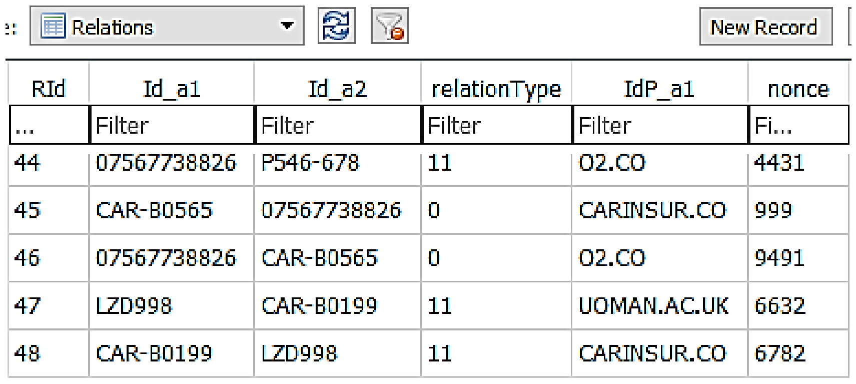


Figure 5-4: The Entities in the $GIdM$ with their Main Methods



ID	ActorId	ActorType	InternetConnectivity	IdP_Id
	Filter	Filter	Filter	Filter
4	P546-678	0	0	NHS-111
5	P675-890	0	0	NHS-111
38	PC-112673	10	1	UOMAN.AC.UK
66	PC-44592	10	1	GAMING.CO
74	PC-5656	10	1	CARINSUR.CO

Figure 5-5: Sample of $ActorInfo$ Table



RId	Id_a1	Id_a2	relationType	IdP_a1	nonce
...	Filter	Filter	Filter	Filter	Fi...
44	07567738826	P546-678	11	O2.CO	4431
45	CAR-B0565	07567738826	0	CARINSUR.CO	999
46	07567738826	CAR-B0565	0	O2.CO	9491
47	LZD998	CAR-B0199	11	UOMAN.AC.UK	6632
48	CAR-B0199	LZD998	11	CARINSUR.CO	6782

Figure 5-6: Sample of $Relations$ Table

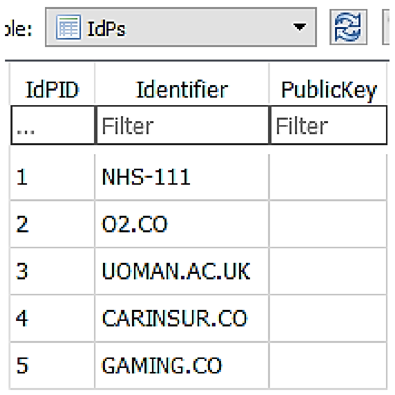
5.2.3. Identity Provider Nodes Configuration

The network topology contains five $IdPs$ that are represented by NS3 nodes, each of which has a SQLight3 engine. The IdP node has two main roles. Firstly, to interoperate with the $GARI$

composer to supply the actors' identity attributes and the relationship attributes. Secondly, to respond to the SPs identity verification requests as described in Section 4.3. The SQLight3 engine is compatible with NS3 and have to be installed prior to the configuration process. For this research purpose, two tables have been created with the SQLight3 called $ActorInfo$ and $Relations$ based on the DB design in Section 4.2.1. the $ActorInfo$ table stores the actors identification data. The IdP identifier is added as a field in this table to refer to the IdP that manages the actor identifier as can be seen in Figure 5-5. The $Relations$ table hosts the actors relationship attributes. Similarly, the IdP identifier was added as an additional field to the table to represent the domain that manages the actor relationship as can be seen in Figure 5-6.

In order to verify an actor identity, every IdP node has an instantiation of four main methods as illustrated in Figure 5-4. These methods are:

- Recv_IdP_app() - which is used to handle the $IV\ Ext.Hdr.$ in a request packet at the IdP socket and to extract its data.
- VerifyDomain() - which is used to verify the existence of a trust relationship with the requester SP node. If it does not exist, a TDR will be contacted to verify the SP node by sending a $VDR\ Ext.Hdr.$; otherwise the next method will be invoked.
- Verify AR() - which has the responsibility of verifying the actor identity based on his/her/it extracted identity and actor relationship attributes.
- SendVerfResult() - this method will send a reply message to the requester SP node with the identity verification result using a $IV\ Ext.Hdr.$ as discussed earlier.



IdPID	Identifier	PublicKey
...	Filter	Filter
1	NHS-111	
2	O2.CO	
3	UOMAN.AC.UK	
4	CARINSUR.CO	
5	GAMING.CO	

Figure 5-7: Sample of TDR Table

5.2.4. Trusted Domains Registry Nodes Configuration

Similar to the IdP nodes, the TDR nodes has been implemented as nodes in NS3 with the SQLight3 engine. It represents a trusted 3rd party that is used by its entities to build a trust relationship with other entities without prior knowledge. Thus, its role is answering the domain verification requests that are received from trusted entities after checking its local list of trusted agents as discussed in Section 4.2.1. However, establishing the entities trust is out of the research objective. Figure 5-7 illustrates the trusted $IdPs$ table fields by the TDR nodes.

Every TDR node has an instantiation of four main methods as illustrated in Figure 5-4. These methods are:

- Recv_TDR_app() - which is used to handle the $VDR\ Ext.Hdr.$ in a request packet at the TDR socket and to extract its data.
- VerifyRequsterNode() - which is used to verify whether the requester node is trusted or not. If it is one of the registered nodes with the TDR, then the next method will be invoked; otherwise the request will be declined.
- VerifyDomain() - which is used to check whether the domain in the inquiry is trusted by the TDR or not. If not, it multicasts the query to its partners in *TDR layer* until the information is found or exceeds the hop's limit using the $VDR\ Ext.Hdr.$.
- SendVerfResult() - which is used to send a reply message to the requester node with the inquiry result using a $VDR\ Ext.Hdr.$

5.2.5. Service Provider Node Configuration

In $GIdM$, the node that is responsible of establishing the actor identity is the SP or the node that works as a gateway/broker for other tiny objects. In the proposed network topology this node is represented by the SP node, which performs all tasks explained in Section 4.3.4. The SP node has an instansiation of the following methods to perform its tasks.

- GetGariValue() - this method role is to handle the $GARI\ Ext.Hdr.$ of a request received at the SP socket and to extract the $GARI$ value.
- GariDeCompose() - this method is responsible for analysing the received $GARI$ into its composition data. It is the implementation of the flowchart in Figure 4-3.
- RelationBuilder() - this method uses the data resulted from the previous method to build the ART. Thus it is the implementation of the flowchart in Figure 4-4.
- VerifyActorDomain() - this method is responsible for verifying the $IdPs$ in the ART as discussed earlier in Section 4.2.4.2. If the IdP does not exist in the $L2TA(SP_v)$, a $VDR\ Ext.Hdr.$ will be firstly used to communicate with the TRD to build the trust relationship with the IdP; otherwise, the next method will be invoked.

- VerifyActorIdentity() - this method is responsible for verifying actor identity based on the actor identity attributes and the relationship attributes. The SP node uses an $IV\ Ext.Hdr.$ to send an identity verification request to the IdP.
- ReasoningSystem() - this method is responsible for making the final decision of establishing the requester identity based on the actor identity attributes and the relationship with the communication objects as explained in the flowchart in Figure 4-6. In addition it sends the identity establishment result to the actor node using a $GARI\ Ext.Hdr.$

5.2.6. Actor Node

In order to start the identity establishment processes, (4) seconds are required to build the global routing table using the $RIPng$ method. Thus, the actor node will send a packet loaded with a $GARI$ to represent actors in a relationship at the second (5) of starting the simulation. This node application has two main methods that are explained below.

- Send_Gari_Actor_app() - this method used to send the packet with the $GARI\ Ext.Hdr.$ to the SP node to trigger the identity establishment system.
- Recv_Gari_Actor_app() - this method will handle the received packet with the identity establishment result using the $GARI\ Ext.Hdr.$

5.3. Global Identity Management Testing Scenarios

In order to test the proposed $GIdM$ in the IoT, three general actors interaction scenarios could be used to represent the actors' interaction models as discussed previously in Section 3.1. The $GIdM$ behaviour in these scenarios will be represented in the following general Sections. Then, these actors interaction scenarios will be the base to design the $GIdM$ testing scenarios.

5.3.1. Domain Interaction Scenario 1: $SP, EA,$ and $Co(s)$ Interacting within a Single Domain/ CoT

This scenario represents the classical IdM model where the trust relationship is already established between the SP, and the $IdP(S)$ to manage the actor identity. Therefore, the SP could interconnect with the $IdP(s)$ directly without the need to communicate with the TDR node. However, in it the actor relationship attribute in addition to actor attributes have been used to establish the EA identity. The sequence diagram in Figure 5-8 shows the $GIdM$ behavior to establish the EA identity based on its relationship with a communication device and the identities are managed by two different $IdP(s)$. The same behavior will be seen if the $GARI$ represents a multiple actor relationship.

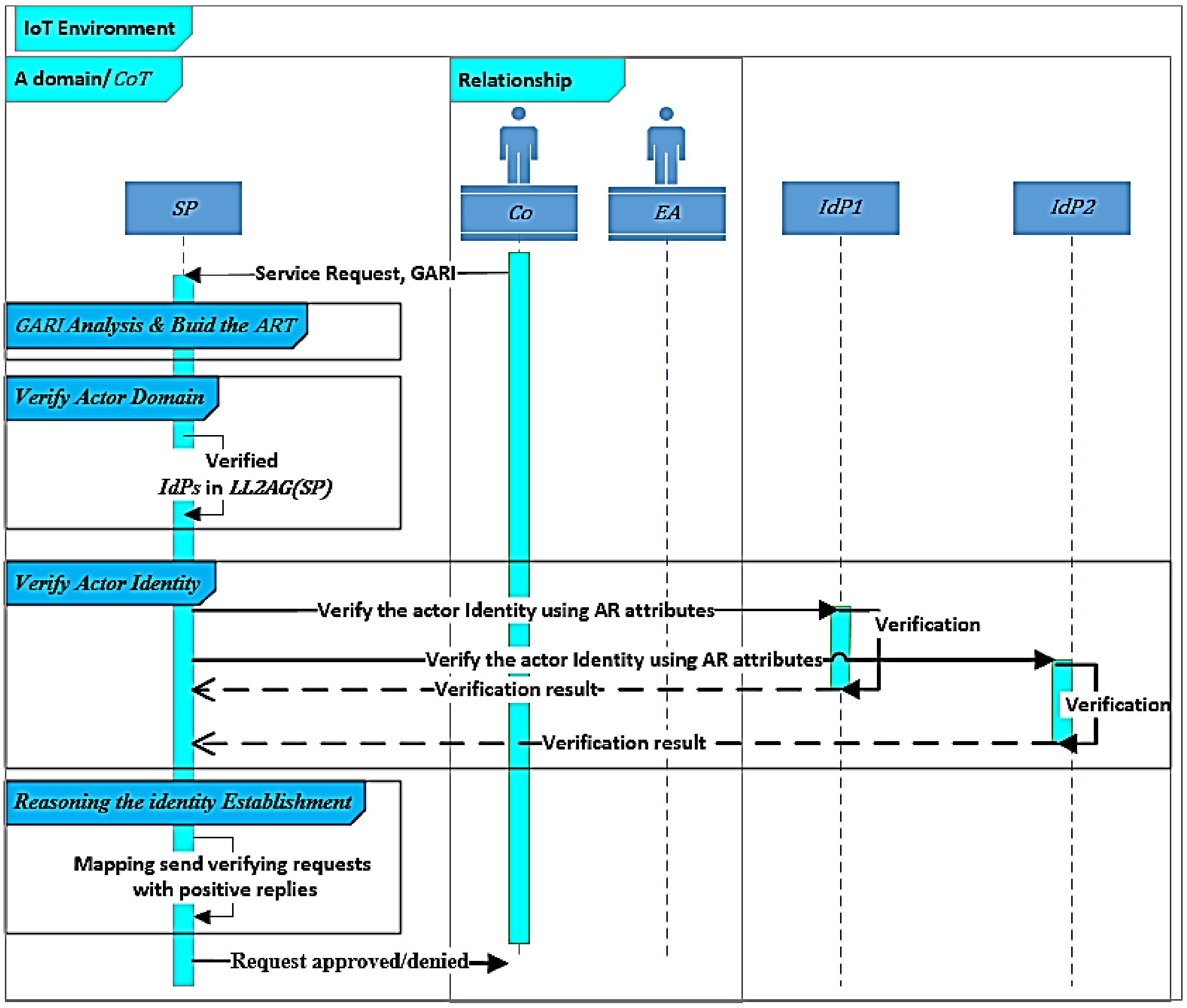


Figure 5-8: The $GIdM$ Behaviour in Testing Scenario 1

5.3.2. Domain Interaction Scenario 2: SP, EA, and $Co(s)$ Interacting within Mixed Domain(s)/$CoT(s)$

In such scenarios, the SP has been trusted by some of the actors' home $IdP(s)$, while the other IdP are unknown. Thus, it relies on a trusted TDR to build the trust relationship with foreign $IdP(s)$. The same approach is used by the foreign $IdP(s)$ to build the trust relationship with the unrelated SP because the trust relationship required is a bidirectional relation. Figure 5-9 illustrate the $GIdM$ in a scenario where the $GARI$ is composed of two actors, the identity of one of them is managed by a trusted IdP, while the other is not. Building a trust relationship between the visited SP and the foreign IdP is a bidirectional activity, thus each entity relies on a TDR to verify the other entity. The sequence diagram presents the basic behaviour of $GIdM$. It would be replicated to cope with scenarios of multiple actors' relationships being represented by $GARI$.

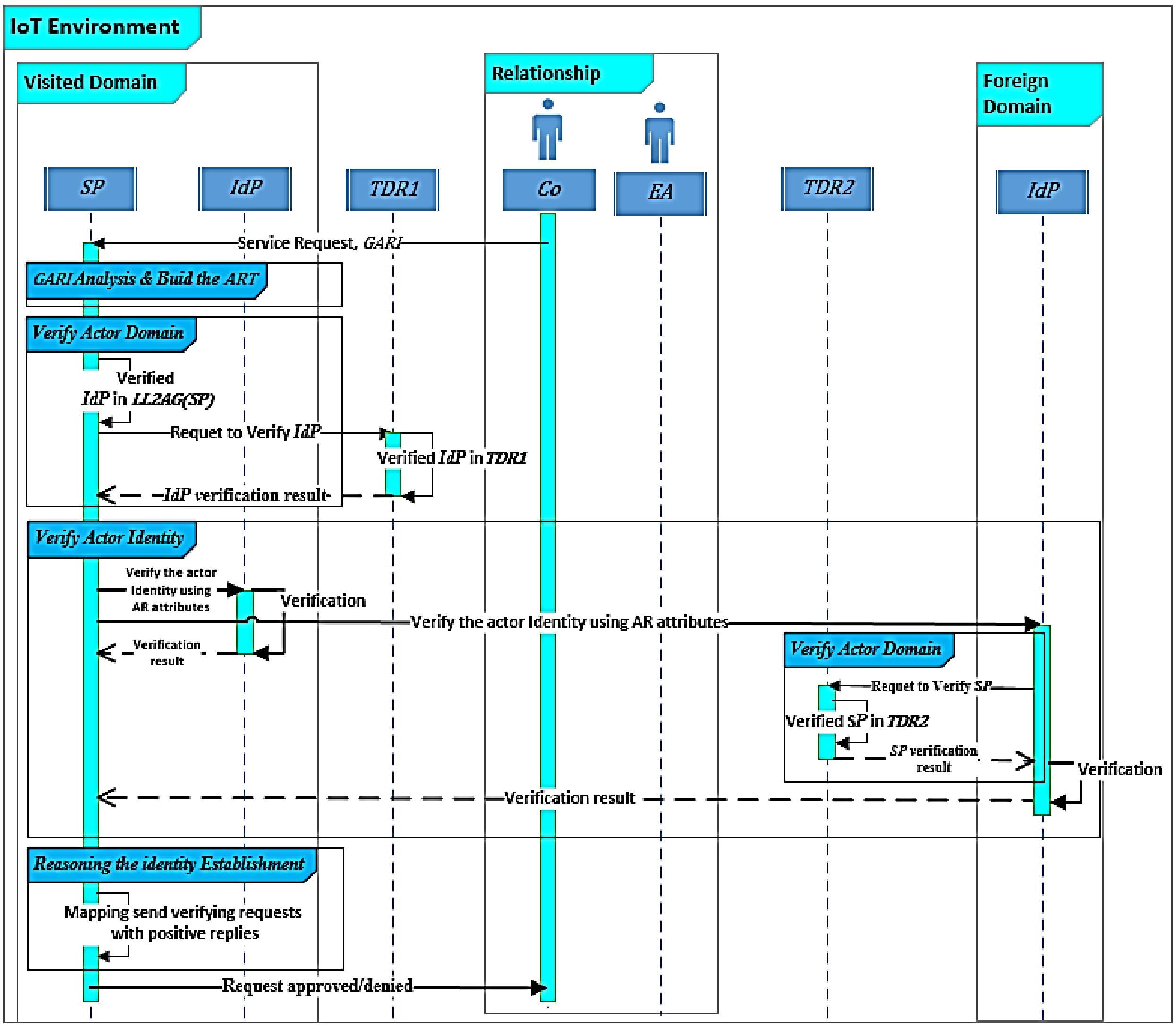


Figure 5-9: The $GIdM$ Behaviour in Testing Scenario 2

5.3.3. Domain Interaction Scenario 3: SP, EA, and $Co(s)$ Interacting Across Domain/ CoT(s)

The last type of the possible interacting scenarios is when the actor node uses a $GARI$ that composes of actors' identities that are managed by foreign $IdP(s)$ with respect to the SP. That means the bidirectional trusts relationships between the SP and the $IdP(s)$ have to be built first, and then, verify the actor identity based on those relationships. Figure 5-10 shows the scenario of actor node uses a $GARI$ to represent two actors in a relationship; where the actors' home $IdPs$ are differ, the SP and $IdPs$ are trusted by different $TDRs$, and there are no previous trust relationships established. Therefore, the SP node starts by verifying the foreign $IdPs$ through its trust $TDRr$. The next step is sending the IV requests to these trusted $IdPs$ that in turn request the SP verification from $TDRm$. Then, they perform the actor identity verification and send the result back to the SP. Finally, the SP doed the identity establishment process to approve the request or deny it. The same approach will be replicated to establish the actor identity in the case where the SP receives a $GARI$ representing more than two actors participating in relationships.

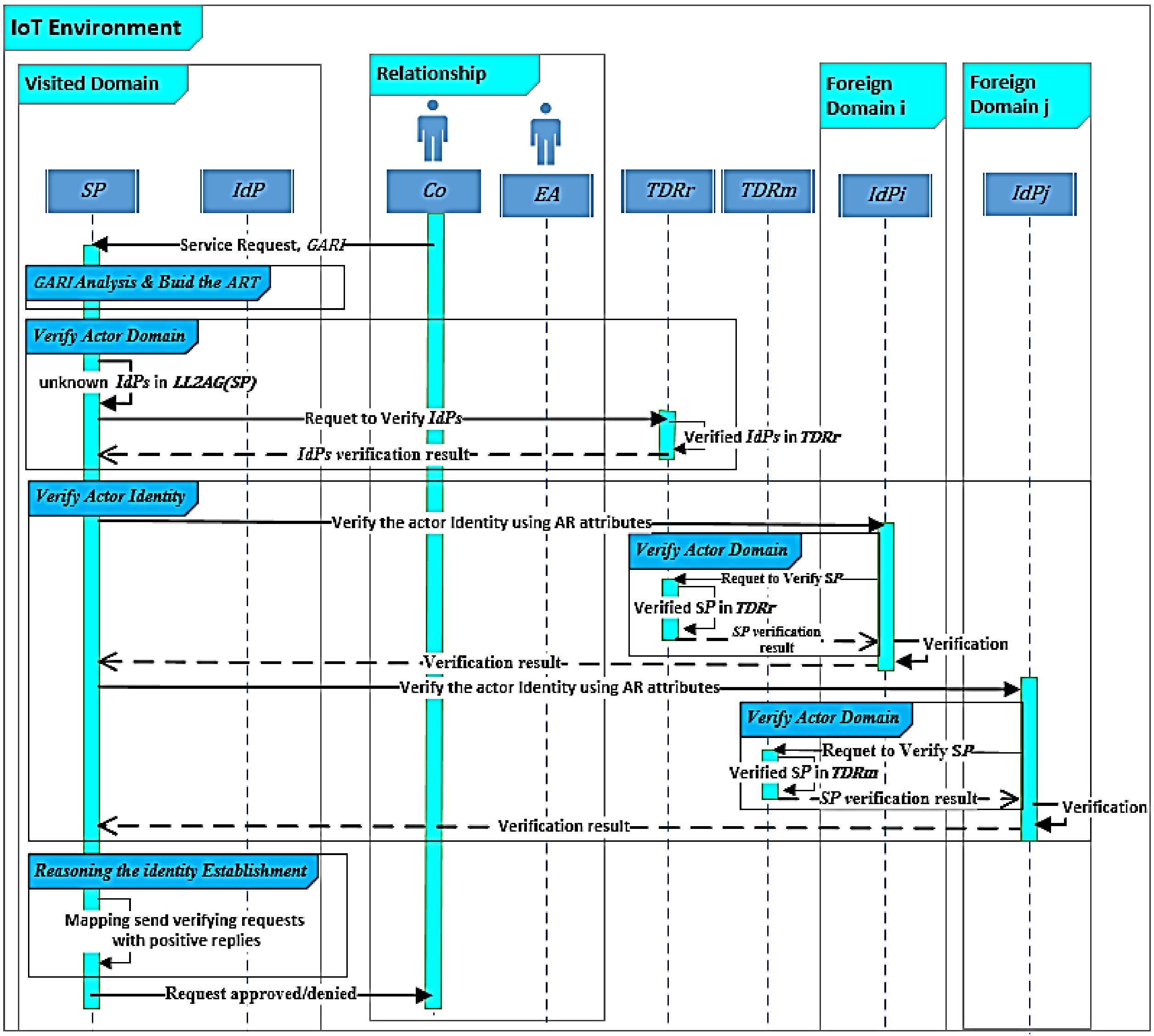


Figure 5-10: The $GIdM$ Behaviour in Testing Scenario 3

5.3.4. Testing Scenarios

This section discusses the possible scenarios to test the interoperability between the entities in $GIdM$ for the purpose of establishing the EA identity behind the communicated device(s) by any SP in the IoT environment. The proposed network topology is composed of five identifier domains with a CoT activated between two of them. Two or three actors participating in a single or double actors' relationship(s), respectively, are enough to show the basis of actors' interaction testing scenarios because using more actors and relationship is only a replication of the same process. These domains $IdPs$ supply their clients with the identifiers that will be used to form the $GARI$ as explained in Section 4.2.3.

The actor interactions criteria presented in Figure 3-1 shows three main interactions criteria that are domain interaction, actor interaction, and the mode of interaction. These criteria are the bases used by entities in $GIdM$ to interact with each other. The type of domains interaction specify three types of relationship between the SP and the home $IdPs$ that manage the actors' identities. Three types of the actors' relationship specify how they interact with each other. In terms of the number of actors' relationships required to allow the EA to access the Internet to request a service or share data, it is either a single or multiple relationship. All these criteria have to be tested to assure that the $GIdM$ has successfully established the identity under different circumstances.

To test the interoperability under these criteria, all the possible interaction scenarios have to be tested to simulate communication in the IoT environment. Therefore, the statistical combination approach has been chosen to get the probable testing scenarios of the following sample space variables: $Actors = \{2,3\}$; $IdPs = 5$ (2 in a $CoT,$ 3 independent $IdPs$); $relationships\ number = \{1,2\}$; $relationship\ types = 3$; $domain\ interaction\ types = 3$. The number of possible testing scenarios can counts using the combination (C) formula with replacement [120], where the selected $IdPs$ are not fixed, as follows:

$$nC_r = \binom{n+r-1}{r} = \frac{(r+n-1)!}{r!\,.(n-1)!} \quad \text{......................................} \quad (6.1)$$

Where,

$n = 5$; The total number of $IdPs$

$r = \{2|3\}$; The number of selected $IdPs$

The number of combinations testing scenarios with the 3 types of relationships are compute as follows:

- The number of testing scenarios combinations, where $(r = 2)$, 1 relationship, and 3 combinations of relationship types, is $(5C_2 * 3 = 15 * 3 = 45)$.
- The number of testing scenarios combinations, where $(r = 3)$, 2 relationship, and 9 combinations of relationship types, is $(5C_3 * 9 = 35 * 9 = 315)$.

The total possible combination of such cases is (360). However, most of them are a replication of scenarios that are not needed to be tested again. Thus, the final number of designed testing scenarios after remove the replicated scenarios are (24), that are depicted in Table 5-5.

