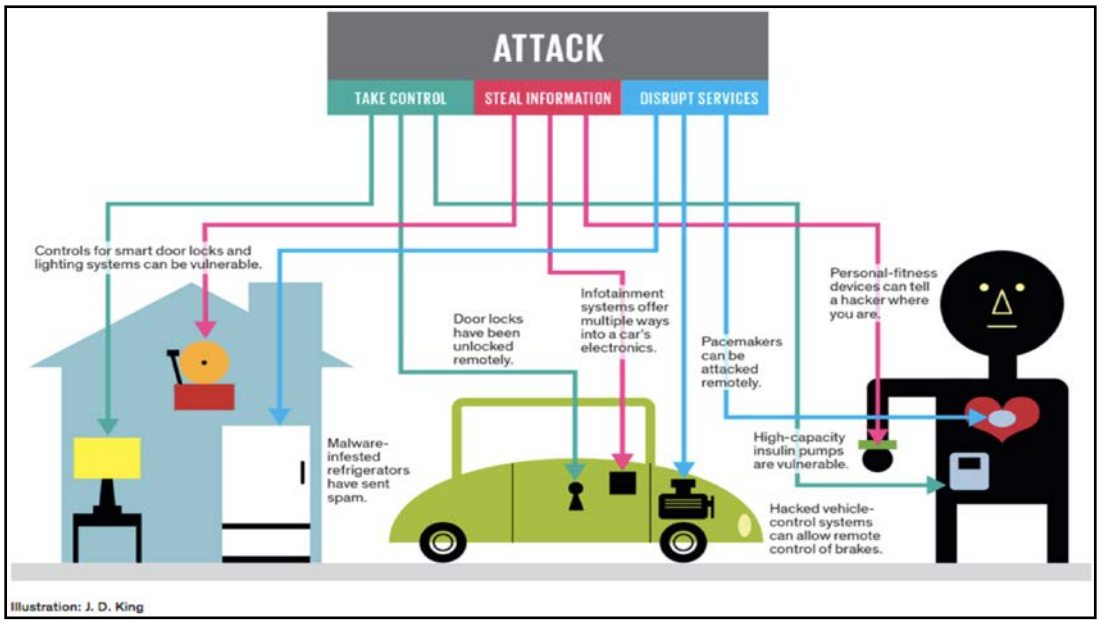
Security of Internet of Things for systems of systems

Djamel Eddine Kouicem

Par **Djamel Eddine KOUICEM**

Sécurité de l'Internet des objets pour les systèmes de systèmes

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 21 novembre 2019
Spécialité : Informatique et Sciences et Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253)
D2518

Sécurité de l'Internet des objets pour les systèmes de systèmes

Djamel Eddine Kouicem

Soutenue le 21 novembre 2019

Spécialité : Informatique et Sciences et Technologies de l'Information et des Systèmes

---

Composition du jury:

Président:

Aziz Moukrim
Professeur des universités
Univ. de Technologie de Compiègne

Rapporteurs:

Dominique Gaïti
Professeur des universités
Univ. de Technologie de Troyes

Pascal Urien
Professeur des universités
Télécom ParisTech

Examinateurs:

Houssam Affifi
Professeur
Télécom SudParis

Ken Chen
Professeur des universités
Univ. de Paris 13

Hicham Lakhlef
Maitre de conférences
Univ. de Technologie de Compiègne

Directeur de Thèse:

Abdelmadjid Bouabdallah
Professeur des universités
Univ. de Technologie de Compiègne

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

Remerciements

Je tiens tout d'abord à remercier le directeur de cette thèse, M. Abdelmadjid Bouabdallah, professeur des universités à l'université de Technologie de Comipègne, qui m'a confié cette thèse, puis je le remercie pour m'avoir guidé, encouragé, conseillé tout en me laissant une grande liberté.

Je voudrais remercier les rapporteurs de cette thèse M. Pascal Urien, Professeur des Universités à Télécom ParisTech, et Mme. Dominique Gaiti, Professeur des Universités à l'université de technologie de Troyes, pour le temps qu'il ont consacré pour relire cette thèse en tant que rapporteurs et pour les remarques pertinentes vis-à-vis de mon travail. J'addresse également mes remerciements à M. Azziz Moukrim, Professeur à l'Université de Technologie de Compiègne, pour avoir présidé mon jury de thèse.

J'associe à ces remerciements M. Houssam Afifi, professeur à l'Institut Mines Télécom SudParis, M. Ken Chen, Professeur des universités à l'université Paris 13 et Hicham Lakhlef, maitre de conférences à l'université de technologie de Compiègne, qui m'ont fait l'honneur d'avoir accepté d'examiner mon travail et participer au jury de ma thèse.

Je tiens à remercier tous les collègues au laboratoire Heudiasyc, pour leur aide et leur bonne humeur. Nous avons partagé de bons moments.

J'exprime ma plus profonde reconnaissance à mon grand père, ma mère et mon père, mes frères et soeurs pour leur soutien sans faille tout au long de ces années. Je tiens à leur dire à quel point leur soutien affectif a été important à mes yeux, ce travail a pu être mené à son terme grâce à leur soutien. Ce travail leur est légitimement dédié.

Table des matières

Liste des figures

Liste des tableaux

Publications

International Journal Publications

1. Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, Hicham Lakhlef. Internet of Things Security : A top down survey. In Journal of Computer Networks, 141(2018),pp.199-221.

2. Djamel Eddine Kouicem, Youcef Imine, Abdelmadjid Bouabdallah, Hicham Lakhlef. Decentralized Trust Management Based Blockchain Protocol for Internet of Things. In IEEE Transactions on Dependable and Secure Computing. Under revision.

International Conference Publications

1. Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, Hicham Lakhlef. Distributed Fine Grained Secure Control of Smart Actuators in Internet of Things. In 15th IEEE International Symposium on Parallel and Distributed Processing with Applications, Dec 2017, Guangzhou, China.

2. Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, Hicham Lakhlef. An Efficient Architecture for Trust Management in IoE Based Systems of Systems. In 13th IEEE System of Systems Engineering Conference, JUNE 19-22, 2018, Paris, France.

3. Youcef Imine, Djamel Eddine Kouicem, Ahmed Lounis, Abdelmadjid Bouabdallah. MASFOG : An Efficient Mutual Authentication Scheme For Fog Computing Architecture. In 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications, July 31th - August 3rd, 2018, New York, USA.

Resumé

l'Internet des objets (IoT) est une nouvelle technologie qui vise à connecter des milliards d'objets physiques à Internet. Les composants de l'IoT communiquent et collaborent dans des environnements distribués et dynamiques, confrontés à plusieurs problèmes de sécurité de grande ampleur. La sécurité est considérée parmi les enjeux majeurs de l'IoT et soulève des défis liés aux contraintes de capacité de calcul et stockage ainsi que le très grand nombre des objets connectés. Dans cette thèse, nous nous intéressons à l'application des outils cryptographiques ainsi que la technologie blockchain pour résoudre les problèmes de sécurité dans l'IoT, à savoir : l'authentification et la gestion de confiance. Dans un premier lieu, nous nous sommes intéressés au problème du contrôle d'accès distant des actionneurs intelligents utilisant des dispositifs IoT. Pour addresser ce problème, nous avons proposé une solution de contrôle d'accès efficace et à granularité fine, basée sur le mécanisme ABE (Attribute Based Encryption) et des chaînes de hachage. À l'aide d'outils formels d'analyse de sécurité, nous avons démontré la sécurité de notre protocole face aux attaques malveillantes. Dans un deuxième lieu, nous avons adressé le problème d'authentification dans les applications IoT basées sur le paradigme du fog computing. Nous avons proposé un nouveau protocole d'authentification mutuelle efficace qui est basé sur la technologie blockchain et la cryptographie à seuil. Dans notre solution, les objets IoT et les serveurs de fog n'ont besoin que de quelques informations à stocker pour vérifier l'authenticité de chaque objet du système. L'authentification est effectuée seulement sur la bordure du réseau sans passer par des entités externes. Ainsi, la latence et la capacité de stockage sont réduites au maximum. Enfin, dans notre troisième contribution, nous avons proposé un nouveau protocole de gestion de réputation basé sur la technologie blockchain et le fog computing, avec la prise en charge de la mobilité des objets connectés. Notre protocole permet aux objets IoT d'évaluer et de partager avec précision la réputation relative aux autres objets de manière scalable, sans se recourir à une entité de confiance. Nous avons confirmé l'efficacité de notre protocole par des analyses théoriques et des simulations approfondies. Nous avons montré que notre protocole surpasse les solutions existantes, notamment en matière de scalabilité, prise en charge de la mobilité, la communication et le calcul.

Mots Clés : *Internet des objets, Sécurité, Blockchain, gestion de réputation, authentification.*

Abstract

The Internet of things (IoT) is a new technology that aims to connect billions of physical devices to the Internet. The components of IoT communicate and collaborate between each other in distributed and dynamic environments, which are facing several security challenges. In addition, the huge number of connected objects and the limitation of their resources make the security in IoT very difficult to achieve. In this thesis, we focus on the application of lightweight cryptographic approaches and blockchain technology to address security problems in IoT, namely : authentication and trust management. First, we were interested on some kind of IoT applications where we need to control remotely the execution of smart actuators using IoT devices. To solve this problem, we proposed an efficient and fine-grained access control solution, based on the Attribute Based Encryption (ABE) mechanism and one-way hash chains. Using formal security tools, we demonstrated the security of our scheme against malicious attacks. Second, we tackled the problem of authentication in IoT based fog computing environments. Existing authentication techniques do not consider latency constraints introduced in the context of fog computing architecture. In addition, some of them do not provide mutual authentication between devices and fog servers. To overcome these challenges, we proposed a novel, efficient and lightweight mutual authentication scheme based on blockchain technology and secret sharing technique. We demonstrated the efficiency of our authentication scheme through extensive simulations. The third problem treated in this work is the trust management in IoT. Existing trust management protocols do not meet the new requirements introduced in IoT such as heterogeneity, mobility and scalability. To address these challenges, we proposed a new scalable trust management protocol based on consortium blockchain technology and fog computing paradigm, with mobility support. Our solution allows IoT devices to accurately assess and share trust recommendations about other devices in a scalable way without referring to any pre-trusted entity. We confirmed the efficiency of our proposal through theoretical analysis and extensive simulations. Finally, we showed that our protocol outperforms existing solutions especially in terms of scalability, mobility support, communication and computation.