Table 5-5: Testing Scenarios of Actor's Interaction

No.	No. of Actors	No. of relationship	Interaction		$IdP0$	$IdP1$	$IdP2$	$IdP3$	$IdP4$
			Actor	Domain					
1	2	1	S.Perm.	Single D./CoT	✓				
2	2	1	Open	Single D./CoT	✓				
3	2	1	Perm	Single D./CoT		✓			
4	2	1	Open	Single D./CoT		✓			
5	2	1	Perm	Single D./CoT	✓	✓			
6	2	1	Open	Single D./CoT	✓	✓			
7	2	1	S.Perm	Hybrid	✓		✓		
8	2	1	Open	Hybrid	✓		✓		
9	2	1	S.Perm	Hybrid		✓			✓
10	2	1	Open	Hybrid		✓			✓
11	2	1	Perm	Across-D.			✓	✓	
12	2	1	Open	Across-D.			✓	✓	
13	3	2	Open-Open: Open	Hybrid	✓	✓	✓		
14	3	2	Perm-Open: Open	Hybrid	✓	✓	✓		
15	3	2	Open-Open: Perm	Hybrid	✓	✓	✓		
16	3	2	Perm-Open: Perm	Hybrid	✓	✓	✓		
17	3	2	Open-Open: Open	Hybrid		✓		✓	✓
18	3	2	Perm-Open: Open	Hybrid		✓		✓	✓
19	3	2	Open-Open: Perm	Hybrid		✓		✓	✓
20	3	2	Perm-Open: Perm	Hybrid		✓		✓	✓
21	3	2	Open-Open: Open	Across-D.			✓	✓	✓
22	3	2	Perm-Open: Open	Across-D.			✓	✓	✓
23	3	2	Open-Open: Perm	Across-D.			✓	✓	✓
24	3	2	Perm-Open: Perm	Across-D.			✓	✓	✓

The above table represents an abstraction of the interaction criteria in the IoT in terms of actor interaction, domain interaction, and the mode of interaction that was represented in Figure 3-1. The typical IoT scenarios in Appendix D are examples of this abstraction. The actor interaction types are permanent, semi-permanent, and open-access that are denoted as Perm, S.Perm, and Open respectively. However, the domain interaction types are within a single domain/CoT, across-domain, or hybrid that are denoted as Single D./CoT, Across-D., and Hybrid respectively. Half of the table scenarios are testing the interaction of two actors in a relationship. Moreover, all of which are of a single mode of interaction where the combination of actor and domain interactions are defined. It worth to note that several $IdPs$ with different domain interaction types will be tested to represent the IoT cases.

The rest of the scenarios are designed to test three actors participating in two relationships. In these cases, the transitive relationship will be added as a third actor interaction to be tested.

For instance, in scenario number (20), the actor interaction type is (Perm-Open: Perm), which means that the interaction of the first and second actors is of a permanent type, the second and third actor is of an open-access type, and the last one shows that there is a transitive relationship between the first actor and the third. The domain interaction types represent the possible collaboration of several $IdPs$ in the IoT environment.

These scenarios represent the base combinations of actors and domains interaction in the IoT. In the case where more actors interact, it is only a combination of them.

5.4. Verifying the Simulated Environment

In this section, the implementation of the $GIdM$ model in NS3 will be verified to confirm it was correctly implemented and resulted the expected results. The model verification is essential to be assured of the reliability of the results, obtained from it. The same approach used in Section 5.1 to verify the $GARI$ composing implementation will be followed here. The $GARI$ from Table 5-3 has been chosen to verify the $GIdM$ model because it is already verified. It represents the participation of three actors from two domains in two relationships, and the existence of a transitive relationship between actors. Moreover, the mixed actors interaction scenario, explained in Section 5.3.2, will be used as a validation environment because it is able to summarise the $GIdM$ behaviours in all cases.

Figure 5-12 illustrates samples of the identity establishment messages that are transferred between the related nodes in NS3 simulation. As explained previously, the first step is sending a request message and presenting the $GARI$ to the SP, which could be seen in Figure 5-12 (a). The result of $GARI$ analysis is four relationships to verify the actor identity that represent two none-free relationships between the patient and two devices as explained by case 2 in Section 5.1. By default of the verification result values are set to “0” that appear in “Verified” field values at this stage, see Figure 5-11.

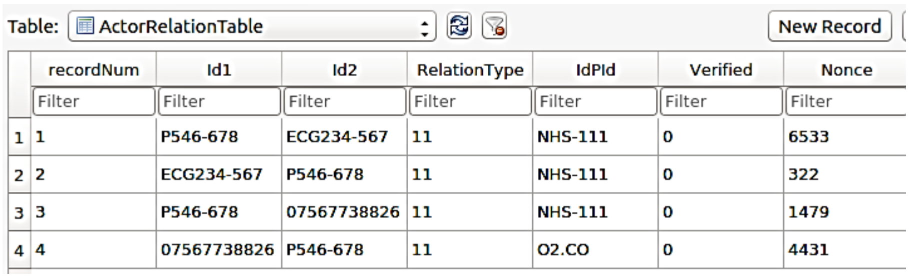


Table: ActorRelationTable

New Record

	recordNum	Id1	Id2	RelationType	IdPId	Verified	Nonce
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	P546-678	ECG234-567	11	NHS-111	0	6533
2	2	ECG234-567	P546-678	11	NHS-111	0	322
3	3	P546-678	07567738826	11	NHS-111	0	1479
4	4	07567738826	P546-678	11	O2.CO	0	4431

Figure 5-11: The ART at the SP node before the identity verification process

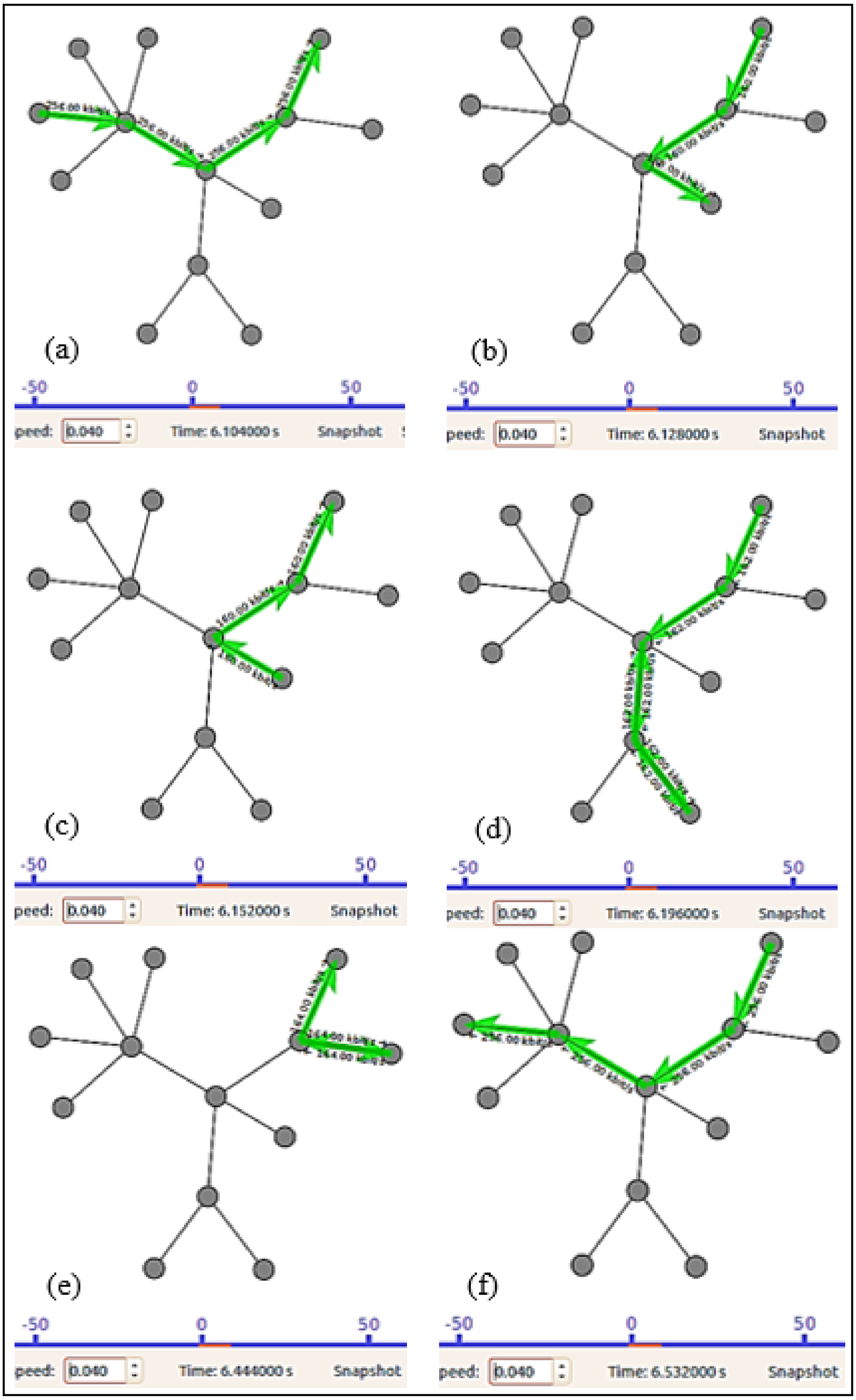


Figure 5-12: Snapshots of $GIdM$ Verification in NS3

From the table in Figure 5-11, the SP have to rely on two $IdPs$ to establish the actor identity, which are the "O2.CO" and "NHS-111". However, the trust relationship between the NHS-111 and the SP does not exist as it was assumed. Therefore, the SP will interoperate with the $TDR0$ to establish this missing trust relationship. Figure 5-12 (b) and (c) represent the domain verification stage request and answer messages respectively.

The third stage is verifying the actor identity by the trusted $IdPs$. Thus, SP interconnects with the $IdPs$, i.e. "NHS-111" and "O2.CO", to verify the actor identity based on the relationships' attributes as seen in Figure 5-12 (d) and (f) respectively. The answers for these actor verification requests will be used to update the "Verified" field in ART as seen in Figure 5-13.

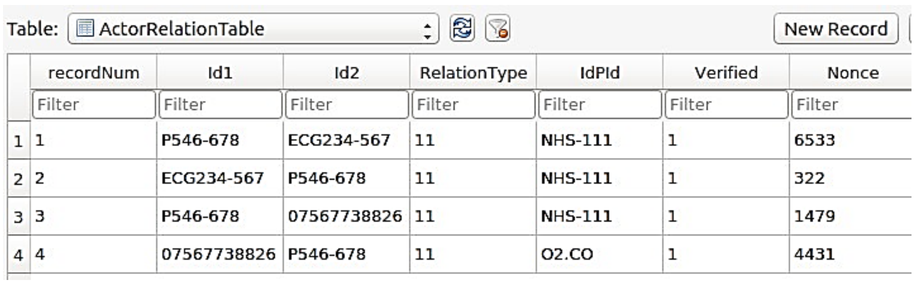


Table: ActorRelationTable

New Record

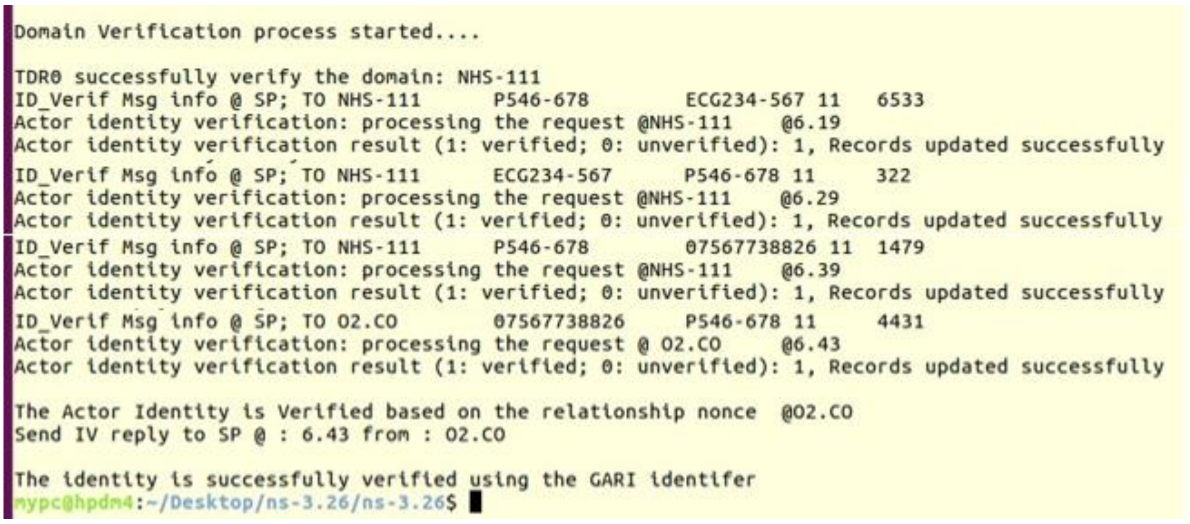
	recordNum	Id1	Id2	RelationType	IdPId	Verified	Nonce
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	P546-678	ECG234-567	11	NHS-111	1	6533
2	2	ECG234-567	P546-678	11	NHS-111	1	322
3	3	P546-678	07567738826	11	NHS-111	1	1479
4	4	07567738826	P546-678	11	O2.CO	1	4431

Figure 5-13: The ART at the SP after actor identity verification process

Finally, the reasoning stage of the identity establishment is mapping the number of requests messages with a positive answer to find whether the identity has been established or not. The ART shows that all actor identity verification request are positively replied when processed by the $IdPs$. Therefore, the actor identity is established in the simulation environment. Figure 5-12 (f) shows the service request result which is sent to the actor node by the SP to confirm the success of the identity establishment. A sample of NS3 output could be seen in Figure 5-14, which shows the above stages.

This scenario verifies of $GIdM$ implementation to establish the effective actor identity using the attributes of the actor's identity and the actor relationship. Two $IdPs$ are involved in the identity establishment process following the mix testing scenario. One of them is participating in a CoT with the SP, while the other is independent. The trust relationship with the foreign IdP is established across-domain relying on the TRD nodes. Therefore, it can be concluded that the implementation is correct and the testing environment is valid.

Figure 5-14: Sample of the NS3 Result


```
Domain Verification process started....

TDR0 successfully verify the domain: NHS-111
ID_Verif Msg info @ SP; TO NHS-111      P546-678        ECG234-567 11   6533
Actor identity verification: processing the request @NHS-111    @6.19
Actor identity verification result (1: verified; 0: unverified): 1, Records updated successfully
ID_Verif Msg info @ SP; TO NHS-111      ECG234-567      P546-678 11     322
Actor identity verification: processing the request @NHS-111    @6.29
Actor identity verification result (1: verified; 0: unverified): 1, Records updated successfully
ID_Verif Msg info @ SP; TO NHS-111      P546-678        07567738826 11  1479
Actor identity verification: processing the request @NHS-111    @6.39
Actor identity verification result (1: verified; 0: unverified): 1, Records updated successfully
ID_Verif Msg info @ SP; TO O2.CO        07567738826     P546-678 11     4431
Actor identity verification: processing the request @ O2.CO     @6.43
Actor identity verification result (1: verified; 0: unverified): 1, Records updated successfully

The Actor Identity is Verified based on the relationship nonce  @O2.CO
Send IV reply to SP @ : 6.43 from : O2.CO

The identity is successfully verified using the GARI identifer
mypc@hpdm4:~/Desktop/ns-3.26/ns-3.26$
```

5.5.Summary

The network simulator NS3 has been chosen to build the testing environment of the $GIdM$ because the proposed solution relies on developing new Ipv6 extension headers to implement the $GIdV$ protocol messages that are supported in NS3. The only thing needs to do in order to test the proposed solution is merge the newly implemented extension headers with the NS3 main classes. On the other hand, the general actor interaction use-case has been considered as the bases to build the test environment that includes five different IdP domains. Two of them are participating on a CoT, while the others are representing foreign domains.

This chapter has presented the $GIdM$ entities implementation using the NS3 as a testing environment. These entities are the $GARI$ composing algorithm, the $IdPs$ nodes, the TDR nodes, and the SP nodes. The identity establishment processes, by the SP, have four main stages that are: $GARI$ analysis and building the ART, verifying the domains, verifying the actors by these domains, and reasoning the identity establishment decision. The $GIdM$ testing scenarios are also explored. The configured simulation has been verified using a scenario in order to assure the reliability of the collected results. The $GIdM$ will be evaluated in the next chapter.

Chapter 6.

Evaluating the Global Identity Management

The evaluation process of the proposed $GIdM$ starts by formally validating the $GIdV$ protocol with a well-known BAN logic. The ProVerif tool performs a formal verification for the proposed protocol. The global actors' relationship identifier and the $GIdM$ are critically evaluated by comparing them with related work using the general requirements as evaluation criteria. Finally, the proposed system has been tested and the overheads are analysed in terms of computation and computation costs to establish the effective actor identity.

6.1. A Conceptual Validation with BAN Logic

A model validation is defined in [131] as *"Substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model"*. To assess the validity of the proposed system to identify the EA in the IoT environment, a mathematical tracing technique has been chosen as a *conceptual model validation*. Sargent in his highly cited paper [132], stated that *"The use of traces is the tracking of entities through each sub-model and the overall model to determine if the logic is correct and if the necessary accuracy is maintained"*. BAN logic is a "logic of authentication" developed by Burrows-Abadi-Needham to provide a formal analysis of the protocol [133]. It is widely used by researchers to validate the security protocols. It is used to prove the proposed system logically to: (1) dynamically establish the trust relationship based on distributed trusted 3rd parties and two-way authentication; (2) authenticate the EA identity based on its relationship(s) with IoT object(s) and a secure nonce.

BAN logic focuses on developing the principles belief and the freshness of the messages. It is composed of four main stages: messages idealisation, initial assumption declaration, goals declaration, and logical proof. Table 6-1, describes the standard BAN notations.

Table 6-1: The Standard BAN Notations

Notation	Description
$P \mid\equiv X$	P *believes* X: Principle P believes X. P acts as if X is true. P knows X is indeed true or the truth of X is justified by some evidence
$P \lhd X$	P sees X: The principal P receives a message containing X. Seeing is NOT believing. P does not necessarily believe X even if P sees X.
$P \mid\sim X$	At some point in the past, P is known to have sent a message including X.
$P \mid\Rightarrow X$	P controls X: P has jurisdiction over X, or P is trusted as an authority on X
$\#(X)$	Fresh(X): X is recent or has not been used before. It is a fresh value (nonce, timestamp).
$P \overset{K}{\leftrightarrow} Q$	P and Q may use the shared key K to communicate. K is only known to P and Q or a principal trusted by P and Q (such as an authentication server).
$\overset{K}{\rightarrow} P$	P has K as a public key. The private key (K^{-1}) is known only to P.
$\{X\}_K$	X is encrypted under the key K
$h(X)$	The formula X is hashed value
$P \overset{Z}{\Leftrightarrow} Q$	P believes that the secret Z is shared with Q.
$\langle X\rangle_Z$	The formula X is combined with the formula Z

The formal BAN postulates [134] that will be used in the proof have been described in the following rules:

Rule (1)**. Message Meaning rule:**

- *For public keys*: If *P* believes that *K* is *Q*'s public key, and *P* receives a message signed with *Q*'s secret key, then *P* believes that *Q*' once said *X:*

$$\frac{P \mid\equiv \overset{K}{\rightarrow} Q,\;\; P \lhd \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}$$

- *For shared secrets*: If *P* believes that secret *Z* is shared with *Q* and sees $\langle X\rangle_Z$, then *P* believes that *Q* once said *X*:

$$\frac{P \mid\equiv P \overset{Z}{\Leftrightarrow} Q,\;\; P \lhd \langle X\rangle_Z}{P \mid\equiv Q \mid\sim X}$$

Rule (2). **Nonce Verification rule:** If *P* believes that *X* is expressed recently (freshness) and *P* believes that *Q* once said *X*, then *P* believes that *Q* believes *X*:

$$\frac{P \mid\equiv \#(X),\ \ P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}.$$

Rule (3). **Jurisdiction rule:** If *P* believes that *Q* believes a message *X*, and, *P* believes that *Q* has jurisdiction/control over *X* then *P* trusts *Q* on the truth of *X:*

$$\frac{P \mid\equiv Q \mid\equiv X,\ \ P \mid\equiv Q \mid\Rightarrow X}{P \mid\equiv X}$$

Rule (4). **Freshness rule:** If one part is known to be fresh, the entire formula must be fresh:

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$$

Rule (5). **Belief rule**: If *P* believes *Q* believes the message set *(X, Y), P* also believes *Q* believes the message *X*:

$$\frac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv (X)}.$$

Rule (6). **Closure observations rule**: if a principle sees a formula then he sees its components, when he knows the necessary keys:

$$\frac{P \lhd \langle X \rangle_Z}{P \lhd X}\ ;\ \frac{P \mid\equiv \overset{K}{\rightarrow} Q, P \lhd \{X\}_{K^{-1}}}{P \lhd X};\ \ \frac{P \mid\equiv \overset{K}{\rightarrow} P,\ \ P \lhd \{X\}_K}{P \lhd X}$$

<u>1. Idealize *GIdV* protocol Messages with BAN Notation</u>

The first step is transforming the protocol messages between the participating principles to idealize form. At this phase all unencrypted messages will be dropped as well as all other data that will not contribute in developing the principles belief. The proposed protocol messages, as explained in Section 4.3, between the principles SP_v, TDR_{SP_v}, TDR_{IdP_x}, and IdP_x are idealized as follows:

M3: $TDR_{SP_v} \longrightarrow SP_v : \left\{ \xrightarrow{k_{IdP_x}} IdP_x\ , T_{r1} \right\}_{K^{-1}_{TDR_{SP_v}}}$

M4: $SP_v \longrightarrow IdP_x : \{N_{SP}, T_{mar1}\}_{k_{IdP_x}}$

M6: $TDR_{IdP_x} \longrightarrow IdP_x : \left\{ \xrightarrow{k_{SP_v}} SP_v, T_{r2} \right\}_{K^{-1}_{TDR_{IdP_x}}}$

M7: $IdP_x \longrightarrow\ SP_v : \left\{ \langle SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x, T_{mar2} \rangle_{N_{SP}} \right\}_{k_{SP_v}}$

M8: $SP_v \longrightarrow IdP_x : \left\{ \langle SP_v \overset{N_{SP_v}}{\Longleftrightarrow} IdP_x, IdP_x \mid\equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x \right), \langle AR_x \rangle_{h(N_x)} \rangle_{N_{IdP_x}}\ , T_{ivr} \right\}_{k_{IdP_x}}$

M9: $IdP_x \longrightarrow SP_v : \left\{ \langle AR_x, V \rangle_{N_{SP_v}}, T_{iva} \right\}_{k_{SP_v}}$

2. Goals

According to BAN analytical procedure, the following goals should be achieved from the proposed protocol.

Goal 1. $SP_v \mid \equiv \left(\xrightarrow{k_{IdP_x}} IdP_x \right)$

Goal 2. $IdP_x \mid \equiv \left(\xrightarrow{k_{SP_v}} SP_v \right)$

Goal 3. $SP_v \mid \equiv \ IdP_x \mid \equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x \right)$

Goal 4. $IdP_x \mid \equiv SP_v \mid \equiv \left(SP_v \overset{N_{SP_v}}{\Longleftrightarrow} IdP_x \right)$

Goal 5. $IdP_x \mid \equiv A_x \mid \equiv AR_x$

Goal 6. $SP_v \mid \equiv \ IdP_x \mid \equiv (AR_x, V)$

3. Initial Assumptions

For the proposed protocol, the initial assumptions are listed as follows:

A1: $SP_v \mid \equiv \left(\xrightarrow{k_{SP_v}} SP_v \right)$;

A2: $IdP_x \mid \equiv \left(\xrightarrow{k_{IdP_x}} IdP_x \right)$;

A3: $TDR_{SP_v} \mid \equiv \left(\xrightarrow{k_{TDR_{SP_v}}} TDR_{SP_v} \right)$;

A4: $TDR_{IdP_x} \mid \equiv \left(\xrightarrow{k_{TDR_{IdP_x}}} TDR_{IdP_x} \right)$;

A5: $SP_v \mid \equiv TDR_{SP_v} \mid \Rightarrow \left(\xrightarrow{k_{TDR_{IdP_x}}} TDR_{IdP_x} \right)$;

A6: $IdP_x \mid \equiv TDR_{IdP_x} \mid \Rightarrow \left(\xrightarrow{k_{TDR_{SP_v}}} TDR_{SP_v} \right)$;

A7: $SP_v \mid \equiv \left(\xrightarrow{k_{TDR_{SP_v}}} TDR_{SP_v} \right)$;

A8: $IdP_x \mid \equiv \left(\xrightarrow{k_{TDR_{IdP_x}}} TDR_{IdP_x} \right)$;

A9: $SP_v \mid \equiv \# (T_{r1})$;

A10: $TDR_{SP_v} \mid \equiv \# (T_{s1})$;

A11: $TDR_{IdP_x} \mid \equiv \left(\xrightarrow{k_{IdP_x}} IdP_x \right)$;

A12: $IdP_x \mid \equiv \# (T_{r2})$;

A13: $TDR_{IdP_x} \mid \equiv \# (T_{s1})$;

A14: $IdP_x \mid \equiv \# (T_{mar1})$;

A15: $SP_v \mid \equiv \# (T_{mar2})$;

A16: $SP_v \mid \equiv \left(SP_v \overset{N_{SP_v}}{\Longleftrightarrow} IdP_x \right)$

A17: $IdP_x \mid \equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x \right)$;

A18: $SP_v \mid \equiv \# (N_{SP_v})$;

A19: $IdP_x \mid \equiv \# (N_{IdP_x})$;

A20: $SP_v \mid \equiv IdP_x \mid \Rightarrow (AR_x)$;

A21: $IdP_x \mid \equiv \left(A_x \overset{h(N_x)}{\Longleftrightarrow} IdP_x \right)$;

A22: $IdP_x \mid \equiv \# (N_x)$.

4. Logical Proof with BAN

The last stage is to prove the protocol logically based on the above BAN's rules to achieve the goals.

According to M3, we could obtain:

$$SP_v \lhd \left\{\xrightarrow{k_{IdP_x}} IdP_x, T_{r1}\right\}_{K^{-1}_{TDR_{SP_v}}} \quad \text{(6.1)}$$

From assumption A7 and (6.1), we apply message-meaning rule (1) to get:

$$SP_v \mid\equiv TDR_{SP_v} \mid\sim \left(\xrightarrow{k_{IdP_x}} IdP_x, T_{r1}\right) \quad \text{(6.2)}$$

From A9 and (6.2), we apply the nonce verification rule 2 to get

$$SP_v \mid\equiv TDR_{SP_v} \mid\equiv \left(\xrightarrow{k_{IdP_x}} IdP_x, T_{r1}\right) \quad \text{(6.3)}$$

By breaking (6.3) up, we get

$$SP_v \mid\equiv TDR_{SP_v} \mid\equiv \left(\xrightarrow{k_{IdP_x}} IdP_x\right) \quad \text{(6.4)}$$

From assumption A5 and (6.4), we apply jurisdiction rule 3 to obtain:

$$SP_v \mid\equiv \left(\xrightarrow{k_{IdP_x}} IdP_x\right) \quad \textbf{(Goal 1)} \quad \text{(6.5)}$$

From M4 we could obtain

$$IdP_x \lhd \{N_{SP}, T_{mar1}\}_{k_{IdP_x}} \quad \text{(6.6)}$$

From (6.6) and using the closure observation rule we get

$$IdP_x \lhd (N_{SP}, T_{mar1}) \quad \text{(6.7)}$$

From A14 and (6.7), we apply the freshness rule and break up the result to get

$$IdP_x \mid\equiv \#(N_{SP}) \quad \text{(6.8)}$$

According to M6, we could obtain:

$$IdP_x \lhd \left\{\xrightarrow{k_{SP_v}} SP_v, T_{r2}\right\}_{K^{-1}_{TDR_{IdP_x}}} \quad \text{(6.9)}$$

From assumption A8 and (6.9), we apply the message-meaning rule to obtain:

$$IdP_x \mid\equiv TDR_{IdP_x} \mid \sim \left(\xrightarrow{k_{SP_v}} SP_v, T_{r2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.10)$$

From assumption (A12) and (6.10), we apply nonce verification rule to get:

$$IdP_x \mid\equiv TDR_{IdP_x} \mid \equiv \left(\xrightarrow{k_{SP_v}} SP_v, T_{r2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.11)$$

From assumption (6.14) and (6.11), we apply jurisdiction rule to obtain:

$$IdP_x \mid \equiv \left(\xrightarrow{k_{SP_v}} SP_v, T_{r2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.12)$$

By breaking (6.12) up we get

$$IdP_x \mid \equiv \left(\xrightarrow{k_{SP_v}} SP_v\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(Goal 2)} \ldots\ldots\ldots\ldots\ldots \quad (6.13)$$

According to M7, and by the closure observation we could obtain

$$SP_v \lhd \left(\langle SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x \rangle_{N_{SP}}, T_{mar2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.14)$$

From assumption A16 and (6.14), we apply meaning rule to obtain

$$SP_v \mid \equiv \; IdP_x \mid \sim \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x, T_{mar2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.15)$$

From A18, the freshness rule and (6.15), we apply the nonce verification rule and break the result to get:

$$SP_v \mid \equiv \; IdP_x \mid \equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x\right) \ldots\ldots\ldots\ldots\ldots\ldots \textbf{(Goal 3)} \ldots\ldots\ldots\ldots\ldots \quad (6.16)$$

According to M8, we apply the closure observation rule to get:

$$IdP_x \lhd \left(\langle SP_v \overset{N_{SP_v}}{\Longleftrightarrow} IdP_x, IdP_x \mid \equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x\right), \langle AR_x \rangle_{h(N_x)} \rangle_{N_{IdP_x}}, T_{ivr}\right) \ldots.. \quad (6.17)$$

From assumption A19 and (6.16), we apply the message meaning rule and nonce verification rule to obtain:

$$IdP_x \mid \equiv SP_v \mid \equiv \left(SP_v \overset{N_{SP_v}}{\Longleftrightarrow} IdP_x\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(Goal 4)} \ldots\ldots\ldots\ldots.. \quad (6.18)$$

$$IdP_x \mid \equiv SP_v \mid \equiv IdP_x \mid \equiv \left(SP_v \overset{N_{IdP_x}}{\Longleftrightarrow} IdP_x\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.19)$$

$IdP_x| \equiv SP_v \mid \equiv \langle AR_x\rangle_{h(N_x)}$ ………………………………………………….. (6.20)

By breaking (6.15) up we get

$IdP_x \lhd \langle AR_x\rangle_{h(N_x)}$ ………………………………………………………… (6.21)

From assumption A21, we apply the message-meaning rule to obtain:

$IdP_x| \equiv A_x \mid \sim AR_x$ ……………………………………………………… (6.22)

From assumption A22, the freshness rule and (6.22), we apply the nonce verification rule to obtain:

$IdP_x| \equiv A_x \mid \equiv AR_x$ ………………………………**(Goal 5)**………………. (6.23)

According to M9 and the closure observation rule, we obtain:

$SP_v \lhd (\langle AR_x, V\rangle_{N_{SP_v}}, T_{iva})$ ……………………………………………………. (6.24)

From A16, we apply the message-meaning rule to get:

$IdP_x| \equiv SP_v \mid \sim \left(\langle AR_x, V\rangle_{N_{SP_v}}, T_{iva}\right)$ …………………………………….. (6.25)

From (6.8), applying the freshness rule and break (6.25) up, we apply the nonce verification rule to obtain:

$SP_v \mid \equiv\ IdP_x \mid \equiv (AR_x, V)$ ………………………………**(Goal 6**)……….. (6.26)

5. Discussion

The $GIdM$ has been designed to achieve two main requirements. The first requirement is to build a trust relationship between SP_v and IdP_x based on distributed trusted 3rd parties and two-way authentication. The second requirement is to verify the user's identity behind the IoT's communicated object based on its relationship with the object and a secret within the IdP_x.

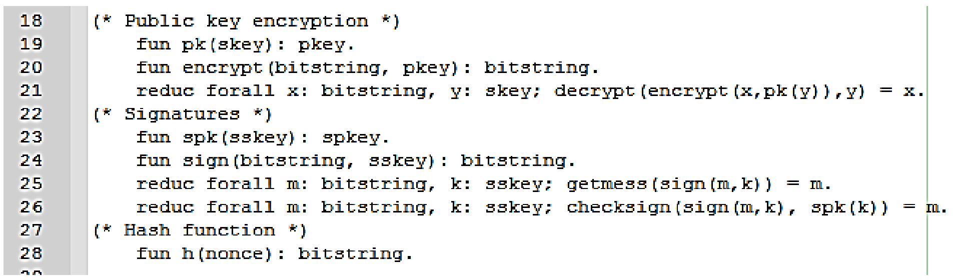
By (6.5), it is proven that the SP_v trusts the IdP_x on its public key k_{IdP_x}. Similarly, the IdP_x trusts SP_v on k_{SP_v} as public key by (6.13). By considering (6.16) and (6.18), a two-way authentication and sharing the secret keys of SP_v and IdP_x respectively is achieved. Based on them, the first requirement of protocol has been achieved, which is to build a trust relationship between SP_v and IdP_x dynamically. From (6.23), it is concluded that IdP_x believes that actor A_x uses the AR_x to prove the identity. Therefore, IdP_x verifies the actor A_x based on its identity and the secret nonce in AR_x. This leads to prove that the SP_v believes that IdP_x considers the AR_x as a true actor relationship and their verification result V is true as well, as in (6.26). Thus, by (6.26) the second requirement of the proposed protocol is achieved. As a result, by satisfying both of the requirements, the correctness of the protocol based on BAN logic is logically proven to establish the EA identity after building the required trust relationship between the participating agents.

6.2.Formal Verification of $GIdV$ Protocol with ProVerif

Model verification in the computing terminology refers *to "Substantiation that a model code is in some sense a true representation of a conceptual model within certain specified limits or ranges of application and corresponding ranges of accuracy"* [131]. A formal verification of any protocol is crucial to identify the hidden weakness and flaw. It is more effective than the informal verification. For instance, a formal verification method lead Lowe to discover the flaw of Needham-Schroeder PK protocol [121], although it passed an informal verification. Therefore, it is crucial for the proposed protocol to pass the formal verification to ensure the protocol design and implementation correctly meet the requirements.

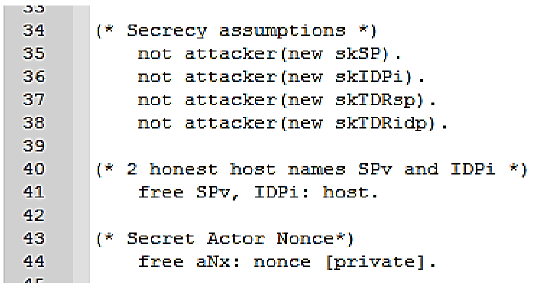
Currently, different verification tools are used formally to analyse and test the protocols security and correction. AVISPA tool [135], FDR [121], Scyther tool [136], ProVerif tool [137] are examples of the verification tools. However, these tools comparison is out of this research scope. The ProVarif tool has been chosen to prove the authenticity property of $GIdV$ because it is widely used for verifying authentication protocols [138]–[140]. Moreover, according to Cremers et al. [136] ProVerif is the fastest tool that checks the authenticity property. It has an active users community and good documentation. The following approach has been followed as a verification process: (1) Model the protocol formally; (2) Declare the protocol rules formally; (3) Formally declare the security goals to be checked; and (4) Select the ProVerif as an automated tool to verify the protocol authenticity property. Appendix G, gives an overview of the ProVerif tool.

To model the $GIdV$ protocol with Proverif using Pi calculus, a free channel (c) has been used to represent the public Internet communication. Four main entities are used to represent the entities SP_v, TDR_{SP_v}, TDR_{IdP_x}, and IdP_x; These entities are: SPv, TDRsp, TDRidp, and IdPi respectively. Each of them has the secure private key skSP, skIDP, skTDRsp, and skTDRidp respectively. Functions to encrypt the public key, signatures, shared secure key and decrypt them are defined. A hash function is defined as bellow.



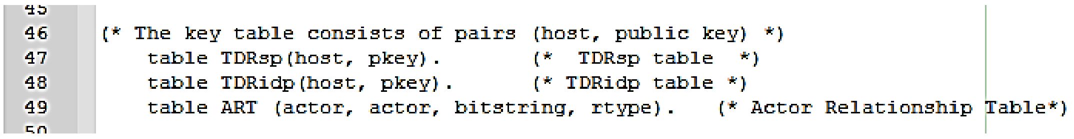
```
(* Public key encryption *)
    fun pk(skey): pkey.
    fun encrypt(bitstring, pkey): bitstring.
    reduc forall x: bitstring, y: skey; decrypt(encrypt(x,pk(y)),y) = x.
(* Signatures *)
    fun spk(sskey): spkey.
    fun sign(bitstring, sskey): bitstring.
    reduc forall m: bitstring, k: sskey; getmess(sign(m,k)) = m.
    reduc forall m: bitstring, k: sskey; checksign(sign(m,k), spk(k)) = m.
(* Hash function *)
    fun h(nonce): bitstring.
```

A set of secrecy assumptions have been set as follows. It is assumed that skSP, skIDP1, skTDRsp, and skTDRidp are up to date private key of the trusted agents SP, IDPi, TDRsp TDRidp respectively. It is assumed that actor nonce aNx is private and just the IDPi knows it.



```
(* Secrecy assumptions *)
    not attacker(new skSP).
    not attacker(new skIDPi).
    not attacker(new skTDRsp).
    not attacker(new skTDRidp).

(* 2 honest host names SPv and IDPi *)
    free SPv, IDPi: host.

(* Secret Actor Nonce*)
    free aNx: nonce [private].
```

Two databases have been defined as tables named TDRsp, TDRidp, that act as distributed *TDRs* servers to be used in the mutual authentication between SPv and IdPi. A third database named ART is defined to authenticate the IoT entity by IdPi.



```
(* The key table consists of pairs (host, public key) *)
    table TDRsp(host, pkey).        (*  TDRsp table  *)
    table TDRidp(host, pkey).       (* TDRidp table *)
    table ART (actor, actor, bitstring, rtype).   (* Actor Relationship Table*)
```

To model the *GIdV* process, they have to be defined as events in ProVerif. Thus, six event have been defined as below:

- event **AuthenticateIdPi:** The event represents that entity SPv authenticates the entity IdPi using the public key and a fresh nonce;
- event **AcceptSPv**: The event represents that entity IdPi accepts the entity SPv using the public key and a fresh nonce;
- event **AuthenticateSPv**: The event represents that entity IdPi authenticates the entity SPv using the public key and a fresh nonce;

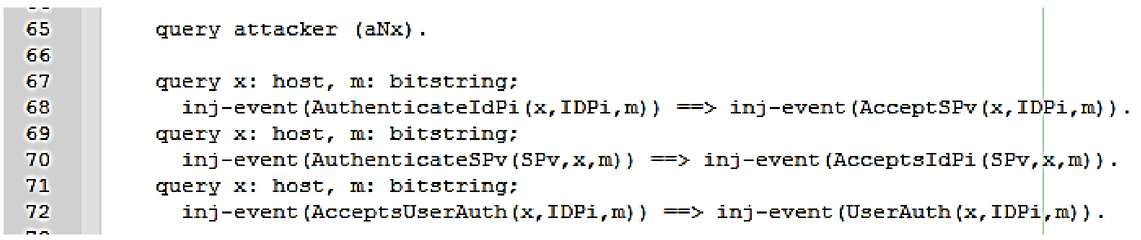
- event **AcceptsIdPi:** The event represents that entity SPv accepts the entity IdPi using the public key and a fresh nonce;
- event **AcceptsUserAuth**: The event represents that entity SPv authenticates the user based on the presented actor relationship attributes
- event **UserAuth**: The event represents that entity IdPi verify the actor relationship attributes based on its database.

A correspondence assertions have been used to represent the relationships between these events to study the required property using the form *"if an event e has been executed, then event e' has been previously executed.*" The syntax is used to represent the query:

$$\text{query } x1{:}\,T1, \ldots., xn{:}\,Tn; \text{ event } (e(M1, \ldots, Mj)) \text{ ==> event } \big(e'(N1, \ldots, Nk)\big)$$

Where, $M1, \ldots, Mj, N1, \ldots, Nk$ are terms generated by the application based on the arguments $x1, \ldots, xn$ of types $T1, \ldots, Tn$.

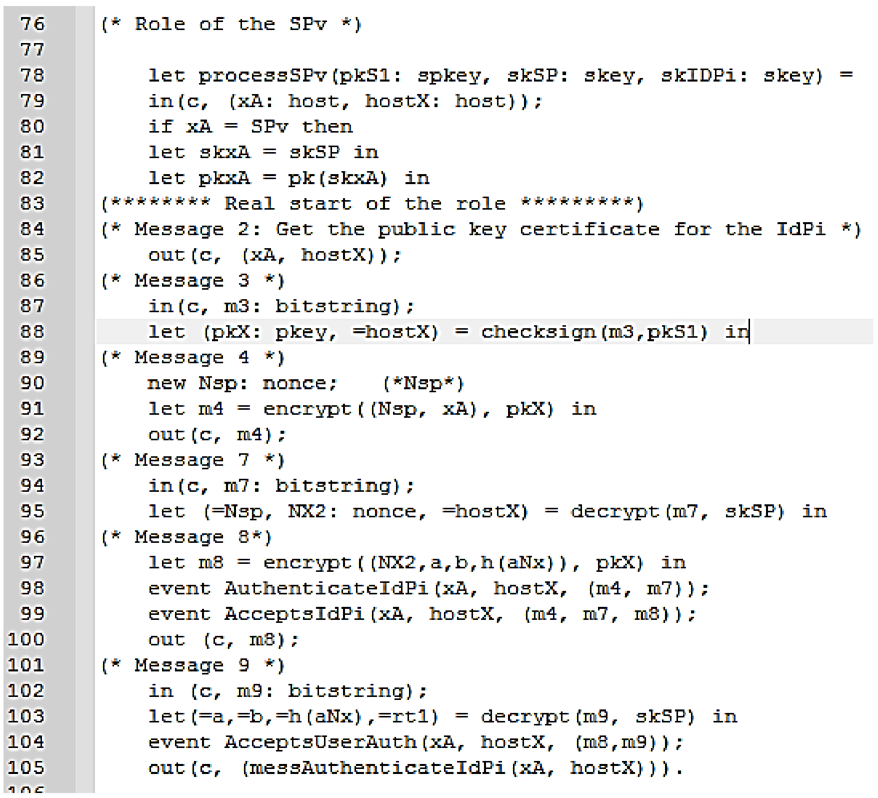
The $GIdV$ properties have been represented in the following queries.



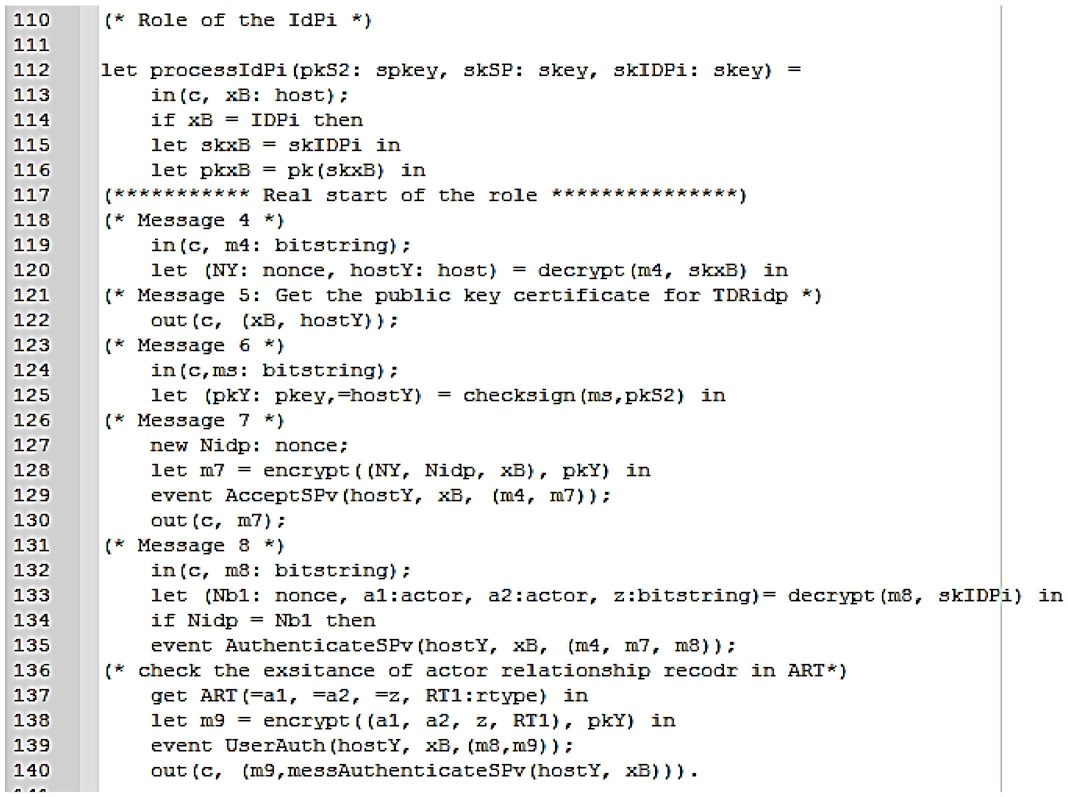
```
65      query attacker (aNx).
66
67      query x: host, m: bitstring;
68        inj-event(AuthenticateIdPi(x,IDPi,m)) ==> inj-event(AcceptSPv(x,IDPi,m)).
69      query x: host, m: bitstring;
70        inj-event(AuthenticateSPv(SPv,x,m)) ==> inj-event(AcceptsIdPi(SPv,x,m)).
71      query x: host, m: bitstring;
72        inj-event(AcceptsUserAuth(x,IDPi,m)) ==> inj-event(UserAuth(x,IDPi,m)).
```

The first query that is represented by line 65 is used to assure the secrecy of entity nonce (aNx). While the next two queries in lines 67 and 69 are used to check whether each entity SPv and IdPi are authenticate to each other. The last query that is represented by line 71 is used to represent the query of whether the user identity is verified by the intended IdP or not.

All the protocol messages stated in Section 4.2 are represented using the Pi calculus. The messages and events of the entities SPv and IdPi will be modelled as follows.