Key Words : *Internet of Things, blockchain, trust management, authentication.*

Introduction

Context and research topic

Internet of Things (IoT) emerged as a new paradigm in which the network connects several physical objects that have the ability to collect and transfer/exchange data over a network and collaborate in order to perform high level tasks without requiring human-to-human or human-to-computer interaction. These objects can be engaged in complex relationships including the composition and collaboration with other independent and heterogeneous systems in order to provide new functionalities, thus leading to the *Systems-of-Systems* (SoS). the concept of Systems of Systems (SoS) can be defined as complex systems composed of several independent sub-systems that work together to achieve a common goal. IoT fits very well with the concept of SoS and shares the same properties of SoS. Indeed, the components of IoT are *heterogeneous*, *autonomous* and are managed by different entities and owners. In addition, IoT components are *geographically distributed* and are constantly evolving and interacting in dynamic environments with other complex systems such as cloud and fog computing. These complex interactions can lead to *emergent behaviorss*, which is one of the fundamental properties of the SoS.

One of the issues that potentially threaten IoT is the security and the privacy of exchanged/collected data that are often deeply linked to the life of users. These considerations lead us to underline the importance of enforcing security mechanisms in IoT applications which play a pioneer role in mitigating IoT risks. Security problems in IoT are most challenging than the existing security problems in Internet of nowadays. Indeed, it is instructive to note that the things are highly resources-constrained in terms of computing capacity, memory and energy which make the existing security solutions absolutely not applicable in this context. Moreover, the high number of connected objects, estimated by Cisco to be about 50 billions of objects by 2020, raises scalability issues.

This thesis is part of the labex MS2T (Control of technological Systems-of Systems). In the context of integrating IoT systems in order to compose complex, large-scale SoS, the main goal of this thesis is devoted to the study the IoT based SoS key challenges and the development of global approaches for securing IoT

based SoS architecture. We investigate different aspects of IoT security including confidentiality, authentication, trust management using cryptographic approaches and new emergent technologies such as blockchain technology.

Contributions

Hereafter, we highligh the main contributions of this thesis.

A top down survey on IoT security : in our first contribution [82], we have accomplished a comprehensive survey that includes the most relevant security challenges in different IoT applications and the existing solutions in the literature. We surveyed existing research works in a new approach which is a top down approach. We studied the most relevant aspects such as lightweight cryptographic approaches, blockchain, Software Defined Networking, the context awareness and the relationships between security and safety in IoT. In the first part of our survey, we presented the different challenges and security requirements inherents to the well known IoT applications. Then, we surveyed the literature solutions according to two main points of view : (1) classical approaches which are generally operating in centralized environments where we have central trusted entities ensuring the proper functioning of security services ; (2) new emerging security solutions which can be handled in decentralized infrastructure.

A new solution to secure remote control of IoT actuators : nowadays, with the advent of Internet of Things, we need efficient mechanisms to secure and control remotely IoT smart actuators by users and controllers using smartphones and IoT devices. We mean by remote control all the actions that could be performed remotely on smart objects. This arises particularly in industrial Cyber-Physical Systems to supervise industrial processes. However, the complex environment of IoT systems makes this task very difficult to achieve regarding the number of connected objects and their resources limitation. In this contribution [80] we tackled the problem of remote secure control of IoT actuators. For that, we proposed a distributed lightweight fine-grained access control protocol based on Ciphertext Policy Attribute Based Encryption mechanism and one way hash chain. The results of formal security analysis, using AVISPA tool, demonstrated that our scheme is secure against various attacks. Moreover, the performance evaluation results demonstrated the scalability and the efficiency of our solution in terms of energy consumption and computation costs.

An Efficient Mutual Authentication Scheme For Fog Computing Architecture : in order to efficiently manage the huge number of IoT objects and data, a new architecture called fog computing has been introduced recently as

the convergence between the Internet of things and cloud computing paradigms. This architecture aims to extend cloud-computing services to the edge of the network. With the new fog-computing paradigm, new challenges appear in prospect as authentication, which is one of the most important challenges.

We developed a new scalable and lightweight authentication scheme that meets the main requirements of IoT [74]. For that, we proposed a novel and efficient authentication protocol which ensures mutual authentication at the edge of the network. Our scheme performs a first registration in the cloud level, and then it uses credentials provided by the cloud to realize any eventual mutual authentication with the fog nodes without any resort to the cloud. In addition, our solution takes into consideration the eventual authentication between fog nodes. We base our construction on blockchain technology and secret sharing technique. The Blockchain is maintained by fog nodes and it allows end users to authenticate any fog node in the architecture. In addition, it allows fog nodes to establish mutual authentication with each other. We showed through experimentations the efficiency of our authentication scheme which provides a low overhead in terms of storage capacity and computation.

An efficient Trust Management Protocol in IoT based on Blockchain technology : IoT can be viewed as service centric architecture where each device, or thing in general, can request services from other devices and it may also provide services for other devices (service providers). IoT service providers may behave dishonestly and maliciously for the purpose of promoting IoT devices to select them for one or many services on behalf of other trusted service providers. Therefore, it is clear that developping a trust management protocol to protect IoT devices from malicious service providers is more than necessary. We proposed a new scalable trust management architecture which is based on blockchain technology and fog computing paradigm [81]. Our solution allows IoT devices to accurately assess and share trust recommendations about other devices in a scalable way without referring to any pre-trusted entity. The blockchain is maintained by powerful fog nodes which offload lightweight IoT devices from trust information storage and heavy computations and save their bandwidth occupations. We have extended this work to develop a new trust management protocol upon the proposed architecture. Our protocol, named **BC-Trust** [83], is efficient and resilient against the known malicious attacks such as bad-mouthing, ballot-stuffing and cooperative attacks. We confirmed the efficiency of our proposal through theoretical analysis and extensive simulations. Finally, we showed that our protocol outperforms existing solutions especially in terms of scalability, mobility support, communication and computation.

Content of the thesis

This dissertation is organized into seven chapters. The first chapter sets the context of our research. Chapter 2 introduces the required technical backgrounds about the Internet of Things (IoT), which are necessary for the comprehension of this thesis. We also discuss the main core concepts around this technology as well as its technical challenges. In chapter 3, we propose a top down survey in which we present the different solutions proposed in the literature dealing with security in IoT. In particular, we discuss the benefits that emerging technologies such as blockchain and SDN (Software Defined Networking) and what they can bring to the IoT security in terms of scalability and efficiency. Finally, we propose a classification and a comparison of existing solutions. In chapter 4, we address the problem of remote secure control of smart actuators. For this purpose, we propose a distributed lightweight fine-grained access control based on Attribute Based Encryption scheme and one-way hash chain. Using a security analysis tool based on formal methods, we demonstrated that our scheme is secure against various attacks. The obtained simulation results show that our solution is highly scalable and efficient in terms of computation and storage.

In the rest of the dissertation, we focus mainly on the application of blockchain technology to address security problems in IoT. We propose two original solutions to address the authentication and trust management problems. In chapter 5, we propose a new efficient mutual authentication scheme in IoT based on fog computing architecture. This authentication scheme has the advantage to be performed at the edge of the network and hence reduces the latency of authentication. Beside, we demonstrate through simulations that our authentication scheme provides a low overhead in terms of storage capacity and computation. In chapter 6, we propose a new scalable trust management blockchain-based protocol for IoT. The blockchain is maintained by powerful fog nodes which offload lightweight IoT devices from trust information storage and heavy computations and save their bandwidth occupations. We discuss the main components of our protocol and how it deals against malicious attacks and its main benefits, especially high mobility support of devices. We confirm the efficiency of our proposal through theoretical analysis and experiments. Finally, we show that it outperforms existing solutions especially in terms of scalability, mobility support, communication and computation.

We end this dissertation by chapter 7, in which we give concluding remarks and summarize our main contributions. We highlight the perspectives and future work directions of our thesis.

Backgrounds about Internet of Things

Nowadays, Internet of Things (IoT) is changing much about the world we live in, the way we drive, how we make decisions, and even how we get energy. Internet of things consists of sensors, actuators and chips embedded in the physical things that around us by making them smarter than ever. These things are connected together and exchange data between them and with other digital components without any human intervention [5]. IoT contributes significantly to enhance our daily life throughout many applications come from different sectors such as smart cities, smart building, healthcare, smart grids, industrial manufacturing among others.

In this chapter, we present the main definitions of the concepts of IoT and its main components. We shed the light on its typical architectures, its main applications. We also provide an overview of some of the key challenges relating to this technology. Moreover, we explore the relations between the IoT and other emerging technologies such as cloud and fog computing and how these emerging technologies are necessary for the development of future IoT.

1.1 Internet of Things

The Internet of things is an enabling technology that allows connecting heterogeneous objects through the Internet. This technology is about the pervasive presence of sophisticated sensors and chips that are embedded in the physical objects around us. These objects work together to achieve common goals by sensing, transmitting and processing valuable data [5]. This vision of the Internet of Things has introduced a new dimension to information and communication technologies : in addition to people and computers, others physical objects will allow us to be connected to Internet from anywhere and at any time. However, these objects bring new challenges due the the low memory, energy, and compuataion capacity.

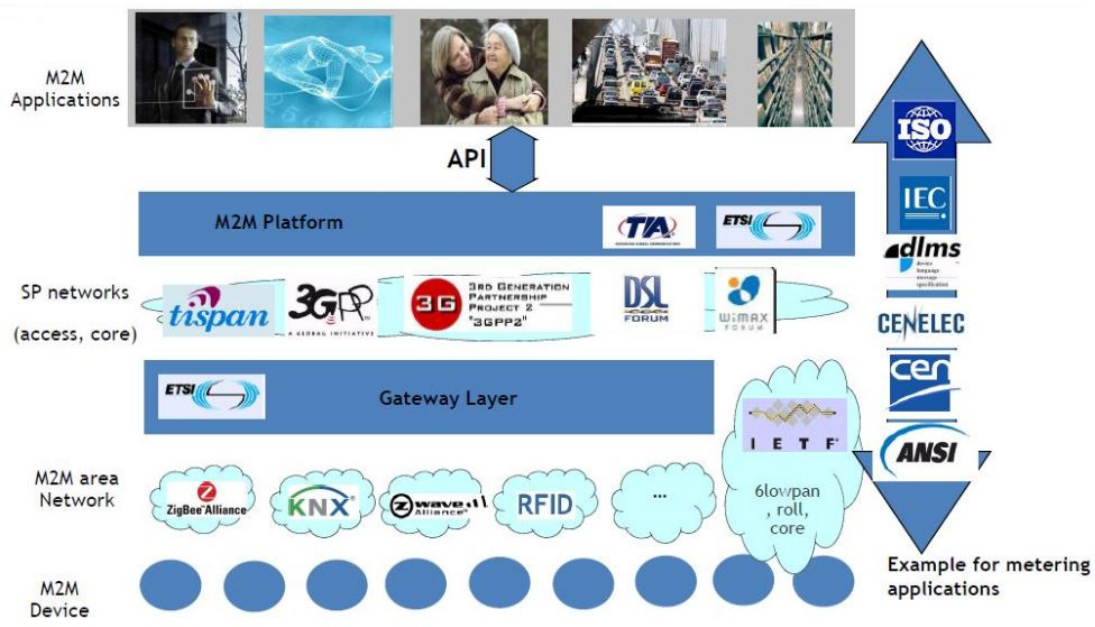


Figure 1.1 – The architecture of IoT (source [136]).