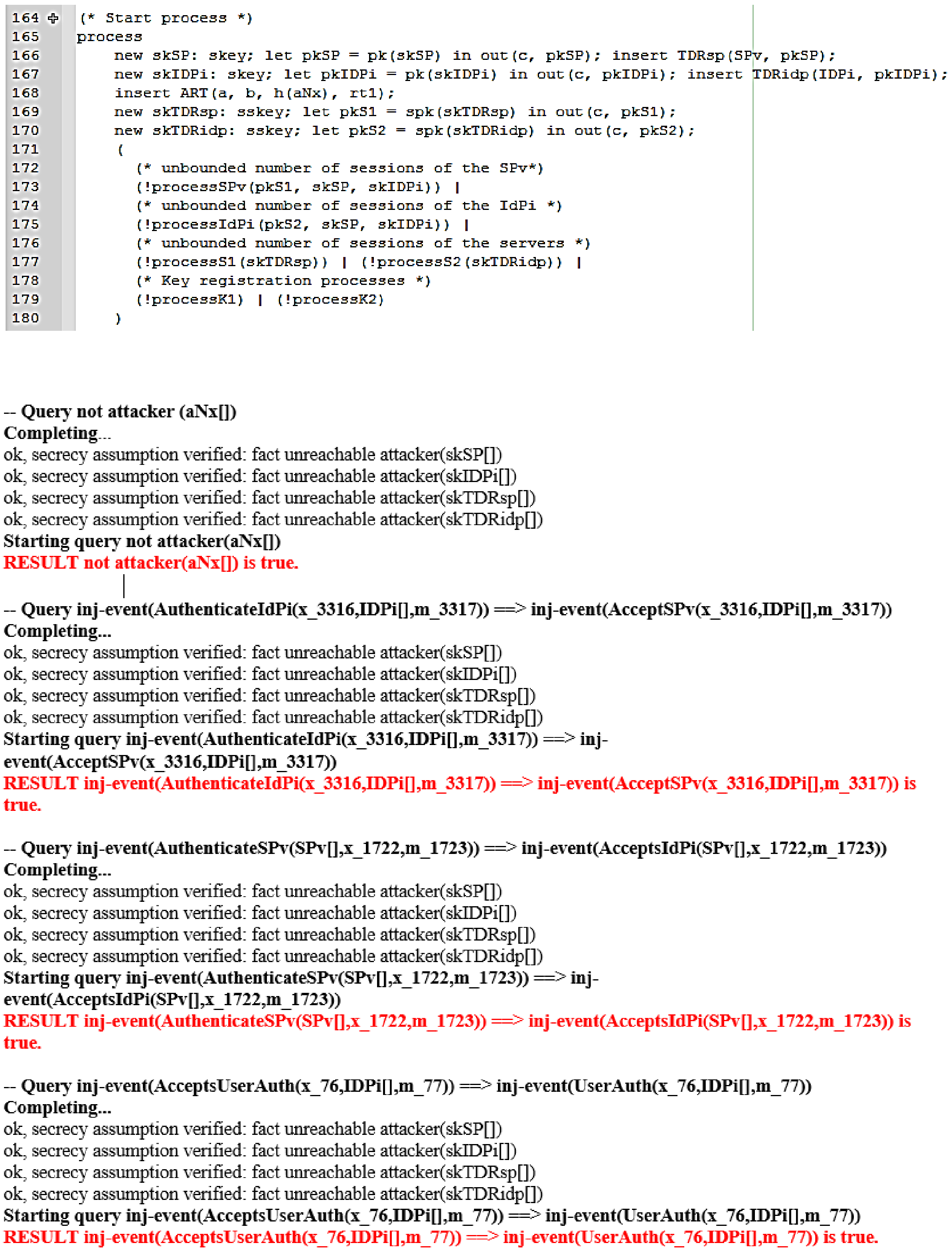


```
(* Role of the SPv *)

    let processSPv(pkS1: spkey, skSP: skey, skIDPi: skey) =
    in(c, (xA: host, hostX: host));
    if xA = SPv then
    let skxA = skSP in
    let pkxA = pk(skxA) in
(******** Real start of the role *********)
(* Message 2: Get the public key certificate for the IdPi *)
    out(c, (xA, hostX));
(* Message 3 *)
    in(c, m3: bitstring);
    let (pkX: pkey, =hostX) = checksign(m3,pkS1) in
(* Message 4 *)
    new Nsp: nonce;    (*Nsp*)
    let m4 = encrypt((Nsp, xA), pkX) in
    out(c, m4);
(* Message 7 *)
    in(c, m7: bitstring);
    let (=Nsp, NX2: nonce, =hostX) = decrypt(m7, skSP) in
(* Message 8*)
    let m8 = encrypt((NX2,a,b,h(aNx)), pkX) in
    event AuthenticateIdPi(xA, hostX, (m4, m7));
    event AcceptsIdPi(xA, hostX, (m4, m7, m8));
    out (c, m8);
(* Message 9 *)
    in (c, m9: bitstring);
    let(=a,=b,=h(aNx),=rt1) = decrypt(m9, skSP) in
    event AcceptsUserAuth(xA, hostX, (m8,m9));
    out(c, (messAuthenticateIdPi(xA, hostX))).
```



```
(* Role of the IdPi *)

let processIdPi(pkS2: spkey, skSP: skey, skIDPi: skey) =
    in(c, xB: host);
    if xB = IDPi then
    let skxB = skIDPi in
    let pkxB = pk(skxB) in
(*********** Real start of the role ***************)
(* Message 4 *)
    in(c, m4: bitstring);
    let (NY: nonce, hostY: host) = decrypt(m4, skxB) in
(* Message 5: Get the public key certificate for TDRidp *)
    out(c, (xB, hostY));
(* Message 6 *)
    in(c,ms: bitstring);
    let (pkY: pkey,=hostY) = checksign(ms,pkS2) in
(* Message 7 *)
    new Nidp: nonce;
    let m7 = encrypt((NY, Nidp, xB), pkY) in
    event AcceptSPv(hostY, xB, (m4, m7));
    out(c, m7);
(* Message 8 *)
    in(c, m8: bitstring);
    let (Nb1: nonce, a1:actor, a2:actor, z:bitstring)= decrypt(m8, skIDPi) in
    if Nidp = Nb1 then
    event AuthenticateSPv(hostY, xB, (m4, m7, m8));
(* check the exsitance of actor relationship recodr in ART*)
    get ART(=a1, =a2, =z, RT1:rtype) in
    let m9 = encrypt((a1, a2, z, RT1), pkY) in
    event UserAuth(hostY, xB,(m8,m9));
    out(c, (m9,messAuthenticateSPv(hostY, xB))).
```

The last part of the ProVarif is the process part that involves an unlimited number of sessions between SPv, IdPi, TDRspv, and TDRidp as follows.



```
(* Start process *)
process
    new skSP: skey; let pkSP = pk(skSP) in out(c, pkSP); insert TDRsp(SPv, pkSP);
    new skIDPi: skey; let pkIDPi = pk(skIDPi) in out(c, pkIDPi); insert TDRidp(IDPi, pkIDPi);
    insert ART(a, b, h(aNx), rt1);
    new skTDRsp: sskey; let pkS1 = spk(skTDRsp) in out(c, pkS1);
    new skTDRidp: sskey; let pkS2 = spk(skTDRidp) in out(c, pkS2);
    (
      (* unbounded number of sessions of the SPv*)
      (!processSPv(pkS1, skSP, skIDPi)) |
      (* unbounded number of sessions of the IdPi *)
      (!processIdPi(pkS2, skSP, skIDPi)) |
      (* unbounded number of sessions of the servers *)
      (!processS1(skTDRsp)) | (!processS2(skTDRidp)) |
      (* Key registration processes *)
      (!processK1) | (!processK2)
    )
```

```
-- Query not attacker (aNx[])
Completing...
ok, secrecy assumption verified: fact unreachable attacker(skSP[])
ok, secrecy assumption verified: fact unreachable attacker(skIDPi[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRsp[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRidp[])
Starting query not attacker(aNx[])
RESULT not attacker(aNx[]) is true.

-- Query inj-event(AuthenticateIdPi(x_3316,IDPi[],m_3317)) ==> inj-event(AcceptSPv(x_3316,IDPi[],m_3317))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(skSP[])
ok, secrecy assumption verified: fact unreachable attacker(skIDPi[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRsp[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRidp[])
Starting query inj-event(AuthenticateIdPi(x_3316,IDPi[],m_3317)) ==> inj-
event(AcceptSPv(x_3316,IDPi[],m_3317))
RESULT inj-event(AuthenticateIdPi(x_3316,IDPi[],m_3317)) ==> inj-event(AcceptSPv(x_3316,IDPi[],m_3317)) is
true.

-- Query inj-event(AuthenticateSPv(SPv[],x_1722,m_1723)) ==> inj-event(AcceptsIdPi(SPv[],x_1722,m_1723))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(skSP[])
ok, secrecy assumption verified: fact unreachable attacker(skIDPi[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRsp[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRidp[])
Starting query inj-event(AuthenticateSPv(SPv[],x_1722,m_1723)) ==> inj-
event(AcceptsIdPi(SPv[],x_1722,m_1723))
RESULT inj-event(AuthenticateSPv(SPv[],x_1722,m_1723)) ==> inj-event(AcceptsIdPi(SPv[],x_1722,m_1723)) is
true.

-- Query inj-event(AcceptsUserAuth(x_76,IDPi[],m_77)) ==> inj-event(UserAuth(x_76,IDPi[],m_77))
Completing...
ok, secrecy assumption verified: fact unreachable attacker(skSP[])
ok, secrecy assumption verified: fact unreachable attacker(skIDPi[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRsp[])
ok, secrecy assumption verified: fact unreachable attacker(skTDRidp[])
Starting query inj-event(AcceptsUserAuth(x_76,IDPi[],m_77)) ==> inj-event(UserAuth(x_76,IDPi[],m_77))
RESULT inj-event(AcceptsUserAuth(x_76,IDPi[],m_77)) ==> inj-event(UserAuth(x_76,IDPi[],m_77)) is true.
```

Figure 6-1: The $GIdV$ Verification Results with the ProVarif

The $GIdV$ verification results with ProVarif 1.96 prove that the protocol passes the verification as seen in Figure 6-1. Each query result is “true”, that means the query successfully passes the verification process. From the first query, it could be assured that the actor relationship nonce (aNx[]) is secured and there is no attacker that could revile it under the assumption the IdPi is the only entity that knows its value. The second and third queries show that the SPv authenticates the IdPi and vice versa. Therefore, it is concluded that the trust relationship could be achieved based on a trusted 3rd parties, i.e. TDRsd and TDRidp. Finally the last query proves the actor relationship attributes, which is used by the IdPi leads to verify the actor identity, hence, the SPv could rely on the trust relationship that resulted from the mutual authentication to accept the user authentication. Therefore, it could be concluded that the $GIdV$ protocol is working and secure against the simulated attacks by the ProVerif tool.

6.3. Critical Evaluation of The $GIdM$

This section evaluates the proposed $GIdM$ based on the elicited requirements in Section 3.1 as evaluation factors. The first three requirements, i.e. **Req.1**, **Req.2**, and **Req.3**, are related to the identification schema that will be used to evaluate the proposed identifier $GARI$. The rest, i.e. **Req.4 – Req.8**, will be used to evaluate the proposed $GIdM$.

Table 6-2: The Criteria Explanation for Identifiers Evaluation

Criteria	Description
Cr. 1	***User identifier.*** Identify the EA's identifier
Cr. 2	***Device identifier.*** Identify the communication device identifier
Cr. 3	***User domain / IdP.*** Identify the EA identifier domain/IdP identifier
Cr. 4	***Device domain/IdP.*** Identify the communication device's identifier domain/IdP identifier
Cr. 5	***Usability across-domain.*** Identify the usability of the identifier by a mobile entity across the home domain.
Cr. 6	***The Actor type.*** Identify the actor type, i.e. **P**erson, **D**evice, **A**pplication, or **S**ervice.
Cr. 7	***Internet connectivity.*** Identify the ability of the communication device to access the Internet
Cr. 8	***Actor Relationship.*** Identify the relationship type between user-device and device-device.

6.3.1. The Global Actor Relationship Identifier Evaluation

In this section the proposed identifier by this research, i.e. the $GARI$, will be evaluated based on its perceived benefits in comparison to other identifiers, listed in Section 2.3.2. The **Req.1, Req.2, and Req.3** will be considered as a basis to derive the evaluation criteria. Table 6-2 illustrates the explanation of the evaluation criteria. Mapping these criteria with the EA identity establishment requirements are illustrated in Table 6-3.

Table 6-3 Mapping the Evaluation Criteria with Requirements

Criteria	Requirement							
	Cr. 1	Cr. 2	Cr. 3	Cr. 4	Cr. 5	Cr. 6	Cr. 7	Cr. 8
Req.1	●	●						
Req.2			●	●	●			
Req.3						●	●	●

Table 6-4: The State of the Art of Identifiers Comparison

Criteria	Identifier Proposals					
	Liu et. al. [69]	Batalla et. al. [70]	Mahalle et. al. [71]	Butkus [64]	Zdravkova [72]	*GARI*
Cr. 1				✓	✓	✓
Cr. 2	✓	✓	✓	✓	✓	✓
Cr. 3				✓	✓	✓
Cr. 4	✓		✓	✓	✓	✓
Cr. 5			✓	✓	✓	✓
Cr. 6	D	D	D/S	✓	✓	✓
Cr. 7						✓
Cr. 8						✓
✓: fulfilled; D: device's identity; S: service's identity; D/S: device or service identity						

The comparison between existing identifier proposals and *GARI* is presented in Table 6-4. The table shows that almost all of the proposals encompass the *device identifier* and its registration *IdP (or namespace)* information, which is represented by Cr.2 and Cr.4 respectively. However, identifying the user type along with the device identifier is proposed only by three identifier formats by Butkus, Zdravkova, and the *GARI* as shown by Cr.1. Moreover, Cr.6 shows that identifying the *EA*, as the actual user of devices in IoT, has less interest by research community compared with identifying devices (D) or services (S). Thus, identifying the user domain/*IdP* in these proposals is missing from them as shown by Cr.3. Cr.5 indicates that the ability of using an identifier by a mobile user/device to get services across-domain are supported by the formats proposed by Mahalle et. al., Butkus, Zdravkova, and *GARI*; while the other proposals can only support the domain identification.

Interestingly, all existing methods ignored the *Internet connectivity type* of the communicated device as shown by Cr.7, hence this gives an indication that they are neglecting to identify a large community of tiny objects that inter-connected using heterogeneous technologies to offer/get services in the IoT environment. Finally, all existing proposals lack any information related to the *actor relationships* as shown by Cr.8. In other words, they did not consider the verity of relationship types in the IoT environment and its impact on identifying the effective actor. By specifying the actor's relationships in $GARI$, SPs will be able to identify the $IdPs$ to be used later in the identification of the effective actor, based on the *actor relationship type* and their identity attributes. Therefore, the existing identifiers are unable to identify passive objects globally in comparison with $GARI$.

To sum up, existing proposals in IoT fail to meet the whole criteria as compared with $GARI$, as discussed above. It will not be possible for the SP to make a distinction between the EA and who makes a connection on behalf of him by using the existing proposals. In comparison, $GARI$ makes it possible by using the ARS options, and all actors identification data. Therefore, the $GARI$ is the most suitable format to use in the IoT environment to identify the EA.

6.3.2. The $GIdM$ Evaluation

The IoT is an open environment where the services are requested by any actor, from anywhere, and using any communication object. The semantic format of $GARI$ will help the SPs to get the actor identification attributes along with the relationship attributes of the interacted actors. In the previous section, ***Req.1 – Req.3*** were used to evaluate the proposed identifier $GARI$ that will be used by entities in the $GIdM$ as the actor identifier. This section evaluates the proposed $GIdM$ by comparing it with the IdMs solutions from the initiatives and academic researchers that were presented in Section 2.6 using the requirements ***(Req.4 – Req.8)*** in Section 3.1 as evaluation factors. Table 6-5 illustrates the comparison of $GIdM$ with these $IdMs$ solutions.

Delegation of the actors' identities requirement, i.e. ***Req.4***, supports the hybrid IdM model where the user and object/device (actors) control their identities when they interact with each other. ***Req.4*** is fully satisfied only by five $IdMs$ that are from the Higgins, Chibelushi, et al., Butkus, Zdravkova and the $GIdM$, while the others consider the user or object identity.

Similarly, ***Req.5*** which is the $IdPs$ awareness of the actor relationship is considered by the Higgins project from initiatives and the solutions of Chibelushi et al., Zdravkova, Abreu et al, and the $GIdM$ from the academic projects. However, the other solutions do not identify the actor's relationship concept in a general form nor consider the alternate and vanish possibility of these relationships.

Table 6-5: Evaluation of $IdMs$ Initiatives and Research Projects in the IoT

Initiative and research projects	**Requirements to establish the *EA* identity**				
	Req.4	**Req.5**	**Req.6**	**Req.7**	**Req.8**
Liberty alliance [84]	U				
Shibboleth [88]	U				
OpenID [90]	U				
Higgins [92]	✓	✓			
OAuth2.0 [93]	U				
PICOS [95]	U				
STROK [97]	U				
Mahalle [71]	U/O				
Chibelushi, et. al. [83]	✓	✓			
Butkus [64]	✓				
Zdravkova [72]	✓	✓			
Abreu et. al. [98]	U	✓			
Bernabe et. al. [99]	U/O				
$GIdM$	✓	✓	✓	✓	✓

✓: fulfilled, **U:** user, **O:** Object; ***Req.4***: Actors' identity delegation; ***Req.5***: The IdP awareness of the actor relationship; ***Req.6:*** Dynamic trust relationship establishment; ***Req.7***: Relationship-based identity establishment; ***Req.8***: Effective protocol to share actor's attributes;

Interestingly, the state of the art IdM solutions fail to support ***Req.6, Req.7**, and **Req.8***. They rely on a static pre-established trust relationship between the communicated $SP(s)$ and $IdP(s)$ within a domain or CoT. In other words, the $SP(s)$ are not dynamically establishing a trust relationship with foreign $IdP(s)$ to verify the actor identity, hence the static form is not suitable for a large number of $SP(s)$ and $IdP(s)$ such as in IoT [119]. Moreover, the EA identity establishment based on the actor's relationship is missing from the state of the art $IdMs$. They are built based on a fixed relationship between the actors, i.e. user and device, without considering the other types of the actor's interaction. Finally, an effective protocol to exchange the attributes of actor relationship is missing as well in these $IdMs$. This is because the attributes themselves have never been introduced by current solutions. However, all these limitations have been discussed in $GIdM$, where the actors identities are represented explicitly in the $GARI$; it supports establishing the bidirectional trust relationship between unknown entities relying on a set of $TDRs$; and the efficiency of the proposed protocol has been approved with ProVerif as discussed earlier.

To sum up, the $GIdM$ comparison with the state of the art $IdMs$ proves that the $GIdM$ with the $GARI$ is the only solution that satisfies the whole requirements to establish the EA identity.

The requirements: dynamically establishing a trust relationship between unrelated SP and IdP, relationship-based identity establishment, and effective protocol to share the required attributes to establish the identity are totally missed in the other $IdMs$ than $GIdM$. Therefore, the $GIdM$ is the most suitable IdM that allows the SPs to establish the EA identity based on the actors' relationship(s) globally in the IoT environment.

6.4. Testing the Identity Establishment Using $GIdM$

This section discusses testing the EA identity establishment by the SP node in the designed $GIdM$ for the IoT environment. The validated testing environment, discussed in Section 6.2, has been used to test the identity establishment in the scenarios that were discussed in Section 6.3.4. In order to confirm that $GIdM$ is correctly establishing the identity, each scenario has been tested twice. These are denoted as $CorrectID$ test and $FakeID$ test. Moreover, the successful results from $CorrectID$ testing with scenario 2 and scenario 3, in Section 6.3, will give a strong indication of improving the $IdMs$ interoperability across-domain by relying on the $TRDs$ nodes. The second test, i.e. $FakeID$ test, aims to examine the ability of $GIdM$ to detect fake or incorrect data that is used to request a service or data. For that purpose, the $IdPs$ are loaded with fake nonce values as part of the actor's relationship attributes. A random formula has been used to generate the nonce values in both cases. In both, the successful identity establishment will be represented by a tick sign (✓), while the failed results will be represented by a cross sign ($\boldsymbol{x}$).

Table 6-6 summarizes the testing results of the identity establishment. On the one hand, from the table, it is clear that all scenarios have successfully passed $CorrectID$ test to identify the EA identity using the correct nonce values of the actor relationship attributes. Moreover, all the scenarios (7-24) with across-domain and hybrid domains interactions have successfully identified the identity. These scenarios indicate that the bidirectional trust relationship between unknown entities is established by relying on the TDR nodes in the $GIdM$. Thus, using the TDR nodes could lead to improving the $IdMs$ entities interoperability across-domain. On the second hand, all scenarios fail to establish the identity using the fake nonce values as could be seen in $FakeID$ test results. In other words, the $IdPs$ are able to detect the mismatching of presented nonce values in the identity verification request with the ones loaded in its databases records.

In summary, the test proves the efficiency of the developed $GARI$ and $GIdM$ to establish the identity in the simulated environment of the IoT.

Table 6-6:Identity Establishment Results

No. of the scenario	Domain Interaction type	Testing results	
		$CorrectID$ Test	$FakeID$ Test
1	Single D./CoT	✓	x
2	Single D./CoT	✓	x
3	Single D./CoT	✓	x
4	Single D./CoT	✓	x
5	Single D./CoT	✓	x
6	Single D./CoT	✓	x
7	Hybrid	✓	x
8	Hybrid	✓	x
9	Hybrid	✓	x
10	Hybrid	✓	x
11	Across-D.	✓	x
12	Across-D.	✓	x
13	Hybrid	✓	x
14	Hybrid	✓	x
15	Hybrid	✓	x
16	Hybrid	✓	x
17	Hybrid	✓	x
18	Hybrid	✓	x
19	Hybrid	✓	x
20	Hybrid	✓	x
21	Across-D.	✓	x
22	Across-D.	✓	x
23	Across-D.	✓	x
24	Across-D.	✓	x

6.5.The Overhead Analysis of the $GIdV$ Communication and Computation

6.5.1. The Formulas of the Communication and Computation Costs

This section analyses the $GIdV$ in terms of communication and computation costs. These criteria are commonly used to evaluate the performance of newly designed protocols. However, it is difficult to compare the $GIdV$ with other works based on the criteria because the proposed solution by this research targets the IoT environment in general without considering a specific application domain or scenario. Therefore, they are discussed with the aim of formulating general formulas to estimate the costs in any scenario. As discussed in Section 4.3, the identity establishment using the $GIdV$ is composed of two main stages. Firstly, to establish a trust relationship between the visited SP and the foreign $IdP(s)$ that will follows the NSL public key protocol to be established. Secondly, to verify the EA identity by the trusted $IdP(s)$. Table 6-7 illustrates a summary of these costs that will be computed as follows.

The communication cost of the $GIdV$ will be estimated using tqhe number of messages required to perform the identity establishment. Establishing the trust relationship(s) between unknown agents requires the interconnection with the $TDRs$ nodes. These processes will cost (6) messages to establish a mutual trust relationship between the nodes SP and IdP. On the other hand, the number of required messages to verify the actor identity are varied based on the type of actors' interaction in the relationship(s) that are either direct or transitive. The direct interaction could cost a message or two as shown by $(Msgs_{IV1})$ in equation 6.27, while the transitive interaction could cost two messages per TR or nothing as shown in equation 6.28.

$$No.of\ Msgs_{IV1} = \begin{cases} 1\,, & if\ T(AR) = 0 \\ 2\,, & Otherwise \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.27)$$

$$No.of\ Msgs_{IV2} = \begin{cases} 2\,, & if\ TR = 1 \\ 0\,, & Otherwise \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.28)$$

The total number of messages $(Msgs_{total})$ will be computed by counting the cost of establishing the trust relationship(s) with the cost of identity verification using the following relation.

$$Msgs_{total} = 6 * n + Msgs_{IV1} + \sum_0^r Msgs_{IV2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.29)$$

$where,$

n: The number of foreign $IdPs$ that required establishing trust relationship

$r\ = 0, \ldots,$ Maximum number of transitive relationship.

The computation cost of the $GIdV$ will be estimated using the computation time measured by millisecond (ms). The establishing time of the trust relationship(s) requires to perform a PKI encryption/decryption and select an agent nonce that is denoted as $T_{E/D}$ and $T_{h(rand)}$ respectively. Each trust relationship cost with unknown IdP process cost, denoted by T_{trust}, will count as follows: $4T_{E/D}$ to agent's authentication ($2T_{E/D}$ per agent) and $2T_{h(rand)}$ to compute the hashing value of the generated nonce for the agents per a trust relationship. Therefore, the computation time to establish a trust relationship is $(4T_{E/D} + 2T_{h(rand)})$. The cost of verifying the actor is denoted by T_{IV}. Each identity verification message by the actor

IdP, explained in Section 5.2.6.3, will cost $(4T_{E/D})$, where each of the request and answer cost is $(2T_{E/D})$.

$$T_{total} = n * T_{trust} + m * T_{IV} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (7.30)$$

$where,$

n: The number of foreign $IdPs$ that require establishing a trust relationship

$n = 0, \ldots, x$

m: The number of actor identity verification messages, m$= 1, \ldots, y$

Table 6-7: A Summary of Computation and Communication Costs

Process	**Overhead**	
	Computation cost (T)	**No. of messages ($Msgs$)**
Establishing an agent trust relationship	$4T_{E/D}$	6
Hashing an agent nonce value	$T_{h(N)}$	0
An actor identity verification*	$4T_{E/D}$	0/1/2

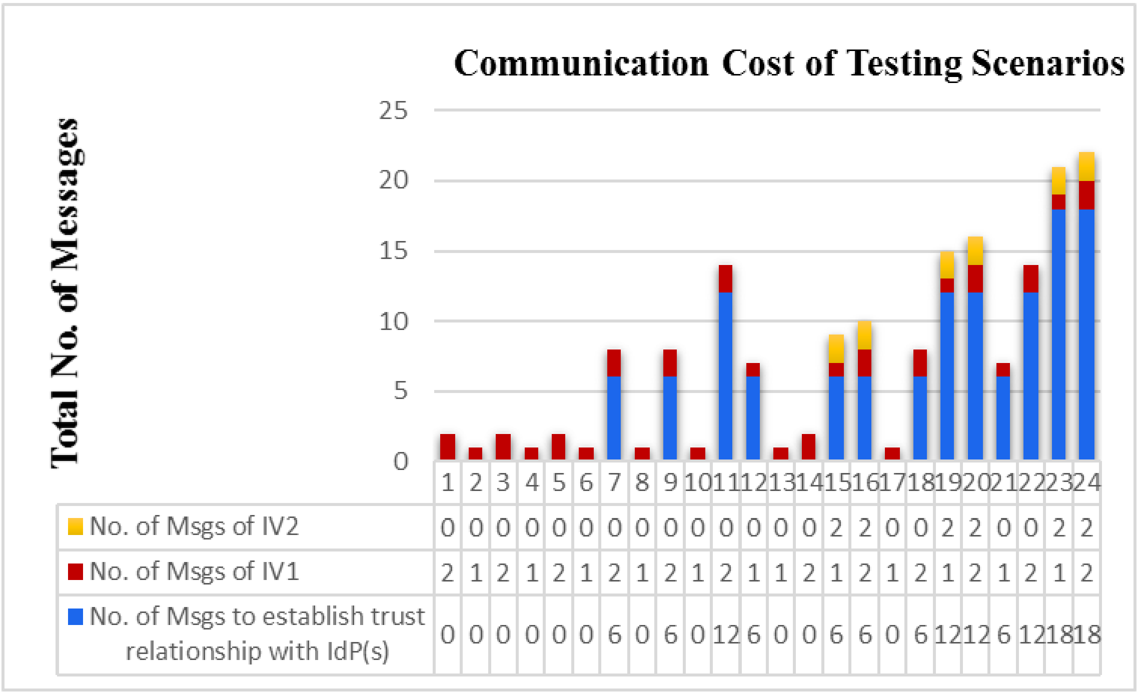
[Note: "*" The number of an actor identity verification massages are deduced from equations 6.27 and 6.28 respectively]

6.5.2. Discussion

This section discusses the communication and computation overhead of testing the identity establishment based on the testing scenarios from the previous section.

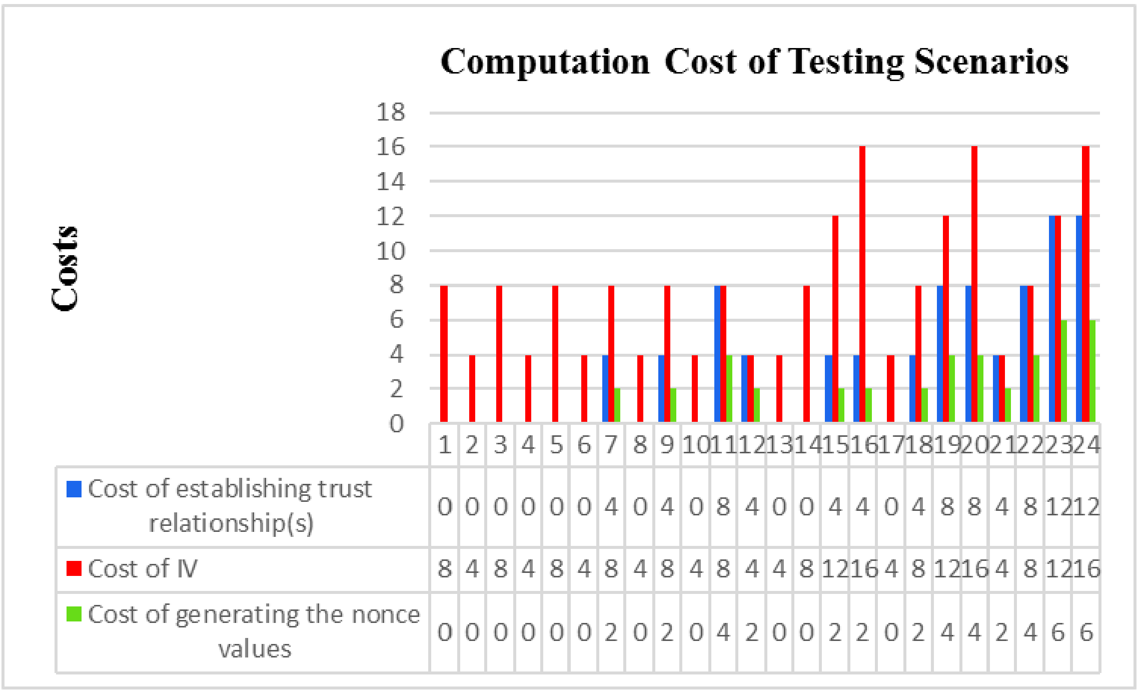
On the one hand, it is clear that the most communication overheads result from the process to establish a trust relationship between the unknown SP and $IdP(s)$ as can be seen in the blue bars in Figure 6-2. This is because each trust relationship costs 6 messages as seen in scenarios (7, 9, 12, 15, 16, 18, and 21). These values will be double in scenarios (11, 19, 20, and 22) or triple like in scenarios (23, 24) because the number of foreign $IdPs$ will be 2 and 3 respectively. Moreover, using the transitive relationship criterion as an additional IdP will improve the confidence of the EA identity, but shows additional computational overhead as seen in scenarios (15, 16, 19, 20, 23, and 24). On the other hand, the communication cost without considering these relationships shows a little overheads like in scenarios (1-6). This could represent the standard cases where the identity establishment will be done within the domain or a CoT because the trust relationship already exists. Therefore, it is concluded that the $GIdV$

communication overhead will be increased when more foreign $IdPs$ are involved in the identity establishment processes in the IoT environment.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
🟨 No. of Msgs of IV2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	2	2	0	0	2	2
🟥 No. of Msgs of IV1	2	1	2	1	2	1	2	1	2	1	2	1	1	2	1	2	1	2	1	2	1	2	1	2
🟦 No. of Msgs to establish trust relationship with IdP(s)	0	0	0	0	0	0	6	0	6	0	12	6	0	0	6	6	0	6	12	12	6	12	18	18

Figure 6-2: A Summary of Communication Cost of Testing Scenarios



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
🟦 Cost of establishing trust relationship(s)	0	0	0	0	0	0	4	0	4	0	8	4	0	0	4	4	0	4	8	8	4	8	12	12
🟥 Cost of IV	8	4	8	4	8	4	8	4	8	4	8	4	4	8	12	16	4	8	12	16	4	8	12	16
🟩 Cost of generating the nonce values	0	0	0	0	0	0	2	0	2	0	4	2	0	0	2	2	0	2	4	4	2	4	6	6

Figure 6-3: A Summary of Computation Cost of Testing Scenarios

The analysis of computational costs of testing the $GIdV$ using the same testing scenarios will be represented by red bars in Figure 6-3. As discussed earlier, the total cost is the count of times required to process the NSL encryptions/decryptions and to perform the hashing of the generated nonce value(s) if needed. It is recognised that the most computational overheads result from the actor identity verification process through all scenarios. The lowest costs are shown in the scenarios where a single identity verification message is required to process like in scenarios (2, 4, 6, 8, and 21). These figures multiply depending on the number of messages required. For instance, the computational overhead figures in the scenarios (16, 20, and 24) become four times the cost of a single message cost. This is because the direct relationship is of permanent type that requires two IV messages; and the existence of a transitive relationship adds additional two IV messages.

Additional computational overheads are shown for verifying the unknown domains in the IoT. It requires $4T_{E/D}$ and $2T_{h(N)}$ to establish an additional trust relationship between the SP and the foreign IdP such as in scenario no. (12). These figures are also multiplied based on the number of foreign $IdP(s)$ that are involved in the identity verification such as in scenarios (11, 19, 20, 22, 23, and 24). The highest computational overheads is $(28T_{E/D} + 6T_{h(N)})$ as shown by scenario no. (24). This overhead results from the processes of trusting three foreign $IdPs$, the processes of generating and hashing six nonce values, and four IV messages. The second high overheads are seen by scenarios (16, 19/20, and 23) with values $(20T_{E/D} + 2T_{h(N)}), (20T_{E/D} + 4T_{h(N)}), (24T_{E/D} + 6T_{h(N)})$ respectively. Similarly, the smallest computational cost is $(4T_{E/D})$ to establish the identity form using a single IV message in some of the scenarios such as (17). Therefore, it is concluded that there is no single equation to compute the computational overhead of using the $GIdV$ to establish the actor identity in the IoT environment.

To sum up, the analysis of the communication and computation overheads show the existence of direct proportional relations between these overheads and the number of IV messages and the number of trust relationships that are required to be established with the foreign $IdP(s)$ to verifythe actor identity. Moreover, there are no general equations to compute these overheads to be used when comparing with other works.

6.6. Summary

This chapter presented the $GIdM$ evaluation process. The conceptual validation formally proves the authenticity criterion of $GIdV$ protocol using a mathematical tracing method. The formal verification proves the authenticity criterion of the proposed $GIdV$ protocol using the ProVerif tool. Furthermore, the proposed $GARI$ and the $GIdM$ are compared with the related work in the field. The comparisons prove that they satisfy the designed requirements that were not considered by the existing works. Moreover, they have successfully passed the intensive test using the basis scenarios that represent the possible actor interaction scenarios in IoT. The $GIdV$ protocol overhead is analysed in terms of the communicational and computational costs. The overhead analysis shows that the overhead is in a direct proportional relation with the number of IV messages and the number of foreign $IdP(s)$ that are involved to establish the effective actor identity.

Chapter 7.

Conclusions and Future Works

7.1.Conclusions

The IoT promises the world where every object has the ability to interoperate with others via the public Internet without human intervention to offer better services to human beings. In other words, objects are interconnected on behalf of the other beneficiaries of the communication that are their owner(s), administrator(s), or user(s), which are denoted as effective actor(s) EA in this research. The IoT environment implies all of these EAs and objects as IoT's actors. From the identity management (IdM) point of view, actors could have different identity attributes managed by different $IdMs$ in each domain they interact with. However, these $IdMs$ are not always interoperable with each other because they are designed on the enterprises demand, hence they often use various identity attributes and methods. That could affects the actors' identity establishment by any domain service provider (SP) in the IoT across their registration domain. Moreover, the objects attributers are widely used to determine their EAs identities as a secondary identity. However, the relationship between these actors are not always fixed, it may change or even vanish. That means, relying on these objects' identities to attribute its EA identity without considering the relationships might fail to truly identify the EA. As a result of this identification failure, the SPs in IoT will not be able to offer the right service to the EA. This situation may affects the realization of IoT services. Hence, it is important to consider them when identifying the EA of the communicated object. This research, therefore, addresses the service providers (SPs) difficulty to truly establish the EA identity behind the communicated objects to offer the right services in the IoT.

The research conclusions can be summarised in the following points.

- This thesis has presented a general actor interaction criteria in the IoT environment in terms of actor interaction, domain interaction, and the mode of interaction. The actors could participate in permanent, semi-permanent, or open-access interactions. Moreover, the actors could interact with each other within a single IdM domain/CoT, across-domain, or a hybrid of some related domains with other unrelated ones. These interactions could be done using a single interaction or multiple interactions.
- The thesis has argued the actor identity attributes in the IoT. It is concluded that the existence of four parameters is sufficient to establish the identity. These parameters are the actor type, Internet connectivity, identifier, and the identifier of IdP. The Internet connectivity of the communication object leads to identify whether a single actor relationship is enough to access the Internet or needs to establish an additional relationship, if it is of a passive object type. This will allow a broad range of tiny and passive objects to be part of the IoT and recognise them globally by following these relationships.
- The communication object identity is widely used as an alternative/secondary identity of the actual requester. The SPs in IoT have to truly identifying the actual requester rather than the identity of the object that communicates on its behalf. This is because the relationship is not always fixed; it is changeable or might be vanished at the end, which preventing the SPs to offer right services to the requester. Therefore, defining such relationship has a significant impact on truly identifying the actual actor of the communicated object.
- The main contribution of this thesis is formulating a new identifier that presents the actors identity attributes along with the actors relationship types in a semantic format. This identifier (called $GARI$) has been modelled using a novel mathematical model. The existing identifier proposals in IoT fail to meet the whole criteria as compared with $GARI$. Therefore, the $GARI$ is the most suitable identifier to use in the IoT environment to identify the actual requester behind the communicated object.
- In the developed $GIdM$ system, the SP could establish a dynamical trust relationship with the independent $IdP(s)$ by using the $TDRs$ nodes as trusted 3rd parties. The $TDRs$ are responsible for holding the trust information of the entities (SPs and $IdPs$) and sharing it with trusted requester entities and other $TDRs$. Thus, the SPs and $IdPs$ could start a trust relationship on the fly with each other based on the $TDRs$.
- The developed new IPv6 extensions help to achieve a syntactical data interoperability between the $GIdM$ entities, where all entities have the same interpretation of the shared

data using these extensions. Meanwhile, the $IdMs$ interoperability across-domain is achieved in the $GIdM$ through using the $GIdV$.

- The correctness of the $GIdV$ protocol to establish the EA identity in the IoT has been proved formally using the logic of authentication (the BAN logic). While, the ProVerif tool has proved the authenticity of the developed protocol ($GIdV$) and its security against the simulated attacks.
- The effectiveness of the developed $GARI$ and $GIdM$ to establish the actual actor identity across-domain has been proved using the basis scenarios of actor interaction in the IoT. Moreover, the $GIdM$ comparison with the state of the art $IdMs$ shows that the $GIdM$ with the $GARI$ is the most suitable IdM that allows the SPs to establish the EA identity based on the actors' relationship(s) globally in the IoT environment.

7.2.Limitations

There are few limitations that are listed below:

- The user's privacy is not considered in the developed $GIdM$ as the $IdPs$ have to be aware of all the user relationships with communication objects.
- The analysis of the communication and computational overhead of the $GIdV$ protocol shows direct proportional relations with the number of required messages to perform the identity verification. However, in spite of the effectiveness of Needham-Schroder-Lowe (NSL) public key infrastructure to establish a trust relationship based on 3rd parties, it has a high computational overhead as compared with other approaches. A further research is required to find a modified approach with minimum overhead.
- This research evaluates the proposed solution using a simulator environment with empirical data to prove the concept. Thus, the network overhead and performance were not covered in this research. It is important to evaluate the $GIdM$ impact on the network performance using real data and real scenarios.

7.3.Future Works

Though this research, some ideas are presented that could be listed as future research directions:

- Managing the entities (SP and IdP) trust for the $GIdM$. Trust management is another interesting field of study required to be integrated in the $GIdM$. In such open environments like the IoT, the uncertainty between these entities prevents the offering of the best services. However, there is some research in that direction, which is required to assess its compatibility with the $GIdM$ before employing them.
- Another direction of research is to assess the ability to predict the effective actor identity by the smart objects/devices. The smart objects/devices can use the neighbour discovery protocol to interoperate with each other to infer the EA identity. Thus, an effective application is required to be developed.

Publications

- A. Majeed and A. Al-Yasiri, “Formulating A Global Identifier Based on Actor Relationship for the Internet of Things,” in *3rd EAI International Conference on Safety and Security in Internet of Things*, Paris, France: Springer, Cham., 2017, pp. 79–91.
- A. Majeed and A. Al-Yasiri, “Consolidate the Identity Management Systems to identify the Effective Actor Based on the Actor Relationship for the Internet of Things,” in *3rd International Congress of Information and Communication Technology (ICICT 2018)*, 27-28 February 2018, London, UK.

References

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] S. Li, L. Da Xu, and S. Zhao, “The Internet of Things: a survey,” *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, Apr. 2015.

[3] S. Forsström, V. Kardeby, P. Österberg, and U. Jennehag, “Challenges when Realizing a Fully Distributed Internet-of-Things – How we Created the SensibleThings Platform,” *ICDT 2014 Ninth Int. Conf. Digit. Telecommun. Challenges*, pp. 13–18, 2014.

[4] K. Ashton, “That ‘Internet of Things’ Thing,” *RFiD J.*, vol. 22, no. 7, pp. 97–114, Jun. 2009.

[5] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of Things security: A survey,” *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017.

[6] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[7] D. van Thuan, P. Butkus, and D. van Thanh, “A User Centric Identity Management for Internet of Things,” in *2014 International Conference on IT Convergence and Security (ICITCS)*, 2014, pp. 1–4.

[8] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[9] M. Abomhara and G. M. Koien, “Security and privacy in the Internet of Things: Current status and open issues,” in *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*, 2014, pp. 1–8.

[10] O. Vermesan and P. Friess, “Internet of Things – From Research and Innovation to Market Deployment.” River Publishers, Denmark, 2014.

[11] Gartner, “Build Your Blueprint for the Internet of Things , Based on Five Architecture Styles - (G00269736),” 2014.

[12] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of Things: Vision, Applications and Research Challenges,” *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[13] V. Vujovi and M. Maksimovi, “Resource : A Connection Between Internet of Things and Resource-Oriented Architecture,” in *Smart SysTech 2015; European Conference on Smart Objects, Systems and Technologies;*, 2015, pp. 1–7.

[14] D. Singh, G. Tripathi, and A. J. Jara, “A Survey of Internet-of-Things: Future Vision, Architecture, Challenges and Services,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 287–292.

[15] P. P. Ray, “A Survey on Internet of Things Architectures,” *J. King Saud Univ. - Comput. Inf. Sci.*, Oct. 2016.

[16] S. Jaiswal and D. Gupta, “Security Requirements for Internet of Things (IoT),” in *International Conference on Communication and Networks, Advances in Intelligent Systems and Computing*, vol. 508, N. Modi, P. Verma, and B. Trivedi, Eds. Singapore: Springer Nature, 2017, pp. 419–427.

[17] O. Bello, S. Zeadally, and S. Member, “Intelligent Device-to-Device Communication in the Internet of Things,” *Syst. Journal, IEEE*, no. 99, pp. 1–11, 2015.

[18] A. Serbanati, C. M. Medaglia, and U. B. Ceipidor, “Building Blocks of the Internet of Things : State of the Art and Beyond,” in *Deploying RFID – Challenges, Solutions, and Open Issues*, C. Turcu, Ed. InTech, 2011.

[19] C. Jardak and J. W. Walewski, *Enabling Things to Talk-Designing IoT solutions with the IoT Architectural Reference Model*. Springer Heidelberg, 2013.

[20] A. K. Ranjan and G. Somani, “Access Control and Authentication in the Internet of Things Environment,” in *Connectivity Frameworks for Smart Devices: The Internet of Things from a Distributed Computing Perspective*, Z. Mahmood, Ed. Cham: Springer International Publishing, 2016, pp. 283–305.

[21] H. Chaouchi, T. Bourgeau, and P. Kirci, *Next-Generation Wireless Technologies*. London: Springer London, 2013.

[22] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, “Choices for Interaction with Things on Internet and Underlying Issues,” *Ad Hoc Networks*, vol. 28, pp. 68–90, May 2015.

[23] Yuan Jie Fan, Yue Hong Yin, Li Da Xu, Yan Zeng, and Fan Wu, “IoT-Based Smart Rehabilitation System,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1568–1577, May 2014.

[24] ITU-T Rec. Y.2720, “NGN Identity Management Framework Recommendation,” 2009.

[25] Yuan Cao and Lin Yang, “A survey of Identity Management technology,” in *2010 IEEE International Conference on Information Theory and Information Security*, 2010, pp. 287–293.

[26] A. Fongen, “Identity Management and Integrity Protection in the Internet of Things,” in *2012 Third International Conference on Emerging Security Technologies*, 2012, pp. 111–114.

[27] R. Yeluri and E. Castro-Leon, *Identity Management and Control for Clouds*, no. IdM. 2014.

[28] G. Alpár, J.-H. Hoepman, and J. Siljee, “The Identity Crisis. Security, Privacy and Usability Issues in Identity Management,” *arXiv Prepr. arXiv1101.0427*, Jan. 2011.

[29] P. Angin *et al.*, “An Entity-Centric Approach for Privacy and Identity Management in Cloud Computing,” in *2010 29th IEEE Symposium on Reliable Distributed Systems*, 2010, pp. 177–183.

[30] A. Jøsang and J. Golbeck, “Challenges for Robust Trust and Reputation Systems,” Saint Malo, France, 2009.

[31] K. Lampropoulos and S. Denazis, “Identity Management Directions in Future Internet,” *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 74–83, Dec. 2011.

[32] K. Lampropoulos and S. Denazis, “Introducing a Cross Federation Identity Solution for Converged Network Environments,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1–13.

[33] D. Evans, “The Internet of Things - How the Next Evolution of the Internet is Changing Everything,” *CISCO white Pap.*, no. April, pp. 1–11, 2011.

[34] I. Friese, J. Heuer, and N. Kong, “Challenges from the Identities of Things - Introduction of the Identities of Things Discussion Group within Kantara Initiative Ingo,” in *IEEE World Forum on Internet of Things (WF-IoT) Challenges*, 2014, pp. 1–4.

[35] E. Avelar, L. Marques, D. dos Passos, R. Macedo, K. Dias, and M. Nogueira, “Interoperability Issues on Heterogeneous Wireless Communication for Smart Cities,” *Comput. Commun.*, vol. 58, pp. 4–15, Mar. 2015.

[36] I. Bojic, J. Granjal, E. Monteiro, and D. Katusic, *Wireless Networking for Moving Objects*, vol. 8611. Cham: Springer International Publishing, 2014.

[37] S.-H. K. S.-H. Kim, S.-H. J. S.-H. Jin, and H.-J. L. H.-J. Lim, “A Concept of

Interoperable Authentication Framework for dynamic relationship in Identity Management," *Adv. Commun. Technol. (ICACT), 2010 12th Int. Conf.*, vol. 2, pp. 1635–1639, 2010.

[38] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, "Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare," *Wirel. Commun. Mob. Comput.*, vol. 2017, pp. 1–10, 2017.

[39] Guangyi Xiao, Jingzhi Guo, Li Da Xu, and Zhiguo Gong, "User Interoperability With Heterogeneous IoT Devices Through Transformation," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1486–1496, May 2014.

[40] S. Soursos, I. P. Zarko, P. Zwickl, I. Gojmerac, G. Bianchi, and G. Carrozzo, "Towards the Cross-Domain Interoperability of IoT Platforms," in *2016 European Conference on Networks and Communications (EuCNC)*, 2016, pp. 398–402.

[41] Gartner, "The Identity of Things for the Internet of Things - (G00270277)," 2015.

[42] Forgerock, "Whitepaper: The Identity of Things (IDoT): Access Management (IAM) Reference Architecture for The Internet of Things (IoT)," *Forg. white Pap.*, 2015.

[43] P. Mahalle and P. Railkar, *Identity Management for Internet of Things*. Denmark: River Publishers, 2015.

[44] K. Lam and C. Chi, *Identity in the Internet-of-Things (IoT): New Challenges and Opportunities*, vol. 9977. Cham: Springer International Publishing, 2016.

[45] E. Bertino, K.-K. R. Choo, D. Georgakopolous, and S. Nepal, "Internet of Things (IoT): Smart and Secure Service Delivery," *ACM Trans. Internet Technol.*, vol. 16, no. 4, pp. 1–7, Dec. 2016.

[46] Kantara Intitative, "Concepts of Identity within the Internet of Things," 2014.

[47] J. B. Bernabe, J. L. Hernández, M. V. Moreno, and A. F. Skarmeta, "Privacy-Preserving Security Framework for a Social-Aware Internet of Things," in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, Ramón Hervás, S. Lee, Chris Nugent, and J. Bravo, Eds. Springer International Publishing, 2014, pp. 408–415.

[48] Happiest Minds Technologies Pvt., "Identity Relationship Management," *Happiest Mind white Pap.*, no. November, 2013.

[49] A. Mordeno and B. Russell, "Identity and Access Management for the Internet of Things

- Summary Guidance," *CSA – Cloud Secur. Alliance*, 2015.

[50] N. Benamar, A. Jara, L. Ladid, and D. El Ouadghiri, "Challenges of the Internet of Things: IPv6 and Network Management," in *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2014, vol. 4, no. 3, pp. 328–333.

[51] S. Clauß and M. Köhntopp, "Identity Management and its Support of Multilateral Security," *Comput. Networks*, vol. 37, no. 2, pp. 205–219, Oct. 2001.

[52] E. Bertino, F. Paci, and N. Shang, "Keynote 2: Digital Identity Protection - Concepts and Issues," in *2009 International Conference on Availability, Reliability and Security*, 2009, pp. lxix–lxxviii.

[53] J. H. Pérez, "Internet Infodiversity: State of the Art and Future Trends," *Qual. Quant. Methods Libr.*, vol. 2, no. 4, pp. 479–485, 2013.

[54] G. Ben Ayed and S. Ghernaouti-Helie, "Digital Identity Management within Networked Information Systems: From Vertical Silos View into Horizontal User-Supremacy Processes Management," in *2011 14th International Conference on Network-Based Information Systems*, 2011, pp. 98–103.

[55] T. Nabeth, "Identity of Identity," in *The Future of Identity in the Information Society*, K. Rannenberg, D. Royer, and A. Deuker, Eds. Springer, 2009, pp. 19–70.

[56] J. Torres, M. Nogueira, and G. Pujolle, "A Survey on Identity Management for the Future Network," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 2, pp. 787–802, 2013.

[57] A. Pfitzmann and M. Hansen, "A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2010.

[58] E. Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*. Artech House, 2011.

[59] K. Cameron, "The laws of identity," *Microsoft Corp*, 2005.

[60] M. Kohli, "Transformation from Identity Stone Age to Digital Identity," *Int. J. Netw. Secur. Its Appl.*, vol. 3, no. 3, pp. 121–136, May 2011.

[61] ITU-T Rec. X.1250, "Baseline Capabilities for Enhanced Global Identity Management and Interoperability," 2009.

[62] IBM, "Building a Next-Generation Identity and Access Management Program," *IBM*

white Pap., 2014.

[63] M. Rouse, “Multifactor Authentication (MFA),” 2014. [Online]. Available: searchsecurity.techtarget.com/definition/multifactor-authentication-MFA. [Accessed: 12-Feb-2015].

[64] P. Butkus, “Identity Management in M2M Networks,” Master Thesis: Aalto University, 2014.

[65] CATR and IERC, “EU-China Joint White Paper on Internet of Thing Identification,” 2014.

[66] EU Framework7 project (CASAGRAS), “RFID and the Inclusive Model for the Internet of Things,” 2009.

[67] European Commission, “Conclusions of the Internet of Things public consultation - Internet of Things Factsheet Identification,” 2013.

[68] E. Jung, Y. Choi, J. S. Lee, and H. J. Kim, “An OID-Based Identifier Framework Supporting the Interoperability of Heterogeneous Identifiers,” in *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, 2012, pp. 304–308.