1.1.1 IoT architecture

Many working groups and consortiums such as oneM2M, ETSI, ATIS, TIA, etc. proposed some generic and horizontal architectures of the IoT, that will position themselves transversely, regardless the high level IoT applications. The most common elements between the large pool of the proposed architectures is that they consider the IoT as an evolution of the M2M (Machine-to-Machine) paradigm.

In Fig. 1.1, we present the typical architecture of the IoT which is proposed by ETSI (European Telecommunications Standards Institute) [136]. This architecture considers three main layers which are M2M domain layer, Network domain layer and application layer. In the recent literature, other models, such as 5-layers model, have been proposed to improve this basic 3-layers model, and hence be able to support the scalability of the IoT. Among the proposed models, the five-layer model is the most suitable model for IoT applications. Next, we provide a brief discussion the five layers of the IoT architecture.

1.1.1.1 Objects Layer

This layer, known also as the perception layer, includes the physical devices, sensor and actuators that are equipped by sensing, actuating and communication capabilities that allow them to observe and construct the perception of its physical environment. This bottom layer is the most important layer of this model and is responsible for the generation of valuable data such as temperature, pressure, weight,

motion, vibration, acceleration, location, humidity, etc. This layer is connected to its upper layer (object abstraction layer) through communication channels using WIFI, Zigbee, 3G/4G/5G standards [136]. The perception layer, thanks to the huge amounts of data created at this layer, is the entry point of what we commonly call Big Data.

1.1.1.2 Object Abstraction Layer

This layer includes all edge components that provide connectivity among the devices in the objects layer and other gateways located at the edge of the objects layer. At this layer, gateways and edge computing servers can perform local analysis and transfer data to the service management layer through various technologies such as RFID, 3G, GSM, UMTS, WiFi, ZigBee, etc. This layer is introduced as an intermediate layer to manage the tremendous number of objects and support some sensitive IoT applications [136].

1.1.1.3 Service Management Layer

In this layer, we find M2M platforms, middleware, API of M2M applications and also cloud computing technologies that can manage data generated at the bottom layers. This layer enables IoT application developers to develop high level applications that are independent of any physical platform and the underlined technologies. At this layer, it is possible to handle the received data, deliver the required micro-services to developers and managers to make decisions and develop business applications.

1.1.1.4 Application Layer

The application layer interfaces with final users and customers through application protocols such as HTTP, MQTT, CoAP, etc. This layer provides high-quality services to users, and it contains all the required softwares to meet the requirements of final customers. These applications should be designed in such away to answer to many markets' needs in different fields such as smart building, transportation, industry, smart grids and healthcare. These applications are deployed on powerful cloud servers to satisfy a good quality of service for final customers and ensure a good level of reliability.

1.1.1.5 Business Layer

The business layer consists on the upper layer of the five-layer model. Its main role is the management of the overall IoT systems, services, applications and users. This

layer takes as an input the data received from the application layer in order to develop more effective business models, predict the customer behaviors, show high level metrics, graphs and flowcharts, etc. Moreover, we can implement and perform big data analytics in order to transform data and information into actions to support decision-making processes. In addition, application and customer monitoring and management are achieved at this layer.

1.1.2 IoT Applications

The IoT cover a wide range of applications and will touch almost every area we face on our daily lives. Among these applications, we can mention the following :

1.1.2.1 Smart Grids

Electrical energy is a treasure which has a very high industrial value, and plays an important role in economic development. Nowadays, we use very modern IT technologies to optimize electricity production by taking into account user demands throughout the electricity distribution line. The smart grid is the technology behind this distribution line. It consists of an integrated network, called also the advanced metering infrastructure (AMI) installed between the electricity production centers and the end customers, whose important role is to coordinate the electricity production with respect to the consumption of end customers. Smart grids represent one of the most attractive areas in IoT. The main goal is to improve the quality of experience of final customers and optimize the electricity production. To better understand in more details how IoT can improve the electricity production in smart grids, the reader is referred to [88, 45].

1.1.2.2 Healthcare

Smart healthcare plays a significant role in healthcare applications through embedding sensors and actuators in patients' bodies for monitoring and tracking purposes. The IoT is used in healthcare in order to monitor physiological statuses of patients. The embedded sensors have the ability to collect information directly from the body area of the patient and transmit it to the physician. This technology has the potential to completely detach the patient from the centralized system which is the hospital while maintaining continuous contact with the physician. Currently, Healthcare based IoT applications represent one of the promising technologies that impact hugely the society which is mainly due to the aging of the population. Indeed, in France, the percentage of people over the age of 60 reached about 24% of the

population in 2015 and will rise to 32% by 2060 [1]. Furthermore, the budget reserved for healthcare applications reached about 12% of the GDP (Gross domestic product) [2]. In this context of population aging and the cost related to the treatment, a great interest emerges to adopt new IoT based technologies to monitor the patients in real time.

1.1.2.3 Transportation systems

Intelligent transportation systems (ITS) represent the next generation of transportation that aims to link people, roads and intelligent vehicles thanks to the development of embedded systems and communication technologies. By connecting and distributing intelligent processors inside vehicles and also through transportation infrastructure, we can make the transportation safer, greener and more convenient. ITS employs four main components, namely : vehicle subsystem (consists of GPS, RFID reader, OBU, and communication), station subsystem (road-side equipment), ITS monitoring center and security subsystem [85]. Connected vehicles are becoming more important with the aim to make driving more reliable, enjoyable and efficient [55]. Actually, we have three types of communications in vehicular networks : V2V (Vehicle to Vehicle), V2I (Vehicle to Infrastructure) and V2P (Vehicle to Pedestrian) [85]. However, recently, a new type of communication has emerged, called V2G (Vehicle to Grid), whose main goal is to ensure electrical Vehicles charging based on energy of smart grid electricity distribution [88].

1.1.2.4 Smart cities

Smart cities consist of one of the most important applications of IoT. Although, there is no formal definition of "smart city", it consists of a new emerging paradigm that aims to enhance the usage of public resources, increase the quality of service to citizens [143]. In this context, sensors are deployed all over roads, buildings, smart cars, etc. to better manage traffic, adapt to the weather, lighting follows the position of the sun, domestic incidents can be avoided with alarms, etc.

1.1.2.5 Manufacturing

Nowadays, IoT plays an important role in the industry. It is considered as a promising solution to automate the process of manufacturing and the control of the production chain. Industrial Internet of Things (IIoT) uses new technologies such as Machine-to-Machine (M2M) communication, Wireless Sensor Networks (WSN),

[1] https ://www.insee.fr/en/statistiques/1281166
[2] https ://www.insee.fr/fr/statistiques/1906695 ?sommaire=1906743

automation technologies as well as Big Data to create an intelligent industrial ecosystem [115]. The main aim of IIoT is to provide better productivity, efficiency, reliability and better control of final products.

1.2 IoT challenges

The IoT is a big industry that should bring a lot of business opportunities and benefits. The IoT accentuates the considerable positive impact already produced in our society, and thus a real change of socio-economic and cultural models. However, these benefits cannot be ultimately achieved without addressing many tricky challenges and issues. We enumerate in the following the main challenges that IoT faces :

1. **Scalability** : the tremendous number of objects connected to the Internet should be considered in many IoT protocols. The number of connected devices surpassed the number of humain population in 2010. Therefore, it is important to design scalable architecture and build efficient protocols that can deal with IoT scalability issues.

2. **Limitation of resources** : most of IoT devices are limited in terms of storage and computation capabilites. Thus, it is important to design lighweight protocols that support this ressource limitation and meet the requirements of customers.

3. **Reliability** : IoT systems are vulnerable to many safety and reliability issues that can cause huge damages. It is mandatory to design reliable systems that works properly under any circumstances especially when it comes to emergency and critical applications like manufacturing, transportation and healthcare applications [136].

4. **Management** : one of the most challenging tasks in IoT is how to provide real-time, lightweigt and secure management protocols to manage Fault, Configuration, Accounting, Performance and Security (FCAPS) of the connected devices.

5. **Mobility** : is another important challenge as most of IoT devices are mobile. It is very important that IoT services be delivered to end mobile user with respecting their requirements.

6. **Interoperability** : IoT applications, platforms, devices and protocols are likely to be heterogeneous. For instance, there is no unique standard that

can support interconnection between all the heterogeneous IoT systems [136]. Therefore, there is a real need to take into consideration the hetereneity aspects of the IoT to build applications that can be easily maintained, extended and integrated with other systems and applications.

7. **Security and privacy** : this challenge is probably the hardest one that hinder the evolution of the IoT. These last years, security and privacy challenge are considered as one of the most important research field in IoT.

1.3 Fog Computing in Support of the IoT

With the tremendous growth in the amounts of data that IoT objects report, we need new emerging technologies that can act as a bridge between IoT devices and cloud computing. During the past several decades, cloud-computing technology showed its efficiency as a powerful tool to store, process, and analyze this data while meeting high level application requirements. However, the centralized nature of the cloud remains a serious problem to meet the requirements of some emerging IoT applications that require very low latency.

Recently, fog computing paradigm has been introduced as an extension of cloud-computing services to the edge of the network. This new architecture integrates network edge devices to overcome several cloud computing limitations related to bandwidth and latency. Fog computing architecture introduces a new rich service layer (computation, communication, storage and control operations) able to interact with any end device with any connection mode such as 4G, WIFI, etc. In addition, thanks to their proximity to the end-users compared to the cloud data-centers fog computing can impressively reduce latency, and provides the expected interoperability and reliability [136, 5]. This paradigm is realized by adding a resource-rich extra layer composed of a large number of edge devices such as routers, base stations, servers located on existing access points, etc. These fog devices provide low latency mobile services while ensuring the connection with cloud data centers for any eventual control operations and data aggregation.

We can consider fog computing as an optimal choice for the IoT designers thanks to the following features [136] :

- **Reduced Latency** : fog resources are close to end mobile users and objects. Thus, all processing and storage operations can be performed at the edge of the network without referring to cloud servers, which allows to provide better delay performance.

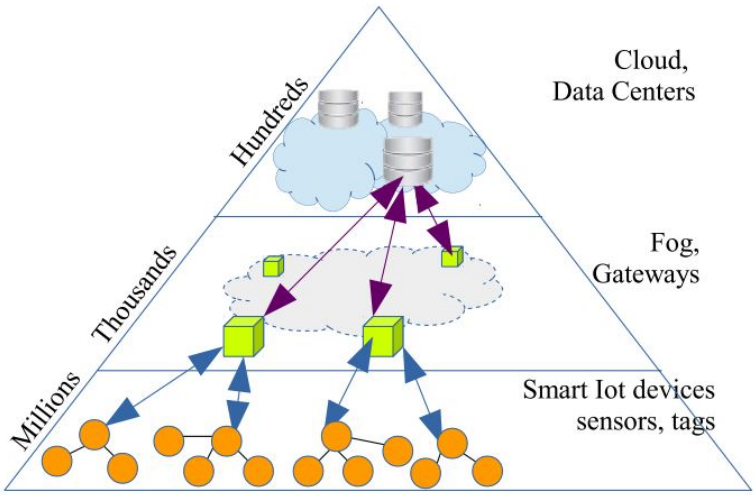


Figure 1.2 – Typical architecture of IoT based fog and cloud computing (source []).

- **Bandwidth reduction** : thanks to storage capabilities and proactive caching mecanisms that are deployed at the edge of the network, fog nodes can reduce the bandwidth consumption of some hungry applications up to 67% [136].

- **Energy efficiency** : in fog computing architecture, heavy data processing and storage operations are offloaded to powerful fog nodes. Therefore, this will substantially optimize storage and computation resources at IoT devices level.

- **Distribution** : fog computing architecture is realized based on "micro" clusters with storage, processing and communication capabilities spread all over the edge of the network. Indeed, it is possible to deploy many "micro" clusters closer to the end-users with a reduced cost compared to cloud data-centers.

- **Scalability** : fog allows IoT systems to be more scalable such that as the number of end-users increase, the number of deployed "micro" fog centers can increase to cope with the increasing load. Such an increase cannot be achieved by the cloud because the deployment of new data-centers is cost prohibitive.

- **Data and service replication** : Fog helps to provide efficient mechanisms to deal with data management and replication of services, which allows to achieve high resilliency and availability.

- **Mobility support** : fog nodes deal efficiently with high mobility scenarios and can track the location of IoT devices to benefit from this information in other decision making services.

- **Standardization** : fog resources is designed in such away they can interoperate with various cloud providers and can support many IoT protocols.

1.4 Conclusion

The Internet of Things (IoT) is one of the promising technologies that has attracted much attention from researchers in the industrial and academic sectors in the last years. As things add capabilities and as more people and new types of information are connected, IoT becomes an Internet of Everything (IoE). We have presented in this chapter the main definitions of these technologies, their architectures, their applications and also their main challenges.

In future, the IoT will raise questions, which will directly concern the security of properties and people. For example, some applications may be closely related to strategic infrastructure such as water and electricity supply, monitoring bridges and buildings, while others will manage information related to the privacy of people like their travels and health conditions.

Internet of Things Security : State of the art

As new emergent technology, IoT suffers from several security issues which are most challenging than those from other fields regarding its complex environment and resources-constrained IoT devices. These last years, a lot of researches are leading to address the various security challenges closely related to IoT such as key management issues [123], confidentiality, integrity, privacy, policy enforcements [121, 122] among many other challenges. The main works in the literature tried to adapt the security solutions proposed for wireless sensor networks (WSNs) and Internet in the context of IoT. However, we must point out that IoT's challenges take a new dimension which is far from being easy to overcome with traditional solutions. In addition, we must emphasize that most security approaches rely to centralized architectures, making their applications in IoT much more complicated regarding the large number of objects. So, distributed approaches are required to deal with security issues in IoT.

In this chapter, we survey the different solutions according to two perspectives, namely the security approaches based on traditional cryptographic approaches and the other approaches based on new emerging technologies such as SDN and Blockchain. Furthermore, we provide a top down review and a comparison study that gives a holistic view of the security in Internet of Things. This review encompasses the different aspects of security in IoT by starting from generic to specific aspects.

2.1 Background on security services

Security consists of all the techniques that aim to preserve, restore and guarantee the protection of information in computer systems from malicious attacks. Daily news puts security at the top of concerns : leakage of personal data and economic espionage, infection of sensitive computer systems, identity theft and fears about card payments are just few examples of threats. The security of computer networks and information systems in general, consists to provide the following services [107] :

- **Confidentiality** : it ensures that information is made unintelligible to unauthorized individuals, entities, and processes.
- **Integrity** : it ensures that data has not been modified by a third party (accidentally or intentionally).
- **Authentication** : it verifies that the sender of the message fits with its pretended identity.
- **Non-repudiation** : it ensures that the sender of the message can not deny having sent the message in the future.
- **Availability** : It ensures that the services of the system should be available for legitimate users.
- **Privacy** : It ensures that users' identities should not be identifiable nor traceable from their behaviors and their performed actions in the system.

Several cryptographic mechanisms have been put in place to deal with the different security threats and ensure the security services mentioned above. We provide in table 2.1 some of those mechanisms.

Security services	**Security mechanisms**	**Some examples**
Confidentiality	message encryption / sign-encryption	symmetric cryptographic mechanisms (AES, CBC, etc) ; asymmetric mechanisms (RSA, DSA, IBE, ABE, etc).
Integrity	hash functions, message signature	hash functions (SHA-256,MD5, etc) ; Message Authentication Codes (HMAC)
Authentication	chain of hash, Message Authentication Code	HMAC, CBC-MAC, ECDSA
Non-repudiation	message signature	ECDSA, HMAC
Availability	pseudo-random frequency hopping, Access control, Intrusion prevention systems, firewalls	Signature-Based Intrusion Detection, Statistical anomaly-based intrusion detection
Privacy	pseudonymity, unlinkability, k-anonymity, Zero Knowledge Proof (ZKP)	EPID, DAA, Pedersen Commitment

Tableau 2.1 – Security services and mechanisms

2.2 IoT Applications : security challenges

Internet of Things enables to improve several applications in various fields, such as, healthcare, smarts grids, smart cities, smart homes as well as other industrial applications. However, introducing constrained IoT devices and IoT technologies in such sensitive applications leads to new security and privacy challenges. In the following, we highlight the important security challenges of IoT applications :

- **Heterogeneity** : communication standards and information system technologies are heterogenueus. the communication between sensor nodes and servers or CPU units in general are done over Internet where networks, protocols and communication mediums are heterogeneous and have different security configurations. It is necessary to provide security standards that work with differents IoT plateforms and protocols.

- **Scalability issues** : the number of connected object has surpassed the number of population in the world and still grow continuosly. The management of security of all these devices present several challenges. For example, it is very hard to efficiently apply updates and security patches accross distributed environements with heteregeneous devices. Beside, the management of cryptographic keys of all these devices to build secure communications is actually a big challenge.

- **Vulnerabilities related to information and communication systems** : it is not enough to ensure security only at IoT devices level. Indeed, it is important to ensure that the communication between these devices and other plateforms and applications (like cloud and fog computing applications) is secure. Integrity, confidentiality and privacy of data, IP spoofing, injection, DoS/DDos attacks are just examples of attacks among others.

- **Data sensitivity and privacy** : IoT devices generate and exhange sensitive data like electricity consumption, health information, etc. These pieces of data must not be leaked by any unothorised entity.

- **Resources limitations** : most of connected devices have limited resources in terms of computation, memory and battery. Since the most of cryptographic solutions are computationally expensive, adapting them to ensure a high level of security while minimizing energy consumption is a hard challenge.

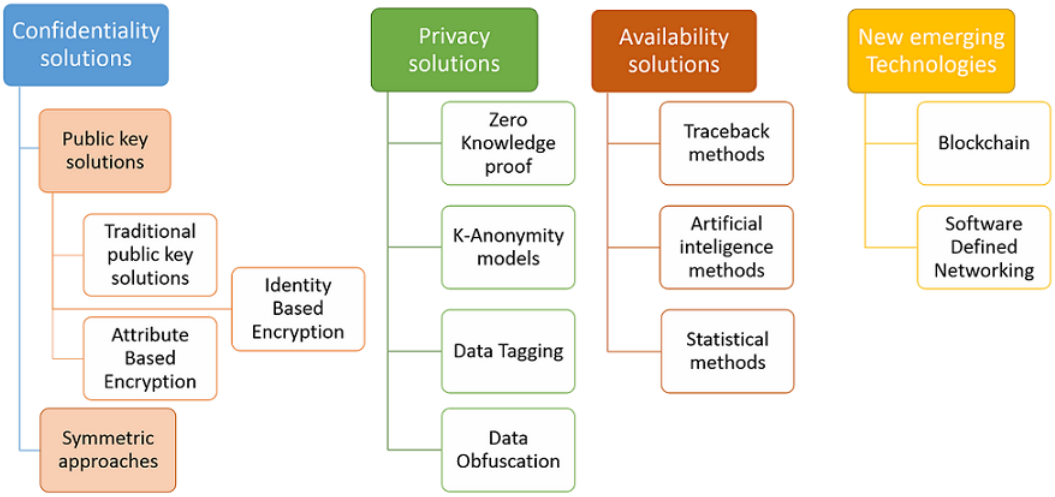


Figure 2.1 – IoT security solutions

2.3 Taxonomy of security solutions in IoT

Security subject is one of the hot research topic in IoT and has attracted a lot of researchers not only from academic and industry but also from standardization organizations. To date, there have been a lot of proposals aiming to address the security problems in IoT. In this section, we propose a classification of these solutions from an architectural point of view and we illustrate in figure 2.1, our classification of security solutions in Internet of Things. We distinguish in the light of this classification two main categories of approaches :

1. **Classical approaches** : this category of solutions groups the cryptographic based techniques that were especially designed for IoT communications or have been adapted from wireless sensor networks or M2M communications. In section 2.4, we present only the most significant solutions and we provide the main limitations of each proposal. We note that in this survey, we focus basically on solutions that ensure : confidentiality, privacy and availability services. It is worth mentioning that most of these solutions operate in centralized environments where we have central trusted entities ensuring the proper functioning of the security services. The cryptographic tools employed to ensure the security services are whether symmetric or asymmetric techniques that we will discuss by pointing out their main advantages and limitations in the context of IoT for each security service.

2. **New emerging security solutions** : This category groups security solutions that are based on new techniques other than cryptographic tools. They are more convenient to meet the scalability issues compared to cryptographic ap-

proaches. In general, the solutions belonging to this category are decentralized. In section 2.5, we focus on two emerging technologies :

(a) Software Defined Networking (SDN), which is a new network paradigm that is revolutionizing the world of networking this last years. Its aim is to provide an environment to develop more flexible network solutions and make the network resources more easy to manage using centralized SDN controller. Many SDN based security solutions for IoT have been proposed in the literature. We will discuss in more details these solutions in section 2.5.1.

(b) Blockchain technology, which is the technology behind the cryptocurrency tools such as bitcoin, aims to make the transactions between entities in a distributed manner (peer to peer architecture without referring to any central trusted server. Moreover, this solution does not require that entities trust each other. In this technology, it is piratically impossible to deny performed transactions once they are validated. Beside its application on the cryptocurrency domain, these last years, a lot of researchers have started to put the light on this technology in order to address security solutions in IoT such as data privacy, access control, etc. We present an analysis of these solutions in section 2.5.1.

We present mainly in section 2.5 the benefits of SDN and blockchain in terms of security, their key advantages, the issues that these technologies can solve and classical approaches can't and also their limitations.