[69] C. H. Liu, B. Yang, and T. Liu, “Efficient Naming, Addressing and Profile Services in Internet-of-Things Sensory Environments,” *Ad Hoc Networks*, vol. 18, pp. 85–101, Jul. 2014.

[70] J. Mongay Batalla and P. Krawiec, “Conception of ID Layer Performance at the Network Level for Internet of Things,” *Pers. Ubiquitous Comput.*, vol. 18, no. 2, pp. 465–480, Feb. 2014.

[71] P. Mahalle, “Identity Management Framework for Internet of Things,” PhD Dissertation: Aalborg University, Denmark, 2013.

[72] V. Zdravkova, “Identity Management Approach in Internet of Things,” Master Thesis: Aalborg Ubiversity, Denmark, 2015.

[73] J. Pato, “Identity Management: Setting Context,” *Hewlett-Packard, Cambridge, MA*, 2003.

[74] G.-J. Ahn, M. N. Ko, and M. Shehab, “Portable User-Centric Identity Management,” in *Proceding of the IFIP TC 11 23rd International Information Security Conference, IFIP 20th World Computer Congress, IFIP SEC’08, September 7-10, 2008, Milano, Italy,*

2008, vol. 278, pp. 573–587.

[75] M. Alzomai, “Identity Management : Strengthening One-Time Password Authentication through Usability by,” Doctoral dissertation, Queensland University of Technology, 2011.

[76] M. S. Ferdous, “PhD thesis: User-controlled Identity Management Systems using mobile devices,” University of Glasgow, 2015.

[77] Y. Zuo, X. Luo, and F. Zeng, “Towards a Dynamic Federation Framework Based on SAML and Automated Trust Negotiation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6318 LNCS, no. M4D, 2010, pp. 254–262.

[78] R. Singh, V. Gupta, and K. Mohan, “Dynamic Federation in Identity Management for Securing and Sharing Personal Health Records in a Patient- centric Model in Cloud,” *Int. J. Eng. Technol.*, vol. 5, no. 3, pp. 2201–2209, 2013.

[79] R. Sánchez, F. Almenares, P. Arias, D. Díaz-Sánchez, and A. Marín, “Enhancing Privacy and Dynamic Federation in IdM for Consumer Cloud Computing,” *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 95–103, 2012.

[80] M. S. Ferdous and R. Poet, “Dynamic Identity Federation Using Security Assertion Markup Language (SAML),” Springer, 2013, pp. 131–146.

[81] C. M. Westphall and C. B. Westphall, “Towards Privacy in Identity Management Dynamic Federations,” in *ICN 2016 : The Fifteenth International Conference on Networks (includes SOFTNETWORKING 2016)*, 2016, pp. 40–45.

[82] H. Boujezza, M. AL-Mufti, H. K. Ben Ayed, and L. Saidane, “A Taxonomy of Identities Management Systems in IoT,” in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, 2015, pp. 1–8.

[83] C. Chibelushi, A. Eardley, and A. Arabo, “Identity Management in the Internet of Things : the Role of MANETs for Healthcare Applications,” vol. 1, no. 2, pp. 73–81, 2013.

[84] Liberty, “The Liberty Alliance Project.” [Online]. Available: http://projectliberty.org/. [Accessed: 15-Jan-2015].

[85] SAML, “Advancing Open Standards for the Information Society.” [Online]. Available: http://www.oasis-open.org/committees/tc_home.php?wg_a.

[86] H. Akram and M. Hoffmann, "Supports for Identity Management in Ambient Environments - The Hydra Approach," in *2008 Third International Conference on Systems and Networks Communications*, 2008, pp. 371–377.

[87] "Kantara Intiative." [Online]. Available: https://kantarainitiative.org/about/.

[88] Shibboleth, "The Shibboleth Project." [Online]. Available: http://shibboleth.net/about/. [Accessed: 15-Jan-2015].

[89] W. A. Alrodhan, "PhD Thesis: Privacy and Practicality of Identity Management Systems," University of London, 2010.

[90] OpenID, "The OpenID project." [Online]. Available: http://openid.net/. [Accessed: 11-Jan-2015].

[91] M. Laurent and S. Bouzefrane, *Digital Identity Management*. London, UK: ISTE Press Ltd, 2015.

[92] Higgins, "Higgins - Personal Data Service."

[93] Open Authorization, "OAuth 2.0." [Online]. Available: https://oauth.net.

[94] API Crazy, "Comparison of OpenID Connect with OAuth2.0 & SAML2.0," 2014. [Online]. Available: https://apicrazy.com/2014/07/23/comparison-of-openid-connect-with-oauth2-0-saml2-0/. [Accessed: 10-Sep-2017].

[95] PICOS, "Privacy and Identity Management for Community Services." [Online]. Available: http://www.picos-project.eu. [Accessed: 01-Sep-2017].

[96] S. G´ orniak, J. Elliott, M. Ford, D. Birch, R. Tirtea, and D. Ikonomou, "Managing Multiple Electronic Identities," 2011.

[97] STORK, "Secure idenTity acrOss boRders linKed." [Online]. Available: http://www.eid-stork.eu. [Accessed: 20-Jul-2009].

[98] V. Abreu *et al.*, "A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT," *Mob. Networks Appl.*, Oct. 2017.

[99] J. Bernal Bernabe, J. L. Hernandez-Ramos, and A. F. Skarmeta Gomez, "Holistic Privacy-Preserving Identity Management System for the Internet of Things," *Mob. Inf. Syst.*, vol. 2017, pp. 1–20, 2017.

[100] K. Grizzle, E. Wahlstroem, and C. Mortimore, "System for Cross-domain Identity Management: Core Schema," 2015.

[101] J. Kohl and C. Neuman, "RFC 1510: The Kerberos Network Authentication Service V5," 1993.

[102] J. Zouari and M. Hamdi, "AIDF: An identity as a Service Framework for the Cloud," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–5.

[103] J. Chen, Y. Liu, and Y. Chai, "An Identity Management Framework for Internet of Things," in *2015 IEEE 12th International Conference on e-Business Engineering*, 2015, pp. 360–364.

[104] A. Witkovski, A. Santin, V. Abreu, and J. Marynowski, "An IdM and Key-Based Authentication Method for Providing Single Sign-On in IoT," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[105] M. Cagnazzo, M. Hertlein, and N. Pohlmann, "An Usable Application for Authentication, Communication and Access Management in the Internet of Things," in *Communications in Computer and Information Science*, vol. 319, no. October, Springer International Publishing, 2016, pp. 722–731.

[106] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-Based Authentication Scheme for the Internet of Things," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016, vol. 2016–Augus, pp. 1109–1111.

[107] Y. Sharaf-Dabbagh and W. Saad, "On the Authentication of Devices in the Internet of Things," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, pp. 1–3.

[108] J. Liu, Y. Xiao, and C. L. P. Chen, "Authentication and Access Control in the Internet of Things," in *32nd International Conference on Distributed Computing Systems Workshops*, 2012, pp. 588–592.

[109] P. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things," *J. Cyber Secur. Mobil.*, vol. 1, no. 4, pp. 309–348, 2013.

[110] A. K. Ranjan and M. Hussain, "Terminal Authentication in M2M Communications in the Context of Internet of Things," *Procedia Comput. Sci.*, vol. 89, pp. 34–42, 2016.

[111] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A Conditional Privacy-Preserving Authentication With Access Linkability for Roaming Service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, Feb. 2014.

[112] K. A. Rafidha Rehiman and S. Veni, “A Secure Authentication Infrastructure for IoT Enabled Smart Mobile Devices – An Initial Prototype,” *Indian J. Sci. Technol.*, vol. 9, no. 9, pp. 520–523, Mar. 2016.

[113] A. Chaturvedi, D. Mishra, S. Jangirala, and S. Mukhopadhyay, “A Privacy Preserving Biometric-Based Three-Factor Remote User Authenticated Key Agreement Scheme,” *J. Inf. Secur. Appl.*, vol. 32, pp. 15–26, Feb. 2017.

[114] P. K. Dhillon and S. Kalra, “A Lightweight Biometrics Based Remote User Authentication Scheme for IoT Services,” *J. Inf. Secur. Appl.*, vol. 34, pp. 255–270, Jun. 2017.

[115] V. Odelu, A. K. Das, and A. Goswami, “A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards,” *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[116] S. Pötzsch, K. Borcea-Pfitzmann, M. Hansen, K. Liesebach, A. Pfitzmann, and S. Steinbrecher, “Requirements for Identity Management from the Perspective of Multilateral Interactions,” in *Digital Privacy, LNCS 6545*, J. Camenisch, R. Leenes, and D. Sommer, Eds. Springer-Verlag Berlin Heidelberg, 2011, pp. 609–626.

[117] T. Kölsch, J. Zibuschka, and K. Rannenberg, “Privacy and Identity Management Requirements: An Application Prototype Perspective,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6545, J. Camenisch, R. Leenes, and D. Sommer, Eds. Springer Berlin Heidelberg, 2011, pp. 735–749.

[118] Independent Centre for Privacy Protection (ICPP), “Identity Management Systems (IMS): Identification and Comparison Study,” 2003.

[119] A. A. Malik, H. Anwar, and M. A. Shibli, “Federated Identity Management (FIM): Challenges and opportunities,” *2015 Conf. Inf. Assur. Cyber Secur.*, no. 1, pp. 75–82, 2015.

[120] P. L. Gatti, *Probability Theory and Mathematical Statistics for Engineers*, 1st ed. Oxon, UK: Taylor & Francis Group, 2005.

[121] G. Lowe, “Breaking and fixing the Needham-Schroeder Public-Key Protocol using FDR,” in *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, 1996, pp. 147–166.

[122] S. Patel, D. R. Patel, and A. P. Navik, “Energy Efficient Integrated Authentication and

Access Control Mechanisms for Internet of Things," in *2016 International Conference on Internet of Things and Applications (IOTA)*, 2016, pp. 304–309.

[123] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "SDN controllers: A comparative study," *Proc. 18th Mediterr. Electrotech. Conf. Intell. Effic. Technol. Serv. Citizen, MELECON 2016*, no. 978, pp. 18–20, 2016.

[124] S. Tayeb, S. Latifi, and Y. Kim, "A Survey on IoT Communication and Computation Frameworks: An Industrial Perspective," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, 2017, pp. 1–6.

[125] A. Jara, L. Ladid, and A. Skarmeta, "The Internet of Everything through IPv6: An Analysis of Challenges, Solutions and Opportunities," *J. Wirel. Mob. Netwworks, Ubiquitous Comput. Dependable Appl.*, vol. 4, no. 3, pp. 97–118, 2013.

[126] S. Ziegler, A. Skarmeta, P. Kirstein, and L. Ladid, "Evaluation and Recommendations on IPv6 for the Internet of Things: Some insight from IoT6 European Research project," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 548–552.

[127] S. Deering and R. Hinden, "RFC 2460: Internet Protocol IPv6 Specification," 1998.

[128] S. Krishnan, J. Woodyatt, E. Kline, J. Hoagland, and M. Bhatia, "rfc6564: A Uniform Format for IPv6 Extension Headers," 2012.

[129] K. Werhle, G. Mesut, and G. James, *Modeling and Tools for Network Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[130] A Discrete-Event Network Simulator: NS3, "RIPng." [Online]. Available: https://www.nsnam.org/doxygen/group__ripng.html.

[131] J. C. Refsgaard and H. J. Henriksen, "Modelling Guidelines - Terminology and Guiding Principles," *Adv. Water Resour.*, vol. 27, no. 1, pp. 71–82, 2004.

[132] R. G. Sargent, "Verification and Validation of Simulation Models," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 4298–4309.

[133] L. Dong and K. Chen, *Cryptographic Protocol: Security Analysis Based on Trusted Freshness*. Beijing: Springer, 2012.

[134] M. Burrows, M. Abadi, and R. Needham, "A logic of Authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990.

[135] AVISPA, "AVISPA." [Online]. Available: http://www.avispa-project.org.

[136] C. J. F. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security

Protocols," in *CAV 2008*, 2008, pp. 414–418.

[137] B. Blanchet, "Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif," *Found. Trends® Priv. Secur.*, vol. 1, no. 1–2, pp. 1–135, 2016.

[138] Kamesh and N. Sakthi Priya, "Security Enhancement of Authenticated RFID Generation," *Int. J. Appl. Eng. Res.*, vol. 9, no. 22, pp. 5968–5974, 2014.

[139] S. A. Chaudhry, H. Naqvi, M. Sher, M. S. Farash, and M. U. Hassan, "An Improved and Provably Secure Privacy Preserving Authentication Protocol for SIP," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 1, pp. 1–15, Jan. 2017.

[140] F. Wu *et al.*, "An Efficient Authentication and Key Agreement Scheme for Multi-Gateway Wireless Sensor Networks in IoT Deployment," *J. Netw. Comput. Appl.*, vol. 89, no. November, pp. 72–85, Jul. 2017.

[141] H. Sundmaeker, P. Guillemin, and P. Friess, *Vision and Challenges for Realising the Internet of Things*. CERP-IoT – Cluster of European Research Projects on the Internet of Things, 2010.

[142] R. Abdmeziem and D. Tandjaoui, "Internet of Things: Concept, Building blocks, Applications and Challenges Challenges," *arXiv Prepr.*, 2014.

[143] P. Mahalle, S. Babar, N. R. Prasad, and R. Prasad, "Identity Management Framework towards Internet of Things (IoT): Roadmap and Key Challenges," in *Recent Treands in Network Security and Applications,* Springer Berlin Heidelberg, 2010, pp. 430–439.

[144] R. Roman, J. Zhou, and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," *Comput. Networks*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013.

[145] N. Carne, "Living in the Cloud, Gateway or On the Edge: IoT's Fragmented Future," 2015.

[146] D. Mahesh, "Smart Shelves for Retail : Redefine your In- Store Experience," *Happiest Mind white Pap.*, 2014.

[147] D. J. Wu, A. Taly, A. Shankar, and D. Boneh, "Privacy, Discovery, and Authentication for the Internet of Things," in *21st European Symposium on Research in Computer Security (ESORICS)*, Heraklion, Greece, 2016, pp. 301–319.

[148] British Gas, "Smart Meters." [Online]. Available: https://www.britishgas.co.uk/smart-home/smart-meters.html. [Accessed: 20-Aug-2017].

[149] D. G. Korzun, “Internet of Things Meets Mobile Health Systems in Smart Spaces : An Overview,” in *Internet of Things and Big Data Technologies for Next Generation Healthcare,* Springer International Publishing AG 2017, 2017, pp. 111–129.

[150] D. Minoli, *Building the Internet of Things with IPv6 and MIPv6*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013.

[151] S. Hagen, *IPv6 Essentials*, 2nd ed. USA: O’Reilly, 2006.

[152] M. Palattella *et al.*, “IoT - IPv6 Integration Handbook for SMEs,” 2014.

[153] NS3, “What is NS3.” [Online]. Available: https://www.nsnam.org/.

[154] G. Carneiro, P. Fortuna, and M. Ricardo, “FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3),” in *Proceedings of the 4th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2009, vol. 3.

[155] S. A. Chaudhry, K. Mahmood, H. Naqvi, and M. K. Khan, “An Improved and Secure Biometric Authentication Scheme for Telecare Medicine Information Systems Based on Elliptic Curve Cryptography,” *J. Med. Syst.*, vol. 39, no. 11, 2015.

Appendices

Appendix A: Smart Object Characteristics in the IoT

Smart objects, in the IoT [7], [141], [142], have the following main characteristics:

1. **Existence.** Things exist in the physical world. By using specific technologies, such as embedded communication devices, things are enabled to virtually exist in the digital world.
2. **The sense of self**. An identity is used by all things to describe themselves such as a car, Landover, or a plate number. Objects have the ability to process a data, make a decision, and react autonomously.
3. **Connectivity**. Objects can communicate other objects or entities nearby (or remotely) via the Internet.
4. **Interactivity**. Objects can interoperate with different types of entities, such as human, devices, real, or virtual, to offer or consume a wide range of services.
5. **Dynamicity.** Objects can interact with other objects without any limitation of time, place, or way. They dynamically can establish and terminate the network connection and can use a variety of interfaces.
6. **Environmental awareness.** Sensors might allow an object to be aware of its deployment contexts, such as water radiation or network overhead. This characteristic is optional because not all things will exhibit it, such as an object enhanced with a radio frequency identification (RFID) tag.
7. **Ownership awareness.** Objects aware of the identity of their owner(s) or user(s). They belong to a single user or a group of users such as a family, an organization, or enterprise
8. **Connectivity domain.** An intranet domain could be allowed to be established between objects owned by the same owner, where each one collaborates with others according to a predefined and fixed scheme

Appendix B: Security Challenges in the Internet of Things

The main challenges in the design and deployment of $IdMs$ are described in this section. Poorly designed IdM systems can increase existing security challenges and create an opportunity to disclose sensitive information of users.

1. Identity and Authentication

It is crucial to study how to manage identity and authentication in the IoT as multiple entities need to authenticate each other to establish trustworthy services [50], [143]. Due to the mobility property, there is a chance entities of the network do not know which partner can be used to create a certain service. For instance, in vehicular networks (VANETs) cars are expected to offer data not just for devices on the roadside but also to other cars. Managing identities of the vastly large community of things that are going to be interconnected become complex in an M:N scenario, where data providers are able to acquire and process information from other sources [144]. Consequently, some kind of authentication must be present in every service provider, including the tiniest of objects.

2. Access Control

Access control challenges in the IoT are closely related to those in any distributed system. For instance, a particular service is composed by aggregating several services and data from different locations and contexts such as a hospital retrieving information from a patient's home and ambulances. The providers of this information have their own access control and permissions whose lifecycles need to be managed [144]. The granularity (providing more information with the right credential) and location are important factors of the access control policies. Also, it is essential to consider the amount of computation resources available to a constrained device to implement a complex access control mechanism[109].

3. Protocol and Network Security

A successful authentication, in most cases, requires a secure communications channel to authenticate entities belonging to different domains. This process will make use of certain user credentials. In the IoT where any entity can connect with any other entity at any time, an extra challenge has arisen. These entities might not know each other in advance and limited devices can exchange information with other limited devices. Convergence and interoperability of technologies for identification and authentication that can work on a global scale are significant issues. This is because it includes the management of unique identities for physical objects and devices, and the handling of multiple identifiers for people and locations and possible cross-domains for the same entity and with associated authentication credentials. Devices that can be accessed directly in the IoT need careful consideration of the overheads caused by incoming connections (e.g. multiple incoming connections that require the use of public key cryptography).

4. Privacy

Data management and privacy benefit from the distributed IoT. The core idea is that every entity has control over the data it generates and processes. As a result, entities can control the granulator of the data they produce; entities can control and define their own access control policies, and entities do not need to provide all the data they produce, only the data requested by the external entity for a particular service (Cavoukian A., 2009 cited in [144]). However, the existence of entities that track users without their agreement will become very intrusive when misused.

5. Trust

In the IoT, trust is considered from two dimensions: trust in the interaction between entities, and trust in the system from the user perspective. There is uncertainty in both the interactions with the data providers and the interactions with the service providers. The distributed infrastructure makes the management of trust more complicated. This is because the data providers must be discovered and queried [30], [144]. Moreover, more relevant information (network status, an existing connection between entities) need more time to check. Thus, it can be possible to have an accurate view of the whole system.

Appendix C: The common IoT Architectures

The IoT could imply devices, cars, buildings, and consumer items, all of which are connected to the Internet. These things embed technology to sense or interact internally with its state or externally with the surrounding environment. To represent the IoT architectures components let us consider a fitness wearable device example such as a Fitbit. The Fitbit is used to monitor the body activities. Is the application embedded on the wearable device? The answer is that it works like a sensor, which gathers the data and sends it to an application for analysis purposes. Indeed, part of the application is embedded in the wearable device while the other parts of the application are hosted in the smartphone application. The user can share his/her fitness information with others by putting some of the application on the cloud service. It is clear that the application's location of such devices is distributed between the wearable itself, a smartphone application and the cloud. Therefore, the fitness data is stored in different places.

The network edge represents the things, which could be any device, house, car, user equipment or any connected things. These "things" are usually connected using a suitable gateway or smartphone. The information is then passed on to the other parties directly or through the cloud. This is a high-level architecture, which may contain multiple intermediary communication devices in the real applications.

The four main computing components, as shown in the following figure that support IoT to deliver value are:

1. ***User Interface:*** Its role is a presentation in multiform. It could range from smartphone applications or web pages to a small LED attached to the things. It could be embedded in multi places concurrently.
2. ***Application logic and rules:*** This component gives the "thing" its functionality and smartness. It could exist on the thing, or in gateways, smartphones, cloud or local control system.
3. ***Data:*** It represents the data shared by the thing with interested parties.
4. ***Analytics:*** This component role analyses the thing data. It could be any algorithm used by different systems.

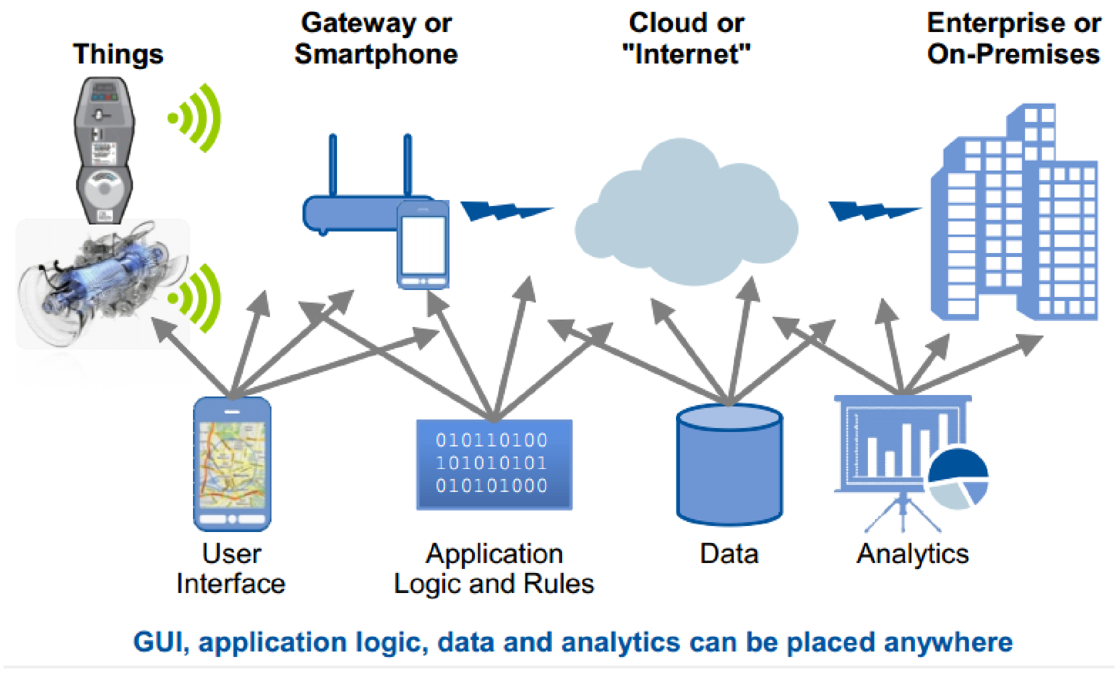

Things
Gateway or Smartphone
Cloud or "Internet"
Enterprise or On-Premises
010110100
101010101
010101000
User Interface
Application Logic and Rules
Data
Analytics
GUI, application logic, data and analytics can be placed anywhere

Figure C-1: High-level IoT Architecture [11]

According to Gartner, the world's leading IT research and advisory company, these components are represented the following main IoT architectures. The enterprises can use anyone of these architectures to implement its IoT environment. Moreover, the combination of these base architectures is also possible [41], [145]. Although each architecture has pros and cons, they are used to build the IoT ecosystems and there is no clue that one of them is likely to dominate.

A. Thing-Centric Architecture

In this type, the thing has the most of computing resources to do its operations. It stores the majority of its data on-board and it has a smartness to implement the algorithms and present the results on its UGI. The thing could also communicate with the Internet to share the information and/or receive control instructions remotely. This architecture could be found in asset-intensive fields such as construction, agriculture, infrastructure utilities and transportation, where the assets have enough computing resources to stand alone and communicate remotely.

B. Gateway/Smartphone-Centric Architecture

In this type, the majority of computing resources are hosted by the gateway/smartphone. The things only sense the data, transfer the data through its communication portal, and implement received commands. They are essentially dumb devices as the gateway/smartphone offers the required smartness. For instance, in the street lights, the smartness required to switch the lights on/off could reside on the gateway instead of on the lights. Similarly, the data of a wearable health monitoring device could be gathered and processed by a smartphone. The smartphone, in this case, serves as the Internet gateway. This architecture is widely used in many IoT applications area and industries such as building systems, utilities, smart city/home, traffic management, and mobile health-monitoring systems.