2.4 Classical IoT security approaches

In this section, we review and discuss the main proposed solutions which are based on cryptographic approaches to address the main security services. Considering the traditional approaches, we focus on : confidentiality, availability and privacy services.

2.4.1 Confidentiality solutions

In the Internet of Things, it is mandatory to protect data exchanged between objects from attackers by means of encryption mechanisms. Hence, we should ensure that only legitimate users are able to disclose encrypted data. For this goal, cryptographic solutions exist to ensure data confidentiality, however, in most cases, these solutions are inefficient or even inapplicable in IoT devices with high resource constraints because they are based on algorithms that are very greedy in terms of storage and

computation. To get an idea about the energy consumption and the effeciency of the different cryptographic algorithms, the reader is invited to read the paper of Malina et al. [96] where intensive analysis was investigated to compare the different cryptographic primitives widely used in security and privacy. Considering the power limitation of smart objects, a lot of cryptographic solutions have been proposed to deal with resources constraint's issues. Basically, these solutions belong to two main classes, namely symmetric and asymmetric cryptographic solutions.

2.4.1.1 Symmetric key solutions

In Symmetric key schemes, each entity in the system should share cryptographic keys with all other entities in the system. The main advantages of symmetric based cryptographic solutions are their efficiency (they are less-computational) and easy to implement in hardware platforms. AES (Advanced Encryption Standard), RC4 and 3DES are just few examples widely used in practice. Although their efficiencies, symmetric key based security solutions suffer from scalability and key management issues. Indeed, this latter emerges as serious problem in Internet of Things where there are a lot of devices that exchange sensitive data in dynamic environments. In Symmetric key based solutions, each device must keep secret keys with all the devices evolving in the IoT system in order to exchange sensitive data. Basically, we can distinguish between two key distribution approaches [59], namely : 1) Probabilistic key distribution and 2) Deterministic key distribution.

In deterministic approaches, each entity must be able to establish a secure link with all other entities to form a full secure connectivity coverage. Therefore the number of shared keys in the system increases quadratically according to the number of entities (for n entities, we need $n(n-1)/2$ keys). Depending on the presence or not of a trust third party during key bootstrapping, we distinguish two sub-categories [59], namely : 1) offline key distribution approach where nodes can share session keys in a distributed way without the intervention of any central entity ; and 2) Server-assisted key distribution where we dispose of a central server that is charged of expensive cryptographic computations and attributes session keys to IoT devices. In contrast, Leap scheme [146] uses a temporary key which is kept in sensor nodes to generate session keys and is removed from the memory when the key agreement is done. For security purposes, Leap requires that sensor devices must not be exposed to attacks during a predefined time after the deployment. In [38], the authors proposed a memory-efficient key management scheme that reduces the storage to only $(n-1)/2$ keys per node. The main idea consists to introduce new mechanisms based on a hash function to generate half of symmetric keys while

storing the other half in sensors' memories.

In Probabilistic key distribution, it's not guaranteed that each node in the network shares a secure key with all other nodes, but the nodes share keys with their neighbors according to some probabilities in such way we must be able to form secure paths [1] between all entities in the network. With this approach, the scalability issues are solved but the key management protocols become less resilient in case of nodes' comprises. In the literature, there are a lot of probabilistic key management schemes. The first probabilistic key distribution scheme for WSNs is the scheme called Random key pre-distribution (RKP) proposed by Eschenauer et al. [51]. In this scheme, each node i in the network is pre-loaded randomly with a set of key ring R_i of size k, selected from a large pool S. After the deployment of sensor nodes, each node i broadcasts its keys' identifiers to its neighbors. The node i establishes a key session between some neighbor j only if the intersection between R_i and R_j contains at least one key ($R_i \cap R_j \neq \emptyset$), and thereby they choose one key among $R_i \cap R_j$ as a session key. In the case of ($R_i \cap R_j = \emptyset$), nodes i and j determine a secure path composed of secure links. The main drawbacks of this approach are its memory consumption required for keys storage and importantly its non resiliency against key compromise attacks. Indeed, if some nodes are compromised by an attacker, all the session keys that these nodes have established with their neighbors will be disclosed which corrupt fundamentally the security of the network. Some enhancements [29, 50] of the basic RKP scheme have been proposed ; namely : Q-Composite scheme [29] enhances the resiliency of RKP by introducing additional requirements in order to establish session keys between nodes, basically two nodes i and j can establish a session key only if they share at least Q keys used to compute a pairwise key obtained by computing the hash of all the concatenated shared keys. In [50], Du et al. proposed a solution to overcome key storage issue of RKP by establishing only the necessary session keys. On the other hands, Blom's scheme [18] is also a very efficient scheme that is very suitable for WSNs and IoT as claimed by some researchers [59]. In Blom's scheme, the secret keys are vectors obtained by simple matrix multiplications. The idea is that, each node i has an identifier I_i randomly generated and known by all nodes in the network. In the deployment phase, private key g_i for the node i is generated from its identifier as follows : $g_i = DI_i$, where D is a secret symmetric matrix generated over the finite field $GF(p)$ where p is a prime. For the node i, in order to share a secret key with node j, it computes $secret_{ij} = g_i^t I_j = g_j^t I_i$. Obviously the security of the scheme is strongly dependent of the secret matrix D which must be kept carefully by a trusted central server. This matrix is used also to add sensor nodes to the network.

[1] path composed from a set of successive secure links

2.4.1.2 Traditional Public key solutions

Traditional Asymmetric approaches group all the methods based on public keys and requires the authority to issue certificates to different users in the system. In this family, we find RSA, DSA, El Gammal, NTRU, ECC cryptosystems, etc. The advantages of these approaches are their flexibility, scalability and key management efficiency. However, these solutions are energy-consuming which are not suitable for constrained devices. NTRU consists of the less computational asymmetric approach based on the shortest vector problem in a lattice [104]. However, it requires more memory space to store keys. Elliptic curves are also in some cases very efficient and can ensure the same level of security as RSA and similar asymmetric cryptographic approaches with keys of small sizes [30]. Indeed, with 80-bit security level, we need only keys of 160 bit contrary to RSA where we need keys of 1024 bits.

The contribution in [92] is twofold. First a signcryption called DQAC scheme has been designed to sign and encrypt query messages which ensures authentication and confidentiality and it also preserves the privacy of users requesting WSNs' data. Second, a distributed access control based on the proposed signcryption scheme in addition to proxy based signature in order to anonymize users' identities. The proposed signcryption technique is based on Elliptic curve and is securely provable under the Computational Diffie-Hellman model.

The authors in [67] considered network users as a set of predefined groups, where each user is assigned to a single group. The groups are constructed in such a way users having the same access privileges belong to the same group. The main proposal consists on "privacy-preserving" ring signature scheme considering the members of each group as the nodes forming the ring. This technique allows IoT devices (signature verifiers) to grant access to legitimate users (signers) without disclosing the identity of each user neither from sensor data owner nor from other users. The only revealed information about queries is the group (gid) containing the signer's group ID from which the query is originated without knowing exactly which signer. The experiments were performed in real Imote2 platform running TinyOS [2] demonstrate the efficiency and feasibility of the scheme in real WSN and IoT applications.

In [64], authors claimed that, actually, existing access control mechanisms like RBAC (Role Based Access control), MAC (Mandatory Access control) are not anymore scalable, difficult to manage and don't fit well with distributed environments like Internet of Things, and hence the need for a new effective access control mechanism is unavoidable. The authors proposed a new access

[2]embedded, component-based operating system : http ://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Documentation_Wiki

control mechanism called capability-based access control (CapBAC), which can overcome the actual issues in terms of scalability and manageability raised with the existing access mechanisms. The idea behind the concept is the usage of capability based authority tokens which are unfalsifiable and easy to communicate and grant seamlessly the access to IoT resources and process.

2.4.1.3 Identity Based Encryption (IBE)

The main issue of transitional public key cryptosystems is that they are not scalable enough. Indeed, they strongly depend on the authority that issues certificates for each user in the system which is required in order to deal with spoofing and identity usurpation. Therefore, certificates raise the complexity of the system. In order to overcome the scalability and the complexity issues, Identity Based Encryption tools have been proposed by introducing a new concept that consists to use unforgeable string related to the user identity (such as user's phone number, email address, etc.) as public key to encrypt data and thereby eliminate the need for certificates. Although their scalability and efficiency, IBE techniques are not very suitable for IoT because they are expensive and incur heavy resource consumption. In the literature, some research works have been investigated to design new, efficient, and lightweight IBE schemes that could support constrained devices.

Using Elliptic Curve Cryptosystems, bilinear maps and hash functions, Chen [36] proposed a new lightweight Identity Based Encryption scheme to secure communications between devices based RFID tags. The main advantage of the scheme is its simplicity and its ability to reduce substantially the computation overhead. However, the authors did not provide any discussion about the security of the scheme.

Fagen et al. [86] addressed the access control problem in WSN in the context of IoT where internet hosts query WSN to get sensor information. The main contribution consists of heterogeneous signcryption (HSC) technique based on two mechanisms : (1) certificateless cryptography (non usage of certificates) that belongs to internet hosts ; and (2) IBC cryptographic technique that belongs to WSN environment. As singcryption technique, the proposed scheme ensures both authenticity and confidentiality with less computation. Moreover, it is useful to control the access between heterogeneous environments.

In [78], a signcryption scheme has specially designed for WSNs in the context of Internet of Things. The scheme is based on elliptic curves and is secure under the Diffie-Hellman computation hypothesis. Nevertheless, this scheme is applied only in contexts where the verifiers are always powerful nodes that have enough

computational resources and it's consequently very heavy for IoT devices.

Fuzzy identity-based Encryption (FIBE) is considered as an enhancement of IBE with introducing error-tolerance property. The main idea behind FIBE is to give the users, having at least k among n attributes, the possibility to decrypt the cipher-text encrypted under the hole attributes (n) [116]. In [99], the authors designed FIBE scheme based on bilinear maps which is securely provable in the full model. Performance analysis demonstrated the applicability of this scheme in IoT.

2.4.1.4 Attribute Based Encryption (ABE)

The concept of Attribute Based Encryption (ABE) has been introduced, first, by Sahai and Waters in Advances in Cryptology EUROCRYPT 2005 [116] as an enhancement of Fuzzy Based Identity Encryption [19, 41]. ABE introduces an expressive way to control the access to private data using policy access structure that defines relationships between a set of attributes [3] used to encrypt data. In ABE system, Key Generation Server (KGS) generates for each legitimate user a private key based on its attributes, and also a public key used to encrypt data based on predefined policy. A legitimate user is able to decrypt data only if it holds the sufficient attributes that satisfy the policy.

- **Key Policy ABE (KP-ABE)** : In this scheme, the data owner defines an access structure A and encrypts data based on a set of attributes I. A user which wants to decrypt the cipher-text must holds the attributes that satisfy the access structure A to be able to derive the private key that decrypts the cipher-text [58] (see figure 2.2).

- **Cipher-text Policy ABE (CP-ABE)** : In this scheme, the encryption is based on the access structure A. A legitimate user is a user who holds a set of sufficient attributes I that satisfies the access structure (policy A) attached to the ciphertext [15] (see figure 2.3).

Attribute-Based Encryption is considered as a promising scheme for many applications such Cloud computing, multicast communication, M2M, etc. Particularly, in Internet of Things' applications, we need often efficient mechanisms that ensure fine-grained access control to IoT data based on the roles of the users in the IoT systems. We can take as an example, the Healthcare applications where EHRs (Electronic Healthcare Records) related to patients are only accessed by physicians and nurses based on their roles in the hospital institution. This is achieved by ABE thanks to

[3] properties related to the users in the system, for example : PhD student can be considered as an attribute

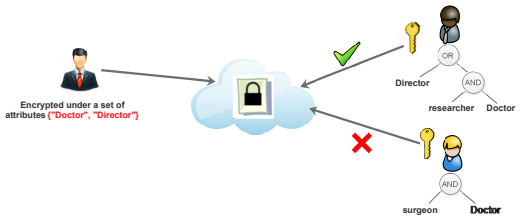


Figure 2.2 – Key Policy ABE (KP-ABE).

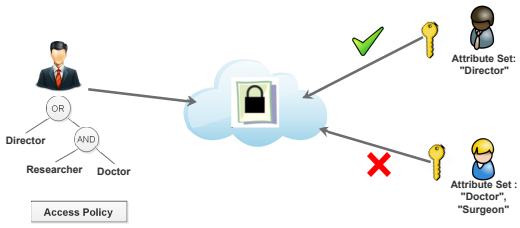


Figure 2.3 – Ciphertext Policy ABE (CP-ABE).

its scalability, efficiency and its fine-grained capability. However, the complexity and the high overhead induced by the cryptographic operations in ABE schemes make its application in resource-constrained devices very difficult. These drawbacks are serious problems to overcome in order to adapt ABE in IoT applications.

In [133], the authors proposed a distributed lightweight ABE solution based on CP-ABE scheme. The solution takes advantage of IoT heterogeneous nature which consists to delegate the most costly cryptographic operations (exponentiation) to more powerful nodes. However the solution consumes a lot of bandwidth, as objects exchange cryptographic information in order to accomplish the encryption process. The cost due to message exchanges is very considerable in the radio field and must not be neglected.

On the other hand, Nouha et al. [110] proposed ABE based solution that ensures a tradeoff between computation and storage capacity of constrained devices. They use a pre-computation technique in order to reduce computation cost. This technique consists to pre-compute and store in a lookup table a set of pairs obtained generally with expensive cryptographic operations done on elliptic curves and pairing group settings. This information is used later to carry out cryptographic operations with very low computations. The main drawback of this solution is that the look-up table must be as bigger as possible in order to overcome dictionary attacks.

Shucheng et al. [141] proposed a distributed fine-grained access control scheme based on KP-ABE for wireless sensor networks called FDAC. The authors consider sensor node properties such as its geographic location, the type of sensor's data, time, its owner, etc. as attributes to define access policies in order to control the access of users to sensor data encrypted under the defined attributes. The main properties of the scheme are that sensor nodes may change seamlessly their attributes as well as its capacity to support data aggregation. The feasibility of the solution is evaluated with real experiments under iMote2 platform.

In [60], the authors addressed the key storage in CP-ABE in IoT context. Mostly the encryption key is constant-size (does not depend on the number of attributes). The proposed solution is provably secure in the selective security model. However,

this solution generates big ciphertexts which create a big problem for IoT devices that are highly constrained in terms of bandwidth and storage.

In contrast, Müller et al. in [101] proposed a multi-distributed-authorities based ABE solution for IoT environments. The solution is kind of an adaptation of ABE to support a distributed access policy among a set of authorities, where the generation of secret keys from the attributes is handled with the collaboration of several authorities. Each authority generates a sub-key taking in consideration its maintained access policy.

The most existing ABE schemes are based on expensive bilinear pairing operations, which are, in general, not suitable for constrained devices in IoT. For this reason, some researches have been conducted in order to propose a lightweight non-pairing ABE schemes. The contribution in [140] is new lightweight ECC-Based ABE scheme that consists on replacing pairing operations by point scalar multiplication on elliptic curves. Under the ECDDH assumption, the authors proposed a security proof of the scheme in the attribute based selective-set model.

In [127], the authors tackled the problem of integrity and authentication in IoT with an expressive attribute based signature (ABS) scheme. The scheme preserves the privacy of signers and don't leak any information about users. However the scheme is still heavy computational for both the signer and the signature's checker as it uses a lot of pairing operations and exponential computations. Thus the scheme is not quite suitable for IoT constrained devices.

In the context of communication based groups in IoT, the authors in [131] proposed to combine Attribute Based Encryption schemes and Publish Subcribe based MQTT messaging architecture in order to ensure data encryption as well as the security requirements in group communications, namely forward and backward secrecy. the proposed solution ensures a flexible keys updating in case of join/leave procedures in MQTT architecture.

In order to study the adaptability and feasibility of applying ABE schemes, namely CP-ABE and KP-ABE, on smart-phone and IoT devices, Ambrosin et al. [9] have conducted intensive experiments in diverse mobile platforms (smart-phones, laptops, etc.) based on different OS (Android, Windows). The obtained results demonstrate the feasibility of ABE in smart-phones and similarly for IoT devices. On the other hand, authors in [1] proposed a lightweight hardware implementation of CP-ABE scheme on Field Programmable Gate Array (FPGA). As a proof of concept, CP-ABE based 16 bits key size was tested with different setups. It's worth noting that with the conducted experiments, the scheme is quiet less power consuming.

2.4.2 Privacy solutions

Actually preserving privacy in IoT is mandatory as data issued by smart objects are very sensitives and inherently related to real life's individuals. The main goal of privacy techniques is to ensure the following requirements :

- **Anonymity** : Property ensuring that a third entity is unable to identify person's identity among other identities in the system.
- **Unlinkability** : Impossibility to cover the persons' identity from the information they produce.
- **Untraceability** : Difficulty to track actions and information issued from the behavior of an entity in the system.

The privacy solutions aim to protect sensitive data and also provide mechanisms that hide users' identities in such way the intruders cannot know about their behaviors. In the following, we discuss some solutions proposed in the literature that address the privacy of data and user's behaviors in Internet of Things.

2.4.2.1 Data privacy

Data tagging is one of the most known techniques, mainly used to ensure privacy of data flows. The idea behind this concept is to associate additional labels called tags, to data flows in order to allow trusted computing entities to reason about flows of private data and thus hide identities of individuals who hold or control data [23]. Nevertheless, tagging mechanisms might cause a challenge for constrained devices as tags' sizes raise according to the size of data and also generate additional expensive computations. In [52], authors demonstrated the applicability of tagging mechanism for constrained programmable micro-controller (PIC) by providing lightweight code templates dedicated to resource-constrained devices in order to add tags to data flows.

ZKP (Zero Knowledge Proof) is a powerful mechanism largely used to ensure the privacy of users' identities. The idea behind ZKP is to allow to one party (prover) to demonstrate to another party (verifier) some property by proving its possessing of some information without disclosing it [30]. This concept is very useful to develop security protocols while preserving the privacy of users' data and properties. In contrast, Ioannis et al. [30] proposed an evaluation of some ZKP protocols based on the Discrete Logarithm Problem on elliptic curves (ECC) for resource-constrained devices. The obtained results demonstrate that using ECC (with 1024 key's length) comparing to RSA provide less execution time and less memory with the same

level of security. Importantly, with small message sizes, the energy related to the communication is minimized. However, beyond some threshold, the ZKP protocols became more overloaded which is due to the fragmentation of messages.

K-anonymity model is another potential approach to protect the privacy of data in Internet of Things' applications. Considering the context of a set of homogenous data stored in a table where each column represents a record of these data which is owned by some specific user. The K-anonymity models aim to protect each record in the table and make it indistinguishable from at least $k-1$ records in the same table by hiding the sensitive information about its owner [129]. These sensitive information may be the ages, the phone numbers, the addresses, etc. This model is largely adopted in big data and cloud applications to protect the privacy of data streams issued by different users. Particularly, in IoT applications, there are also some attempts to adopt k-anonymity models [106, 70, 72]. In [70], authors proposed context aware k-anonymity model with conjunction to other privacy protection mechanisms to protect data issued from sensor nodes in WSN. Huo-wang et al. [72] investigated a clustering technique to propose a k-anonymity model to hide sensitive data about the locations of sensor nodes in IoT context. The idea behind the solution is to gather the data related to the sensor nodes located in different regions in different classes to make them indistinguishable.

2.4.2.2 Privacy of users' behaviors

In Internet of Things, users and objects perform actions in the systems such as access to sensor data, control remote actuators, etc. Therefore, it's mandatory that their behaviors should be protected against malicious intruders. In what follows, we discuss some works that aim to protect the privacy of users' behaviors.

In [145], the main contribution is a privacy-aware access control protocol called DP^2AC in Wireless Sensor Networks based on RSA blind signature mechanism. In this solution, the owner of data signs the hash of an arbitrary integer m generated by some user x which forms an access token. So, the user x uses the token $< m, (\sigma(m) = (h(m))^d$, where $h(m)$ and $\sigma(m)$ are respectively the hash of the integer m and the signature of the message m using the owner's private key $d >$ to prove its capability to access data. The verifier which holds the data, checks if $h(m) = \sigma(m)^e = h(m)^{ed}$ to control the access of the user x without necessarily leaking any information about its identity. The protocol has the advantage to be simple and efficient. However, it does not ensure fine grained access as all users have the same privileges to access sensor data.

According to [43], decentralized approaches can enhance privacy more than

centralized approaches as they do not rely to any central entity which might track data flows and thus can probably deduce sensitive information of individuals from the exchanged data. In contrast, Alcaide et al. [7] proposed a fully decentralized authentication protocol that preserves the privacy of users. Besides, users in the system are authenticated by data collectors in a flexible manner based on Anonymous Access Credentials which are unlinkable.

In [124], authors proposed a capability-based access control mechanism by introducing lightweight tokens to access CoAP [4] (Constrained Application Protocol) IoT resources while preserving the privacy of data over end-to-end communications. The token is exchanged in GET CoAP requests and contains the necessary information to control the access to device resources such as request Id, subject Id, Device Id, Issuer Id, Issued time, ESDSA signature, etc.

Recently, Samet et al. [132] investigated a new mechanism based on Data Obfuscation schemes in order to preserve the privacy of the exchanged metrics in smart grid AMI networks. The idea of data obfuscation is that each gateway creates and distributes obfuscated values to smart meters. Then, smart meters slightly disturb the sensed data based on obfuscated values and transmit them again to the utility control center, which can do estimation about the received data containing basically the electricity consumption of smart meters. This solution is less-computational which makes it applicable in resource-constrained devices. However, it generates a lot of overhead in the AMI network infrastructure.