C. Cloud-Centric Architecture

The thing in this architecture relies on the cloud to use its resources to host the application, store data, and data analysis. The things role in this type is similar to gateway architecture where they have as little computing and storage capability as possible. However, this architecture requires a persistent Internet connection with a cloud. For instance, a home thermostat might have only sensors onboard to sense temperature while the application that controls the house temperature is hosted by the cloud. The house temperature records are stored by the cloud as well.

D. Enterprise-Centric Architecture

This type represents the “Intranet of Things”, where things are kept inside the enterprise border. The applications, data storage, and the analysis are done inside the enterprise as well, for instance, a factory that has multi-connected things. These things are located in the same area where there is no need for a remote connection using the public Internet. They use LAN or WAN topologies for communication.

Appendix D: Scenarios Analysis for Requirements Elicitation

A scenario driven approach has been used to elicit the design requirements for the identity management system for the IoT environment. Each of the following typical IoT scenarios is presented using:

- Brief introduction.
- Description.
- Motivation.
- Requirements

Scenario 1. Requesting Services within an IdM Domain

The classical model for requesting services is within an IdM domain. In this model, the interacted actors' identifiers are managed by the same IdP. The services are offered to the requester within the IdM domain border. In other words, the SP is managed by an isolated or a centralized IdM model. From a networking perspective, the communication between the requester and the SP is either locally through a local area network or remotely through a virtual private network (VPN). Such interaction could be seen in an enterprise assets tracking in distributed branches, disaster management systems, traffic management system in smart cities, and more. Table D-1 presents the scenario analysis.

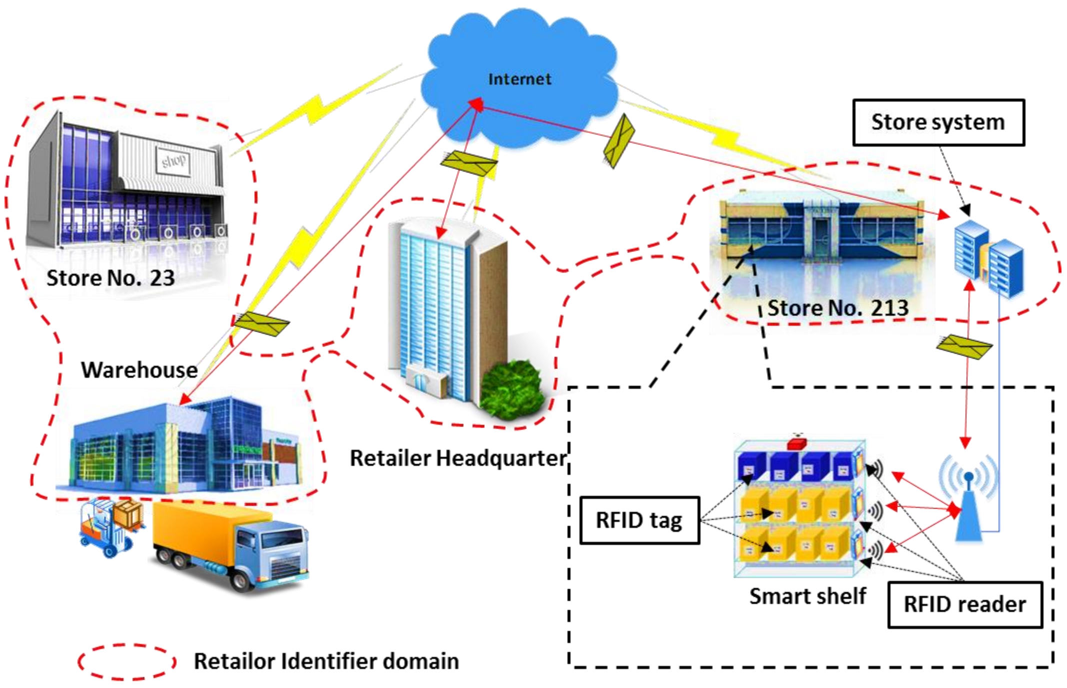


Figure D-1: A Smart Shelf Scenario

Table D-1: An Explanation of a smart shelf scenario.

Description
In a smart retailer use case, the smart shelves technology can improve the retailer business. The smart shelves continually monitor the stock levels and predict products demand. They embed sensors, usually using RFID readers, which interact with the products tags. In the case where a product's quantity is lower than the stock level, the RFID reader detects the situation. Next, it alerts the store management system and sends a replenishment request to the warehouse to avoid an "out of stock" situation. It is unnecessary for the warehouse to be on the same local area network. Furthermore, the RFID reader can inform the headquarters about the current buyer demands for a future sales plan [146]. From the headquarter viewpoint, this technology facilitates remotely monitoring the store's stocks, and fixing problems before they become severe in nearly real-time. Thus, to ensure its continued operation, they have to maintain that the hardware and applications interaction, products monitoring, remote alerting, and automatic dispatching capabilities are continued. Figure A-1 illustrates the scenario.
IdM Motivation
When the RFID reader detects a low level of a product A, its identifier will be used when interacting locally with the store management system, or externally with the warehouse in order to transmit the product A identification code. The RFID reader has at least two identifiers: a unique identifier as a retailor asset and the IP address to share data via the Internet. From the IoT viewpoint, the RFID reader could interoperate with the warehouse database to check the availability of the product, send updated data, and request the stock replenishment. RFID reader roles in this scenario are to sense, process and transmit traffic on behalf of the EA, which is the store No. 213. All identities of the tags, readers, branches, warehouses, applications, and devices are managed within a single IdM system. However, the retailer warehouse is still required to establish the identity of the requester even when the request is being done over a VPN.
Requirements
Establishing the EA identity in such scenarios needs an IdM supports the following requirements. 1. The SP should be able to differentiate between the EA identifier and the communication device identifier. This requires representing them in a semantic format.

2. The SP should establish the EA identity before provisioning the request. In other words, the store No. 213 has to be identified by the warehouse before accepting its request / order. To do so, it is important for the SP to recognise the following:
   - ➢ How the EA interacts with the communication object to transmit the data/request? In other words, the SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the RFID reader serves store No. 213 based on the permanent relationship.
   - ➢ What is the EA type (i.e. person, legal entity, device, or application) to map each actor to its permitted role in the domain? The EA is a legal entity which cannot request without an association with a communication device.
   - ➢ What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor to its permitted role in the domain? The RFID reader can accesses the Internet, i.e. active. Thus, it can make requests on behalf of the store.
3. The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship. For instance, the EA, i.e. the store No. 213, and the RFID reader should delegate their identifiers to form a relationship between them.
4. The communication object should be aware of its relationship with the EA, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP.
5. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.
6. The SP should have an appropriate authentication protocol to establish the EA identity based on its relationship with the communication device(s) and the actor's characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Scenario 2. Requesting Services within a Circle of Trust (CoT)

Services could be requested from another domain based on an agreement between them. In the CoT, identifying the entity is based on the single sign on (SSO) agreement between the $IdPs$. SSO means that the IdP in a visited domain will accept the identifiers issued by other members in the CoT. To demonstrate this interaction, let us consider a scenario of collecting data from a smart home within a CoT. Table D-2 presents the scenario analysis.

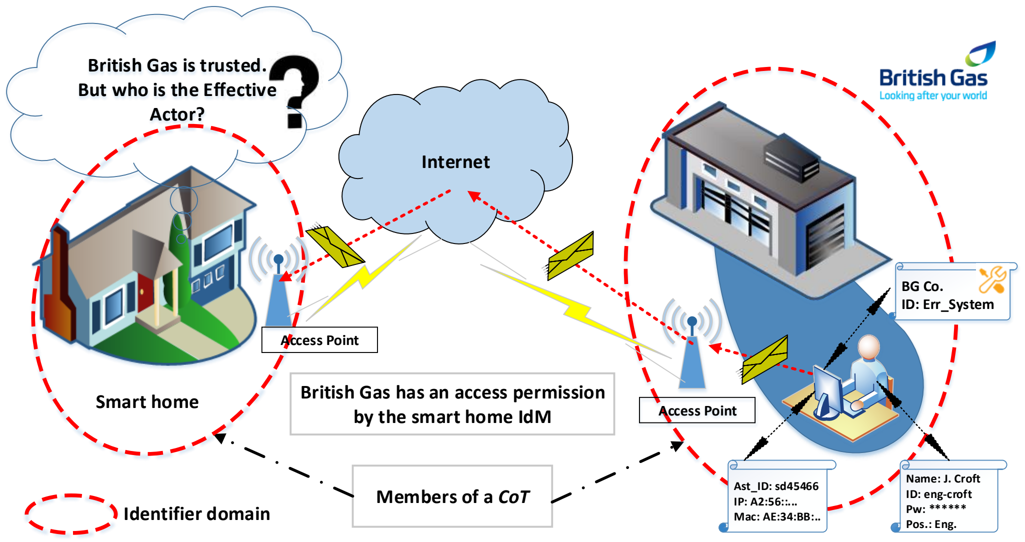


Figure D-2: A Remote Data Collection within Related Domains

Table D-2: An Explanation of a remote data collection within related domains

Description
A smart home implies the IoT-enabled devices such as an alarm system, heating control system, security cameras, health-monitoring devices, smart locks, smart meter devices, and more. Some devices deal with or are controlled by multiple people such as family members, friends, employees, and service engineers [147]. In the scenario, Mike is the homeowner that is managed by an IdM. The IdM manages and controls accessing these devices, and their generated data, to/from the Internet. British Gas is one of the biggest energy and homecare companies in the UK. It has an ambitious plan to support all its customers with smart meters by 2020 [148]. These devices could be accessed remotely via the Internet to collect nearly real-time readers. Figure D-2 illustrates the scenario. Assume that the company has a homecare system to remotely monitor, check, and collect related data from some smart meters and appliances. The system has to interact periodically with these smart meters, a smart boiler, carbon monoxide detectors, and some other home appliances. The collected data might be used to update records, auto-discover errors or problems in their early stages and report them to interested parties like engineers. In this case, there is a collaboration between the home and British Gas with respect to $IdMs$, which allows the company's system to access the required data from participant devices. When the homecare system detects an abnormal situation, it reports the case to one of the company engineers for further investigation.

IdM Motivation

On the one hand, the company has its own IdM system to manage its staff and assets identities. The engineer could uses his identification details (e.g. user name and password) within the company's (or within a pre-established CoT) IdM domain, to login to the system, which is installed on the company computer. The computer has several identifications within the company such as a unique asset identifier and the network IP address to use to interoperate with the other devices (send or receive data or service). Similarly, the homecare system has its own identity, as one of the company's applications, to be used when interacting with other systems. On the other hand, the smart home has its own IdM system to manage the identities of the home members, devices and appliances. Furthermore, it participates with the energy company in a customer CoT, which allows sharing data between the members. Similarly, the home IdM system has to identify and authenticate the actual requester's identity, i.e. the EA, before it acknowledges any request within this CoT.

In this scenario, the EA of the communication is the homecare system. It is unable to interact with the smart meters without using suitable devices like the computer to send a request to gather some data from to the smart home meters or other appliance. Although, there is a collaboration between these $IdMs$, i.e. in the home and the company, the SP in the home has to assure the requester identity rather than the requester's device identity. It is essential for the home owner and the company to establish the identity of the actual actor behind the communicated device or object. Otherwise, this data could be accessed by an unauthentic party which could cause serious damage or endanger the home owner. Furthermore, it could prevent the home owner from getting better services that the requester intends to supply.

From a networking viewpoint, the computer uses its IP address to send the access request which represents the machine location in the Internet network rather than the identity of the system or the requester. However, there is no guarantee that all entities in the IoT will have a fixed IP address. Also, the IoT environment contains many entities that cannot directly connect to the Internet through the Internet protocol. Thus, the IP will not help with identifying the requester in the IoT. In other words, there is a need to decouple the locator address and the identifier that represents the requester in IoT.

Requirements

To establish the EA identity in such scenarios needs an IdM supports the following requirements.

1. The SP should be able to differentiate between the EA identifier and the communication object/device identifier. This requires representing them in a semantic format. In other words, a semantic format is required to represent these actors to the SP.
2. The SP should establish the EA identity before provisioning the request. In other words, the homecare has to be identified by the smart home IdP before his request be accepted. Generally, it is important for the SP to recognise the following:
   - How the EA interacts with the communication object to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the computer communicates with the smart meter on behalf of the homecare system based on its permanent relationship with the system.
   - What is the EA type (i.e. person, legal entity, device, or application) to map each actor to its permitted role in the domain? The EA is of application type which cannot request without an association with a communication device.
   - What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor to its permitted role in the domain? The computer can access the Internet, i.e. active. Thus, it can requests data on behalf of the homecare.
3. The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship. In other words, the EA, i.e. the homecare system, and the computer should delegate their identifiers to form a relationship between them.
4. The communication object should be aware of its relationship with the EA, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP.
5. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.
6. The SP should have an appropriate authentication protocol to establish the EA identity based on its relationship with the communication device(s) and the actor’s characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Scenario 3. Requesting Services across Unrelated Domains or a CoT

Such collaboration is quite normal in an IoT environment because of the IoT openness. Entities, in IoT, could be interconnected with others that might belong to independent domains and use different identification data based on the anywhere basis. Moreover, due to the nomadic nature of some of the IoT entities across its IdM domain, the environment should support them to access the services across their IdM domain. This section analyses two cases that are (1) requesting services or data across different $CoTs$; (2) requesting services or data between two independent domains. These cases are analysed in the following two scenarios. However, such cases suffer from a serious interoperability challenge between the IdM systems across-domain.

Scenario 3.1. Requesting Services across Different $CoTs$

This scenario discusses remotely gathering data from a smart home across different $CoTs$. Table D-3 presents the scenario analysis.

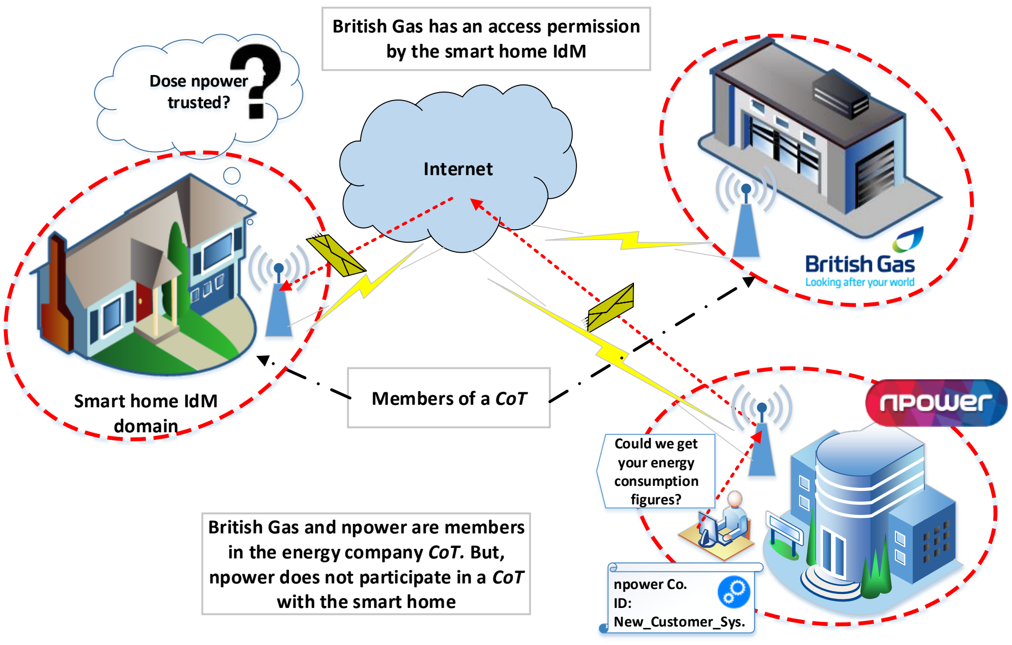

British Gas has an access permission by the smart home IdM
Dose npower trusted?
Internet
British Gas
Looking after your world
Smart home IdM domain
Members of a CoT
npower
Could we get your energy consumption figures?
British Gas and npower are members in the energy company CoT. But, npower does not participate in a CoT with the smart home
npower Co.
ID:
New_Customer_Sys.

Figure D-3: Remotely Gathering Data from Unrelated $CoTs$ Scenario

Table D-3:An Explanation of a remotely gathering data from unrelated $CoTs$ scenatio.

Description
Continuing with the smart home scenario to demonstrate the case of requesting a service across two different $CoTs$, the energy supplier Npower intends to attract new customers by

offering their services with a reasonable price. In order to offer a realistic and competitive quotation to the new customers, Npower requires to know the customer's actual energy consumption. Thus, its system interconnects with the smart meters in the smart home to collect a customer energy consumption behaviour through a period. Npower uses its own IdM to manage its staff and assets. Furthermore, Npower and British Gas are participating in an energy companies CoT. Figure D-3 illustrates the scenario.

IdM Motivation

The smart meter (or home IdP) receives a request to collect the energy consumption behaviour singed by the npower system identifier. Although, the smart home participates in the client CoT with the British Gas and the energy companies CoT implies the npower and British Gas, the request is directed form the npower system directly to the smart meter. Moreover, the home's IdM permits only the British Gas assets and staff accessing its meters. That means, there is no way to extend the clients CoT to imply new member, i.e. the npower. This is because there is no prior trust relationship between the smart home domain and the npower. Therefore, establishing trust relationship between unrelated domains is a pre-request to the identity establishment across-domain in IoT environment.

The smart meter (or home IdP) receives a request to collect the energy consumption behaviour from the Npower using the identifier of Npower system. The smart home participates in the client CoT with British Gas. Meanwhile the energy companies participate in another CoT, which implies both Npower and British Gas. However, these two $CoTs$ are not linked together, hence the services across these $CoTs$ are not permitted. Moreover, the home's IdM permits only the British Gas assets and staff accessing its meters. That means, there is no way to extend the clients CoT to imply new members, i.e. the Npower. This is because there is no prior trust relationship between the smart home domain and the Npower. Therefore, establishing a trust relationship between unrelated domains is a pre-request to the identity establishment across-domain in the IoT environment.

Requirements

To establish the EA identity in such scenarios needs an IdM, which supports the following requirement.

1. The SP should be able to differentiate between the EA identifier and the communication object/device identifier. This requires representing them in a semantic format. In other words, a semantic format is required to represent these actors to the SP.

2. The actor identifier should be paired with its home registration domain IdP identifier. Thus, the SP will be aware of the domain that manages the identifier involved in the EA identity establishment process. In other words, the smart home IdP needs to trust the domain that manages the requester system identity before acknowledge its request.
3. The SP should establish the EA identity before provisioning the request. In other words, the Npower system has to be identified by the smart home before accepting the request. Generally, it is important for the SP to recognise the following:
   - ➢ How the EA interacts with the communication object to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the Npower system communicates the smart meter based on its permanent relationship with company's computer.
   - ➢ What is the EA type (i.e. person, legal entity, device, or application) to map each actor to its permitted role in the domain? The EA is of application type which cannot request without an association with a communication device.
   - ➢ What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor to its permitted role in the domain? The computer can access the Internet directly, i.e. active. Thus, it can makes the request on behalf of the Npower system.
4. The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship. In other words, the EA, i.e. npower system, and the computer should delegate their identifiers to form a relationship between them.
5. The communication device/object should be aware of its relationship with the EA actor, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP. In other words, computer needs to be aware of its relationship with the npower system and be able to represent it to the smart home IdP.
6. The SP should be able to dynamically establish a trust relationship with unrelated domains IdP in order to involve it in the identity establishment. In other words, the smart home IdP needs the ability to establish a trust relationship with the Npower before starting to establish its system identity.
7. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.

8. The SP should has an appropriate authentication protocol to establish the EA identity based its relationship with the communication device and the actor's characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Scenario 3.2. Requesting Services across Different Domains

This scenario explains requesting service across different domains in the Eduroam scenario. Table D-4 presents the scenario analysis.

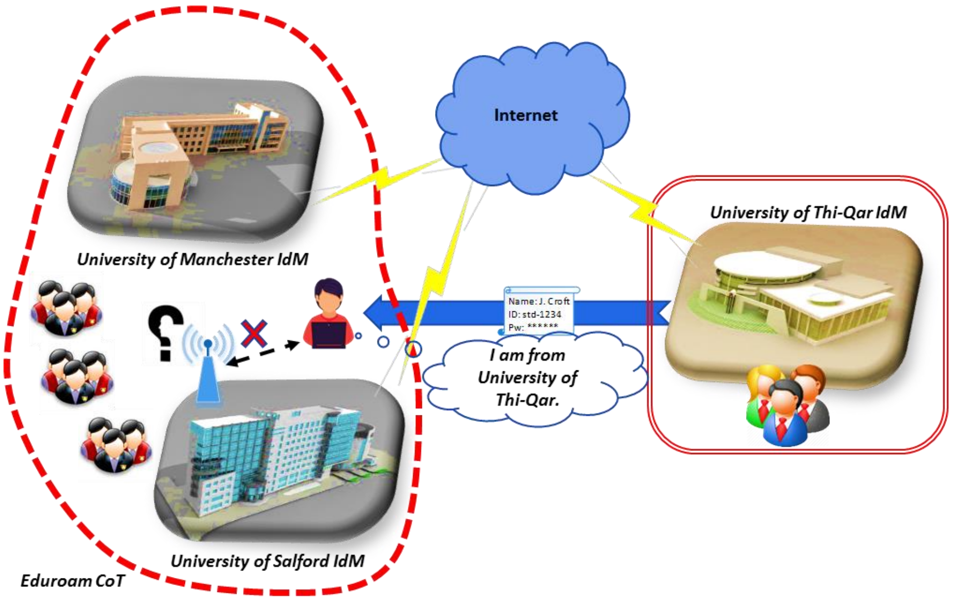

Internet
University of Manchester IdM
University of Thi-Qar IdM
Name: J. Croft
ID: std-1234
Pw: ******
I am from University of Thi-Qar.
University of Salford IdM
Eduroam CoT

Figure D-4: Eduroam Scenario

Table D-4: An Explanation of an Eduroam scenario

Description
The Eduroam is the collaboration of some of the university's campuses participating in a CoT to offer their services across these campuses to their staff and students. Both the University of Manchester ($UoMan$) and University of Salford (UoS) participate in the "eduroam " group (Eduroam is a free, worldwide wireless Internet service provided by JISC.). In the eduroam, any member from the participating universities can access the other universities services by using his/her home university/organisation identification. Imagine a staff member from an external educational institution of the eduroam CoT, e.g. University

of Thi-Qar ($UoTQ$), attending a workshop at UoS. The visitor requests a service from UoS using his/her identification from $UoTQ$ and one of the UoS laptops. Consequently, the SP, in UoS, receives the requester identification and directs it to the IdP to authenticate the requester. Although, the requester identification contains his/her identifier and the IdP identifier of $UoTQ,$ the request will be restricted because $UoTQ$ is not member in this eduroam CoT. However, this action opposes the IoT aimto allow nomadic entities to get a right service at any domain. Overcoming such a restriction is essential to the success of the IoT smart environment. Figure D-4 illustrates the scenario.

IdM Motivation

From the IdM viewpoint, before acknowledging the request, UoS has to do two things: first, be assured that $UoTQ$ is trusted within the eduroam participants; next, establish the visitor identity. To get the first step, SP in UoS should have the ability to dynamically trust the external domain and establish a trust relationship with him to perform the actor identity establishment. Thus, the requester identifier should be paired with its home IdP identifier to facilitate its recognition in IoT environment. After that, the SP in UoS should have the ability to establish the visitor staff identity using his/her educational institution identification.

Requirements

To establish the EA identity in such scenarios needs an IdM which supports the following requirements.

1. The SP should be able to differentiate between the EA identifier and the communication object/device identifier. This requires representing them in a semantic format. In other words, a semantic format is required to represent these actors to the SP.
2. The actor identifier should be paired with its home registration domain IdP identifier. Thus, the SP will be aware of the domain that manages the identifier to involve in the EA identity establishment process. In other words, the IdP in UoS needs to know which domain manages the requester system identity before acknowledge the request.
3. The SP should establish the EA identity before provisioning the request. In other words, the identification of the EA is required by the UoS before accepting the request. Generally, it is important for the SP to recognise the following:
   - How the EA interacts with the communication object to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the relationship

type between the visitor and the $UoS's$ laptop is of open access relationship type because it is used by anybody.

- ➢ What is the EA type (i.e. person, legal entity, device, or application) to map each actor to its permitted role in the domain? The EA is of person type which cannot request without an association with a communication device.
- ➢ What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor to its permitted role in the domain? The laptop device can access the internet, i.e. active. Thus, it can request on behalf of the visitor.

4. The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship. In other words, the EA, i.e. visitor staff, and the laptop should delegate their identifiers to form a relationship between them.
5. The communication device(s)/object(s) should be aware of its relationship with the EA actor, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP. In other words, laptop device needs to be aware of its relationship with the visitor and be able to represent it to the UoS.
6. The SP should be able to dynamically establish a trust relationship with unrelated domains IdP in order to involve it in the identity establishment. In other words, SP in UoS needs the ability to establish a trust relationship with the IdP of $UoTQ$ before starting to establish the requester identity.
7. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.
8. The SP in UoS should has an appropriate authentication protocol to establish the EA identity based on relationship with the communication device(s) and the actor's characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Scenario 4. Requesting Services using a permanent-interaction between the EA and the communication objects.

The classic way of interaction between EAs and the communication object(s) is through a permanent relationship to access services on the Internet. IoT implies many use cases of this type in different domains, for instance, in a smart home when a homeowner uses his/her owned smart devices to get services or data; in an industrial domain when a factory central system tracks its machines status on distributed branches using its owned tracking devices; and so on. A typical scenario has already been discussed in the smart shelf scenario. In such cases, the EA identity is tightly linked with these objects identity. Therefore, the SPs offer their services based on their previous knowledge that the object is interconnected on behalf of the EA.

From an IdM poitn of view, all actors have an identity to be used within the domain. This means that both the EA and the communication object, in a relationship, have their own identity. In such scenarios, the communication object identity is used as an alternative or a secondary identity for its user based on the fixed relationship type with the user. Therefore, the SP can easily identify the EA to offer him right services.

Scenario 5. Requesting Services using a Semi-Permanent Interaction Between EA and the Communication Object

The interaction between actors is not always fixed but could be changeable. This situation is typically seen in many IoT application areas where a group of devices and objects are permitted to serve or use by multiple EAs. The interaction lifetime is varied and could be held for a long or limited period. The long period cases could be found, for instance, where university staff and students use cases who are authorised to use the university PCs to participate in online conferences. A short lifetime relationship could be seen in the cases where another actor is permitted to use an object for a predefined period. The scenario of paying a road toll charge automatically illustrates this type of relationship as follows. The idea, of the scenario, is to use some special devices that can link with the bank account or other payment methods of the actor to pay the charge automatically on their behalf. These devices usually use short-range communication technologies such as RFID, BlueTooth (BT), Near Field Communication (NFC) or ZigBee. Table D-5 presents the scenario analysis.

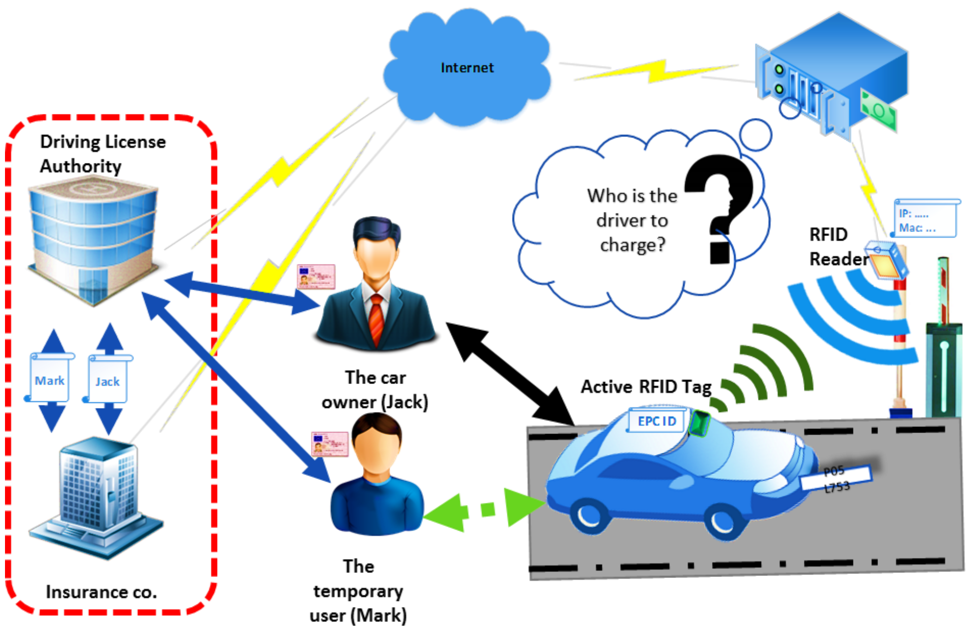


Figure D-5: A Road Toll Scenario

Table D-5: An Explanation of a road toll scenario.

Description
Mahalle et al. in [43] mentioned the case where the legal car owner (Jack) permits his friend (Mark) to use his car for a temporary period by adding him as a second driver for a limited time. Jack and Mark have their insurance and road toll accounts with related companies. The car has an active RFID tag with a control system. The RFID tag role is to transmit the driver data to the reader. The driver's data is gathered by the control system from his driving licence when he attaches it to a card reader in the car. The driving licence data is linked to his road toll account to pay any charges from the driver account. Mark drives the car to another city. As soon as the car reaches the road toll gate, the active RFID reader magnetic field detects the car RFID tag and gathers the identification data from the car's tag. The tag identification, i.e. the "electronic product code (EPC)" represents the car's identification data and its legal owner, i.e. Jack. Its unable to show the current driver, i.e. Mark, when he drives Jack's car. In such cases, the IoT is expected to help the SPs, which is the road toll company, to identify the correct account to charge. Therefore, the communication objects in IoT should maintain their relationship with the actual actor and should be able to identify him to the SP to get the right service. Figure D-5 illustrates the scenario.

IdM Motivation

The relationship between the user as EA and the communication device and its impact on offering the services is explicitly explained in the road toll scenario. The EPC, as RFID tag identifier, is used to link the driver's account with the road management company to collect the road charge from the driver's account. However, when the car driver is changed, i.e. the driver's relationship with the car is changed, the EPC should refer to Mike's account instead of Jack's. Otherwise, the IoT application's trust and accuracy will be degraded. For this, consider the relationship types have a direct impact on establishing the EA identity because that relationship(s) itself is not static as it may change or even vanish. Thus, the IoT environment requires an IdM with a new identification method that identifies the effective actor based on his relationship with the communicated object. This could facilitate the entities mobility characteristic in the IoT environment. Having such an IdM in the IoT with heterogeneous entities with the ability to effectively identify the actual entity is a critical task.

Requirements

To establish the EA identity in such scenarios needs an IdM, which supports the following requirements.

1. The SP should be able to differentiate between of the EA identifier and the communication object/device identifier. This requires representing them in a semantic format. In other words, a semantic format is required to represent these actors to the SP.
2. The actor identifier should be paired with its home registration domain IdP identifier. Thus, the SP will be aware of the domain that manages the identifier involved in the EA identity establishment process. In other words, the road management system as a SP needs to know which domain manages Mark identity to debt the charge.
3. The SP should establish the EA identity before provisioning the request. In other words, the identity of Mark has to be established by the toll company before collecting the charge. Generally, it is important for the SP to recognise the following:
   - How the EA interacts with the communication object to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the relationship type between Mark and the car is of semi-permanent relation because he is already added by Jack as a second driver in the car registration authority.

- What is the EA type (i.e. person, legal entity, device, or application) to map each actor to its permitted role in the domain? The EA is of person type which cannot request without an association with a communication device.
- What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor to its permitted role in the domain? The car's active RFID tag cannot access the Internet directly, i.e. passive. Thus, another relationship is required to transmit the data to the company server.
- What is the type of the transitive relationship if existing? The SP should recognise the transitive relationship type between the EA and the communication object that transmitted the data/ request. For instance, the relationship type between Mark and the RFID reader is of open access relationship type, hence, no transitive relationship could be represented.

4. The interacting actors, i.e. the EA and the communication object, should delegate their identities to form an actor relationship. In other words, the EA, i.e. Mark, and the car should delegate their identifiers to form a relationship between them.
5. The communication device/object should be aware of its relationship with the EA actor, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP. In other words, the car control system needs to be aware of its relationship with the current driver and be able to represent him to the road toll company.
6. The SP should be able to dynamically establish a trust relationship with unrelated domains IdP in order to involve it in the identity establishment. In other words, road management system as a SP needs the ability to establish a trust relationship, if not existing, with the external IdP, that manages Mark's identity, before starting to establish his identity.
7. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.
8. The road toll company should have an appropriate authentication protocol to establish the EA identity based on relationship with the communication device(s) and the actor's characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Scenario 6. Requesting Services using an Open Access Interaction Between EA and the Communication Object

In the IoT vision, things can interact with others to get services or data regardless of their communication technology. To cope with this vision, the current interaction bases between IoT actors should consider the open access interaction in which, the actors could dynamically establish a relationship to perform a required task and terminate it when the task is complete. In another words, the EA could interact with the communication object and access the Internet without requiring a pre-registration, as seen in the other types of interaction. In such cases, the object identity could not be considered to identify the EA identity because it represents a gateway/broker for the EA to access the Internet. Thus, the SP should be able to determine who is the EA and which domain could participate in the identification and authentication process and some domain specific information. A demonstration of such a relationship could be seen in the interaction of a health monitoring scenario. Table D-6 presents the scenario analysis.

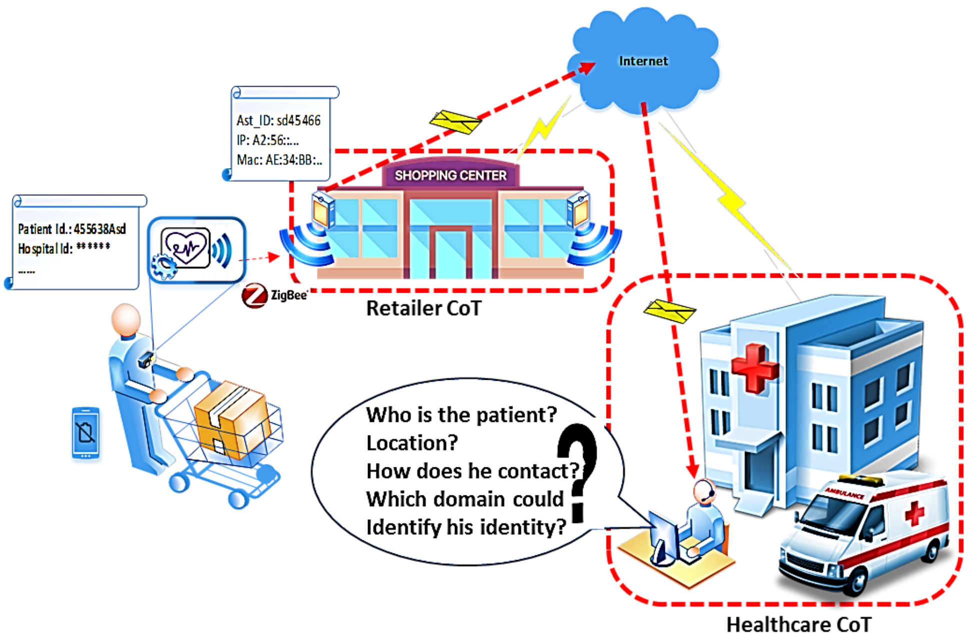


Figure D-6: A Health Monitor Scenario through an open-access Relationship

Table D-6: An Explanation of a Health Monitoring scenario.

Description
Patients health monitoring is one of the IoT applications that serve people with chronic diseases. In such cases, medical sensors like ECG, blood pressure, pacemaker, and more are wearable on or implantable in the patient's body to monitor it's activities [149]. They usually use short-range communication technologies such as BT, NFC, or ZigBee to transmit data. To pass the data to their doctor or hospital system, another device with Internet access is required like IoT gateways; e.g. smartphones, tablets, or any other active device. Considering the case where the patient is walking in a shopping centre, these sensors would detect abnormal activity, which would require an emergency service. Figure D-6 illustrates that the monitor device sends its messages through a ZigBee[2] technology to the smartphone; but, unfortunately, the patient's smartphone has run out of battery. Instead, a nearby free service available in the shopping centre which could be used to access the Internet. Thus, the monitor device connects to that service and transmits its emergency messages to the hospital. Two different types of relationships have been established in this case. The first one is between the patient and the monitoring device which has detected the changes in his situation. The second relation is between the monitoring device and the centre's access point which works as a broker between that device and the Internet.
IdM Motivation
Here, we have two different cases to transmit data. The first one is based on a permanent relationship between the patient and his/her smartphone, while the second is based on a free relationship between the patient and the access point at the shopping centre. It is important to consider such cases when designing an IdM for the IoT. Although the hospital's healthcare system receives the message from the shopping centre's access point, it needs to identify who sent it, i.e. the patient situation and location so that the right staff could be sent. However, the access point address in the centre cannot be used to identify the patient because there is no relationship between the patient's sensors and the shopping centre's access point. It could represents the patient location rather than his/her identity because of the nomadic nature of entities and objects in the IoT environment.

[2] "ZigBee is the IEEE 802.15.4 communication protocol used to create personal area networks".

Identifying entities in such an environment become a challenge to face when developing an IdM system. Thus, in order to decide which domain could be involved in establishing the EA identity, the SP requires to know the EA relationship type(s) with the communication object that interconnects on his/her behalf..

Requirements

To establish the EA identity in such scenarios needs an IdM, which supports the following requirements.

1. The SP should be able to differentiate between the EA identifier and the communication object/device identifier. This requires representing them in a semantic format. In other words, a semantic format is required to represent these actors to the SP.
2. The actor identifier should be paired with its home registration domain IdP identifier. Thus, the SP (or the visited domain IdP) will be aware of the domain that manages the identifier involved in the EA identity establishment process. In other words, the healthcare system as a SP needs to know which domain manages patient identity when processing the request.
3. The SP should establish the EA identity before provisioning the request. In other words, patient's identity has to be established by the hospital's healthcare system to send the right staff. Generally, it is important for the SP to recognise the following:
   - How the EA interacts with the communication object(s) to transmit the data/request? The SP should recognise the relationship type between the EA and the communication object that transmitted the data/ request. For instance, the relationship type between the patient and health monitor device is a semi-permanent relationship, while the relationship type between the health monitor device and the patient smartphone is a permanent relationship. The relationship type between the health monitor device and the centre's access point is free as well.
   - What is the *EA* type (i.e. person, legal entity, device, or application) to map each actor with its permitted role with the domain? The *EA* is of person type which cannot request without an association with a communication device.
   - What is the Internet connectivity type (i.e. passive or active) of the communication device to map each actor with its permitted role with the domain? The health monitor device cannot access the Internet directly, i.e. passive. Thus, another relationship is required to transmit the data to the company server.

- What is the type of the transitive relationship if existing? The SP should recognise the transitive relationship type between the EA and the communication object that transmitted the data/ request. For instance, the relationship type between the patient and his/her smartphone device is of a permanent relationship, which represents a transitive relationship.

4. The interacting actors, i.e. the EA and the communication object(s), should delegate their identities to form an actor relationship. In other words, the EA, i.e. patient, the smartphone, and the centre's access point should delegate their identifiers to form relationship(s) between them.
5. The communication device(s)/object(s) should be aware of its relationship with the EA actor, which communicates on its behalf. This relationship should be registered within the actor domain IdP. It should also be identifiable, recognizable and provable by the SP. In other words, the health monitor device, the smartphone, and the centre's access point needs to be aware of its relationship with the patient and be able to represent them to the hospital's healthcare system.
6. The SP should be able to dynamically establish a trust relationship with unrelated domains IdP in order to involve it in the identity establishment. In other words, SP needs the ability to establish a trust relationship, if not existing, with the external IdP, that manages the patient's identity, before starting to establish his/her identity.
7. The SP should establish the EA identity based on its identifier and the actor relationship instead of the IP address. This is because the IP address refers to the communication object location on the network topology rather than its end user.
8. The hospital as a SP should have an appropriate authentication protocol to establish the EA identity based on its relationship(s) with the communication device(s) and the actor's characteristics, i.e. identifiers, types, Internet connectivity, and the responsible IdP identifiers.

Appendix E: IPv6 benefits for the IoT

There are several characteristics that show IPv6 will be a key enabler for the IoT [150]–[152]:

1. ***Scalability**:* IPv6 offers a highly scalable address scheme. It offers more than 2 billions addresses per square millimetre that results from 2^{128} unique addresses of the Earth surface, which represents 3.4×10^{38} addresses. It is quite sufficient to address the needs of any present and future communicating device.
2. ***Solving the NAT barrier**:* Because of the limitation of the IPv4 address space, the Network Address Translation (NAT) allows several users to share the same public IP address. This solution is working but has two main drawbacks. (1) The users do not have their own public IP address, which turns them into homeless Internet users. They can access the Internet, but they cannot be directly accessed from the Internet. (2) The end-to-end connection will be broken and dramatically weakens any authentication process.
3. ***Mobility:*** IPv6 provides strong features and solutions to support mobility of end-nodes, as well as mobility of the routing nodes of the network.
4. ***Address self-configuration**:* IPv6 provides an address self-configuration mechanism (Stateless mechanism). The nodes can define their addresses in an autonomous manner.
5. ***Neighbour discovery (ND):*** IPv6 provides ND to enable the nodes to discover each other's presence, to determine each other's link-layer address, and to find routers and maintain reachability information about the active neighbour. It effectively replaces ARP by adding a new message for ICMPv6.
6. ***Tiny stacks available**:* IPv6 application to the Internet of Things has been researched for many years. The research community has developed a compressed version of IPv6 named 6LoWPAN. It is a simple and efficient mechanism to shorten the IPv6 address size for constrained devices while border routers can translate those compressed addresses into regular IPv6 addresses. In parallel, tiny stacks have been developed, such as Contiki, which takes no more than 11.5 Kbyte.
7. ***Fully Internet compliant**:* IPv6 is possible to use a global network to develop one's own network of smart things or to interconnect one's own smart things with the rest of the World.

Appendix F: The Network Simulator - NS3

NS3 is an open source simulator for research and education purposes [153] that was started in 2006. The NS3 is not an evolved version of NS2, but is rather a new network simulation environment, which aims to offer an efficient environment to develop the communication networks. NS3 software is composed of C++ libraries that can work together in addition to external libraries to build the environment. NS3 uses the OOP concept to manage the libraries interactions. However, the user can use C++ or Python languages to write the code. These libraries are generally classified into four categories as follows: “core”, “simulation”, “common”, and “node” libraries [154]. The core library deals with the kernel of NS3 like debugging, random number generator, smart pointers and callbacks. The simulator library manages the time, schedulers, and events. The common library contains the interaction controller units like tracing and objects monitor. Finally, the node library manages the simulated network classes like node, net devices, and channel.

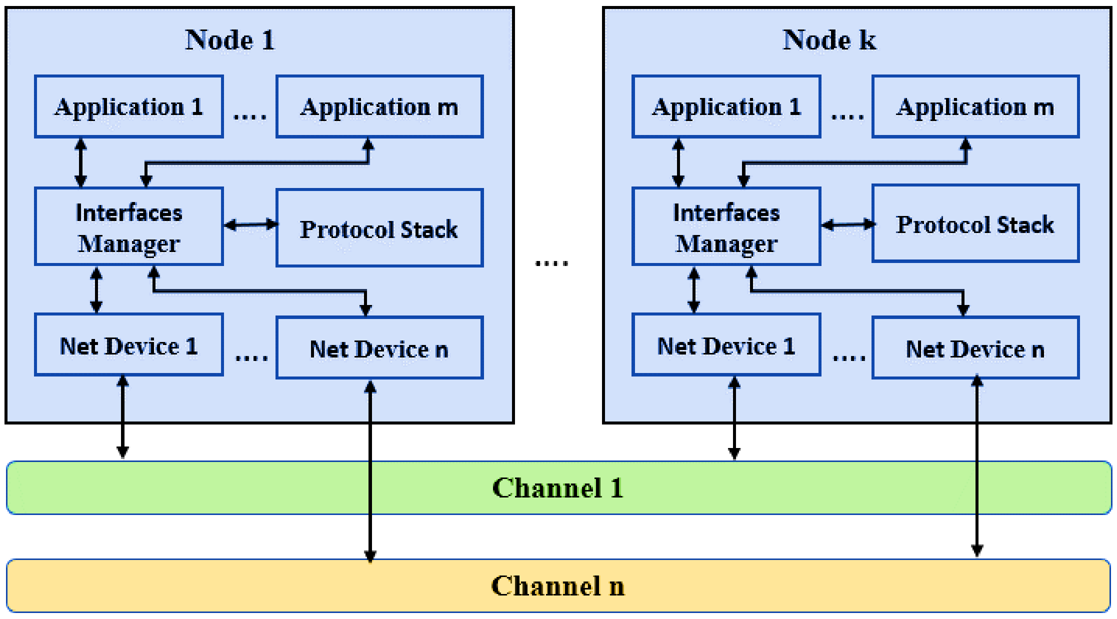


Figure F-1: The Base Classes’ Abstraction in NS3

A real network is abstracted in NS3 by a group of classes that represent the simulated network components. The core classes in each network are node, application, channel, and net device classes [154]. The node class represents the node type, i.e. end point node or router. The application class specifies the simulated user programmes that initiate activities in a specific scenario. The channel class mimics the carrier medium between the communicated nodes like wireless and Ethernet. Finally, the net device class represents the interfaces cards that connects nodes to the defined channels. These classes are aggregated by the NS3 simulator to simulate the components interaction of the real network. The previous figure illustrates the base classes' abstraction in the NS3 simulator.

Building a protocol stack with NS3 requires basically defining the required classes by the developer and then defining the classes' interaction relying on the low level API of the interfaces manager [154]. NS3 offers a helper classes to facilitate creating the new modules faster. Helper classes contains subclasses and APIs for each required function, it work as a broker by passing the new class variables to the internal APIs without a need to deal with them directly. NS3 provides a wide range of helpers to facilitate almost all the core processes in the simulation.

Appendix G: An Overview of the ProVerif Tool

The ProVerif is an automated formal security protocol verifier [138], [140], [147], [155]. ProVerif examines the protocol security functionality like authentication, secrecy, etc. by multiple executing the protocol concurrently. It supports many cryptographic function such as symmetric/asymmetric encryption, hash function, etc. To verify the protocol security, ProVerif analyses the protocol by applying unlimited sessions and messages [137]. ProVerif model contains three main parts: declarations, processes, and main. The declaration part declares channels, variables, functions, and security primitives. The process part is used to model the role of the participant parties. The main part is reserved to scrutinise the protocol.

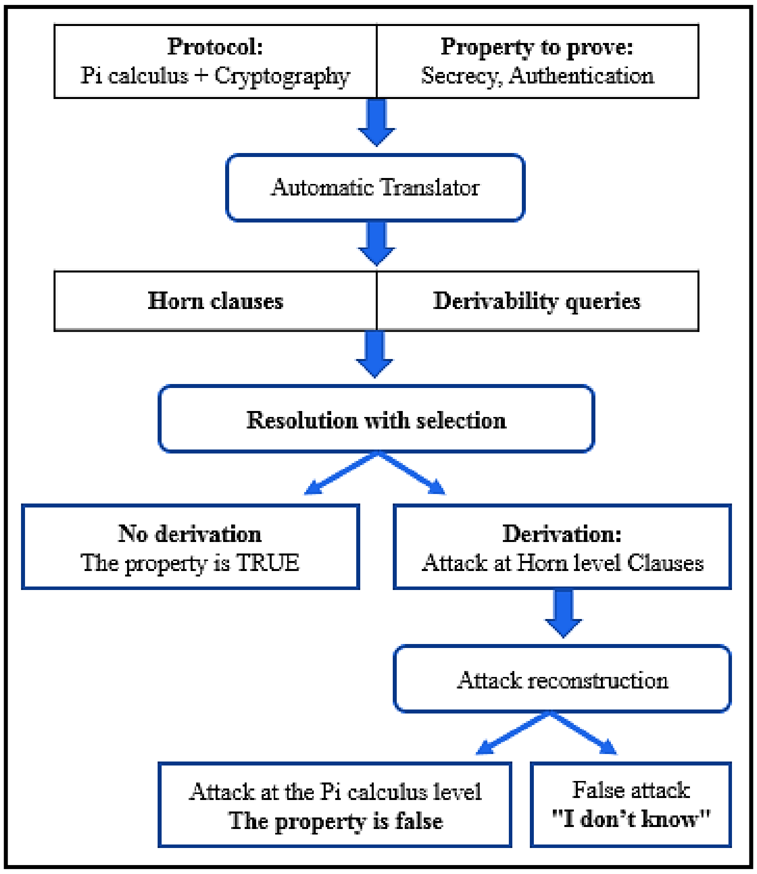


Figure G-1: ProVerif Structure [137]

ProVerif follows the following approach to verify the protocol security properties [137]. The above figure illustrates the ProVerif structure.

1. Representing the protocol with a cryptography the Pi calculus. The security properties that are targeted have to be declared as ProVerif input as well.

2. Automatic translation of the input information into two types of the ProVerif internal logic using Horn clauses as follows: (1) the protocol as a set of Horn clauses; (2) derivability queries to prove the required properties.
3. A resolution with free selection used by ProVerif to test the ability of deriving a fact from the clauses. If the fact is immune to this that means the intended security characteristic has been proven. Otherwise, the characteristic might suffer from an attack or a false attack resulting from the Horn clauses abstractions.