2.4.3 Availability solutions

In IoT, the availability of the system is one of the most important security services needs to be protected against malicious attacks (like DoS/DDoS) or unintentional failures. Very often, the damages caused by the violation of the availability are tremendous which can be economical losses (in manufacturing systems) or safety damages (in transportation systems). Furthermore, ensuring the availability is a very challenging task because attackers exploit all types of vulnerabilities in different levels (network, software design, cryptographic algorithms, etc.) to break the system. For example, in October 21, 2016, one of the largest American computer companies providing DNS service, DYN (Dyn Managed DNS) was attacked by hackers who used a type of DDos attack exploiting IoT devices. During this attack, many known sites were blocked for 10 hours, such as Amazon, BBC, PayPal, etc. The attackers take advantage of comprised IoT devices (such as surveillance cameras) infected with the malicious software named Mirai to relay massive packet streams.

[4] Considered as an alternative of HTTP in IoT environments

2.4.3.1 IoT DoS/DDoS countermeasure approaches

IP Traceback methods are powerful mechanisms largely adopted in IP based networks such as Internet to detect DoS and IP flooding attacks in real-time. These methods focus mainly to enhance the security of IP based lightweight protocols basically designed as adaptations of the traditional TCP/IP protocols in the Internet of Things. DTLS [5] (Datagram Transport Layer Security), 6LoWPAN [6] (IPv6 Low power Wireless Personal Area Networks), RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks)) are just examples among other protocols widely adopted in the world of IoT which provide confidentiality and integrity of end-to-end exchanged information between IoT devices [117]. However, these protocols are not initially designed to deal with the most common IP based DoS/DDoS attacks. Many solutions have been investigated to enhance DTLS based transport layer and RPL based 6LoWPAN routing protocol in order to turn them more robust and secure against DoS attacks. In these solutions, IP routers and IoT gateways inspect and analyze packets in order to identify eventual malicious behaviors and take actions accordingly.

Regarding TCP/IP transport layer, the contribution in [95] consists on an enhancement of the DTLS protocol in order to mitigate DoS/DDoS against IoT devices and gateways. The enhancement is done by extending the process of the DTLS handshake with an additional cookie exchange technique where the server, before resource reservation, sends an authentication cookie's code to the client through *HelloVerifyRequest* message. This later, upon receiving the message, could authenticate the server and sends again to the server a new authentication cookie encapsulated in *Hello* message. To prevent IP spoofing attacks during the handshake phase, a mutual authentication step is done between the client and the server through a Gateway.

On the other hand, in TCP/IP network layer and specifically in the routing level, many security enhancements of RPL and 6LoWPAN based IoT architectures are proposed. In contrast, Kasinathan et al. [76] proposed an architecture to protect IoT based 6LoWPAN devices against DoS attacks as well as jamming and tampering attacks in the context of the European project called *ebbits* [7]. The main contribution is twofold : first, the design of Intrusion detection manager that is charged to protect constrained devices against DoS attacks. Second, the design of the IDS (Intrusion Detection System), operating in promiscuous mode, that is responsible to monitor 6LoWPAN packets and raises alerts in case of any misbehavior. The solution is

[5] An alternative standard of TLS, it is a UDP-based protocol which is less network overloaded
[6] Lightweight based IPv6 protocol to address IoT devices
[7] https ://www.fit.fraunhofer.de/en/fb/ucc/projects/ebbits.html

based on Suricata IDS [8] that uses the signature based detection technique. Likewise, Hummen et al. [71] investigated the attacks related to 6LoWPAN fragmentation mechanism, basically two attacks were studied : fragment duplication attacks and buffer reservation attacks which both of them aim to prevent the availability of the IoT devices. They proposed a mitigation approaches that counter to these attacks. In the routing level, Rghioui et al. [113] surveyed the potential DoS attacks that could disturb RPL and 6LoWPAN IoT protocols. They proposed also mitigation solutions of theses attacks based on IDS approach. Likewise, recently, [120] focused on intrusion detection in RPL based 6LoWPAN. They proposed some extensions of the protocol by exploiting the ETX (Expected Transmissions) metric as a mechanism to prevent malicious nodes.

Recently, Cusack et al. [44] discussed and compared many IP traceback approaches based on some metrics such as storage requirements, processing overhead, bandwidth overhead, scalability, etc.

Artificial intelligence techniques such as Artificial Neural Networks (ANN) are considered as one of the most powerful techniques used to design security solutions as IDS for example IDS. In [48], authors investigated the application of ANN to detect DoS attacks in IoT. They evaluated two kinds of ANNs, namely : Multilayer Perceptron with Limited Weights and Multilayer Perceptron with normal weights in order to verify which one is more adequate as an IDS in IoT. It's worth noting that both of ANN techniques reduce false positive detection under training process. However, they consume a lot of memory which makes them not quite suitable for constrained IoT devices.

Others researchers [93] investigated the possibility of applying Cumulative Sum (CUMSUM) algorithm in order to detect DDoS attacks in the context of IoT. The main aim of CUSUM algorithm is to detect real time changes in statistic process issued from data streams. The DDoS detection is done by analyzing the network traffic and computing statistics about it. The algorithm handles, continuously, these statistics to eventually detect changes which are related to any misbehavior in the network traffic. A trade-off between False Positive Rate and Detection Rate is also investigated by playing on CUMSUM algorithm parameters.

Other works have tackled with DoS attacks related to routing protocols in WSN and Internet of Things. Indeed, security of routing protocols is a fundamental field of research as many IoT applications use in general wireless mesh or ad-hoc network infrastructures to exchange data in real time. It is the case, for example, of AMI in smart grids and ad-hoc infrastructures in Vehicular Networks. In [6], authors interested in healthcare applications. They studied several mesh routing protocols

[8] https ://suricata-ids.org/

in order to choose the most robust and secure protocol against DoS attacks. They focused on one type of DoS attacks that aims to divert the routing protocol behavior from its initial function. For example, routing attacks that force some network nodes to reroute data to inappropriate destination. Simulation results confirmed that PASER protocol is the most suitable for Healthcare applications and it is resilient against Hello Flooding attacks.

2.5 New emerging security solutions for Internet of Things

The IoT promises to connect everything together anywhere and everywhere. All devices must interact efficiently with each other in a secure, scalable and reliable ways. Actually with the current centralized architecture, it could be difficult and challenging to deal with scalability in huge IoT networks. This issue may be solved by adopting a new approach of security emerged away from the current centralized model. New emerging approaches deal very efficiently with scalability, interoperability and compatibility issues. Hereafter, we discuss two emerging technologies which are being adopted as approaches to ensure security in IoT environments and deal very efficiently with scalability issues.

2.5.1 Software Defined Networking based solutions

The Software Defined Networking (SDN) is a new paradigm that has revolutionized the world of networking, thanks to the programmability and the intelligence it has introduced into the network. The main idea behind this concept, which began in 2011, is to separate the network control plan and the data plan. Using this paradigm, we can do centralized control and configuration of networks as well as dynamic management of network traffic. In SDN architectures, devices (routers, switches, gateways and IoT devices in general) do not make control decisions like forwarding tables and ACL rules [68]. Instead of that, they learn these rules from central component called SDN controller, which is managed to take all decisions in the network using protocols like Openflow. Devices in SDN architecture handle packets based on flow tables dictated by SDN controller.

SDN is an efficient solution to meet some challenges in IoT environments where most of devices have limited network resources. As a result, SDN deployment in conjunction with NFV (Network function Visualization) can optimize efficiently the resource allocation in IoT devices. Therefore, it introduces some many opportunities in order to overcome some challenges of reliability, security, scalability and QoS in

IoT applications in more efficient and flexible way [68]. Hereafter, we discuss some SDN based solutions that address the security issues in IoT.

The main contributions in [53] are twofold. First, the authors proposed a new multi-domains SDN based IoT architecture that supports both networks with or without infrastructure. Second, they designed a distributed security model to manage security policies between multiple SDN domains. In order to address the conflict issues that appear from the enforcement of the security policies on the several domains, the solution takes advantage of the grid of security paradigm that aims to solve security heterogeneity issues. So, each SDN controller is charged to push security policies inside its domain and coordinates with other SDN controllers outside the domain.

In [25], authors presented an openflow [9] based SDN architecture for IoT devices. The proposed architecture includes IoT gateways that are managed to identify attacks and anomalies in order to determine which devices are acting maliciously and which are the compromised nodes in the network. To do that, each gateway analyzes the network traffic dynamically. So, upon the detection of an anomaly or an abnormal behavior such as generated periodic flows (DoS attacks), the gateway applies an appropriate mitigation action (block, forward, apply QoS) depending on the anomaly.

The work in [134] considers the heterogeneous IoT infrastructure as a couple of connected clusters or segments, where in each segment, there are IoT nodes that support Openflow protocol and have sufficient resources in terms of computation and energy. These IoT nodes act as SDN gateways, and are charged to : 1) authenticate nodes in the same segment, and 2) enforce adequate security rules using Openflow protocol. The SDN gateways exchange between them the security rules in order to establish secure, end-to-end connections between IoT nodes in different segments.

Gonzalez et al. [57] present a new SDN framework to overcome the different kind of attacks in IoT environments. The proposed framework is based on the architecture called SDCSN (Software Defined Cluster Sensor Networks) proposed in [108] that consists of multi-domain SDN architecture. Each SDN domain (cluster) has a SDNCH (SDN Cluster Head) that is charged to monitor and secure SDN domain and prevent outside and inside attacks. The mechanisms employed in order to implement an SDN firewall are based on analysis of flow entries on the application level.

Other works investigated SDN approaches as solutions to prevent against malicious attacks such as DoS and also implement efficient IDS. In this contrast, Lee et al. [84] tackled the problem of availability in IoT based gateway environments

[9]The most known SDN protocols, proposed by ONF

for which they proposed an SDN based solution to prevent IoT gateways from DoS attacks and evaluate the main impacts. In order to evaluate the impact of different kinds of Dos attacks on IoT gateways, the software solution was implemented using Raspeberry Pi2 platform, OpenWRT operating system as a wireless Router and Opendaylight as an SDN controller.

Aydeger et al. [11] proposed a SDN-based MTD (moving target defense) mechanism to defend against specific types of DDoS attacks called Crossfire. The SDN-based mitigation approach consists to enhance the packet forwarding process in such away routes containing congested links are avoided.

2.5.1.1 Main challenges of SDN in terms of security in IoT

During these last years, there are a lot of discussions about SDN and its benefits in the industry of networking. However as new emerging technology, SDN is not enough mature to address the security issues in Internet of Things. Hereafter, we discuss some potential challenges that are still difficult to overcome with SDN based approaches :

- In general SDN based security solutions are designed to operate in centralized architectures. Therefore, the centralized SDN controllers emerge as a potential single points of attacks that should be protected against attacks such as DDoS for example.
- Southbound interface between SDN controller and data plan is vulnerable to threats that could degrade the performance of the network. As example, Openflow protocol suffers from integrity as mentioned in [22].
- SDN approaches suffer from scalability issues. Indeed, SDN controllers can not deal efficiently with the large number of IoT devices in the underlying data plan network.
- In highly dynamic environments like vehicular networks, where network topology changes frequently and a lot of messages are exchanged between vehicles, centralized SDN approaches is still limited. Indeed, gathering all these changes from the underlying network to enforce security policies and configurations using SDN approaches takes a lot of time.

2.5.2 Blockchain based solutions

Blockchain is a new effective technology that has revolutionized the world of cryptocurrency these last years. this technology has received a great attention

by researchers in various fields. Until now, its application has recognized a great success in financial applications and smart contracts, but some researchers claim that it's worth investigating to think out of the box and try to figure out other application domains than cryptocurrency that this effective technology can improve considerably as Internet of things and security domains. In this section, we introduce the technology blockchain, its benefits as well as some security based solutions in the field of IoT.

2.5.2.1 Review on Blockchain

Blockchain consists mainly of a secure distributed database (a.k.a public ledger) containing all the transactions that have been made by all the participating entities. The main aim of this database is to allow heterogeneous nodes to communicate and exchange assets between each other in a completely distributed and secure way, without relying to any trusted central entity. Basically, each node in the blockchain does not trust any other node, however, it trusts the whole blockchain network. In the blockchain, each node holds a pair of cryptographic keys (public and private key) that allows to generate transactions and interact with other nodes in the network. In addition, these transactions are immutable. Indeed, it is hard to falsify any transaction once added to the blockchain. In the distributed (P2P) architecture, blockchain network, it is mandatory that the whole nodes reach a consensus state about the validation of each transaction. We note that in order to insert a transaction in the blockchain, the majority of validation nodes called miners or validators need to validate it. In what follows, we explain the different steps from transaction generation until the validation of the transaction in the blockchain [21] :

- When an entity A wants to exchange some asset (crytomoney in case of bitcoin) with another entity B, it generates first a transaction containing the asset and sign the transansaction with its private key. Then, it broadcasts the transaction to all the peers in the blockchain.

- Each validator node (known also as miner in bitcoin) periodically (10 minutes in the case of bitcoin) gather a set of transactions in one single block. It must verify the correctness of each transaction before adding it into the block.

- Each validator must locally execute the adopted consensus technique to validate and broadcast the block to all the peers in the network.

- After verifying the format of the transactions, the other nodes verify the correctness of the validated block received by each validator node. If the the

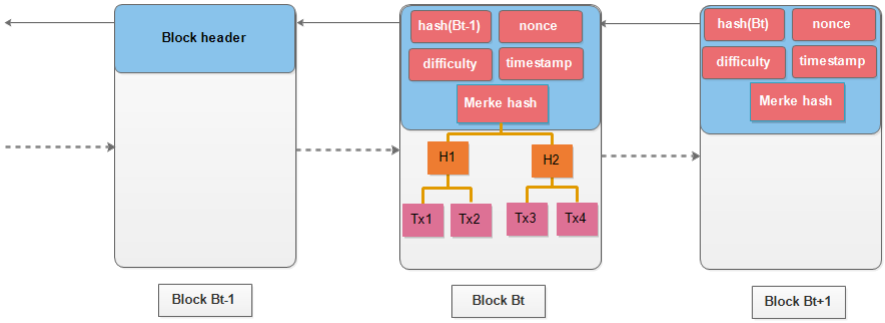


Figure 2.4 – Blockchain structure in bitcoin system

block is correct and the consenus is reached, all the peer add this block to the blockchain. Otherwise, the block is discarded.

Basicaly, we have two types of blockchains : public blockchain (permissionless) and private blockchain (permissioned). In public blockchains, any node in the network is able to perform and participate in the consensus process to validate transactions. On the other hand, in private blockchains, the consensus processus is performed only by a subset of nodes that form a consortium.

2.5.2.2 Consensus mecanisms

Depending on the type of the blockchain, we distinguish several consensus protocols [26]. In the following, we highlight three most known consensus protocols :

Proof of Work (PoW) : the process of validation in PoW is done by a subset of powerful nodes called the miners that must solve a heavy mathematical puzzle. This puzzle consists on finding a *nonce* value in such a way, when it is contactenated with the hash of the previous block $H(B_{t-1})$, the merkle hash of the current block $MerkleHash$ and the *timestamp*, the obtained hash must start by a certain number of zeros depending on the *difficulty* of mining. In other words, the nonce value must verify $h(h(B_{t-1})||MerkeHash||timestamp||nonce) < difficulty$. Once the block is validated, it is simple for each node in the blockchain to verify whether the validation of the block is done correctly. In practice, it is impossible to to falsify or update one block yet validated without redoing the same heavy validation process for this block and all its subsequent blocks in the blockchain.

Proof of Stack : in proof of stack concensus, each node must prove its possession of some stack (coins in the case of cryptocurrency). This stack value is used to choose the validator for each block, so nodes that hold the highest amount of stack are more likely to be selected. The concept of stack can be generalized to represent any kind of assests such nodes' trust level of nodes.

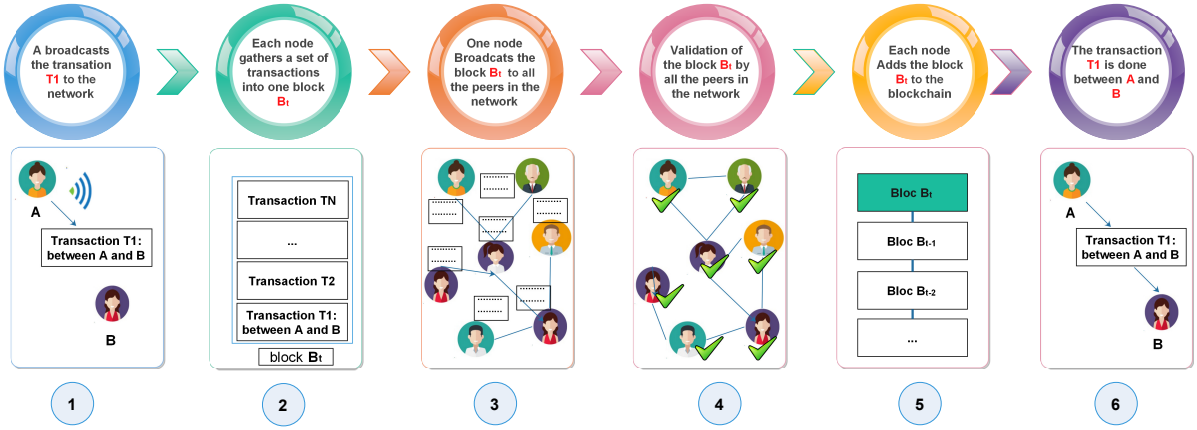


Figure 2.5 – Blockchain : steps of transactions' validation process.

Practical Byzantine Fault Tolerance (PBFT) : PBFT was originally proposed to solve the problem of Byzantine generals [28]. This protocol can tolerate Byzantine faults up to $1/3$ faults. Indeed, the validation of each block is done only if each node receives at least $2/3$ of "*block commit*" messages from the other validator nodes. This protocol is very efficient especially in private blockchains as there is no need to perform heavy compulations during validation process and it is fairly distributed compared to PoS protocol which tends to be centralized.

In figure 2.5, we illustrate the different steps evolved in the blockchain transaction treatment process. We illustrate also in figure 2.4 the general structure of a block (the case of bitcoin system).

New emerging IoT based applications will be taking advantage of secure and private transaction messaging, decentralization of communications, privacy by design which are all very important features for IoT [12]. As IoT continues to grow, sensors and devices are becoming more common places to communicate information like location, temperature and other properties. Often, this information needs to be shared between different entities and exploited for big data analysis and also for monitoring purposes in some critical applications. Blockchain can help to create tamper-resistant record which allows all participating smart objects to access the same data in more consistent and safer way. In addition to data flow management, blockchain consists of an efficient way for automating business and creating smart contacts among smart devices without referring to central entities. We mean by smart contracts, all kinds of digital rules forming the terms of a contact [39]. Concretely, a smart contract consists of a computer program that is automatically executed by smart objects, and defines a set of rules and conditions based on the terms of the contract. Blockchain could help to ensure the smooth running of the contracts in a distributed way. We already have some examples of applications

that are non financial such as global identity registry systems (namecoin [89], Blockstack [8] among others), insurance applications [94], online voting [114], supply chain provenance [77], decentralized peer to peer storage platform (storj [138]) etc. Moreover, recently, in the literature, some blockchain based solutions have been proposed to solve some security and privacy issues in IoT. We discuss some of these solution in the following sections.

2.5.2.3 Benefits of blockchain in IoT

Hereafter, some added values that blockchain technology can bring to IoT and security domains [43] :

- **Decentralization** : Because of the decentralized architecture of IoT, blockchain is most suitable as a security solution in IoT. The decentralized architecture of blockchain makes security solutions most scalable and can solve the problem of single point of failure and becomes more robust to DoS attacks.
- **Pseudononymity** : The nodes in blockchain are identified by their public keys (or the hash of public keys). These pseudonyms don't link any information about the identity of the participating nodes.
- **Security of transactions** : Each transaction, before being sent to blockchain network, is signed by the node and must be verified and validated by miners. After the validation, it's practically impossible to forge or modify transactions already saved in the blockchain. This provides a proof of traceable events in the system.

2.5.2.4 Secure IoT transactions

Using blockchains, some works were focused on secure IoT transactions in decentralized architectures. We discuss hereafter some of those proposals.

The first IoT platform based blockchain solution was developed by IBM in 2013. This platform is called ADEPT (Autonomous Decentralized Peer-To-Peer Telemetry) [10] which consists of proof of concept of a decentralized and secure IoT platform based on Ethereum protocol which is a seamless solution to deal with devices contracting and transactions in a most scalable way. So IoT devices can define and set autonomously their own roles, responsibilities and permissions in the whole IoT ecosystem and also can do transactions and complex negotiations between themselves.

In [54], authors proposed an HTTPS protocol for IoT devices by eliminating the intermediary devices (like mobiles) to create secure HTTPS channel (classical

solution). The session key is generated by PBKDF2 (Password-Based Key Derivation Function 2) algorithm and the IoT transactions are stored in a blockchain maintained between several devices. This solution could be enhanced by introducing a priority among transactions.

Recently, Kamanashis et al. [17] proposed a multi-layer security architecture for smart cities that integrates the blockchain as a distributed database layer to share and store heterogeneous IoT data related to the smart city environment such as traffic, temperature, location, humidity, etc. The data storage aims to share these data in a secure way among different smart cities' components. The architecture is designed to deal with scalability and reliability issues that are very challenging in smart cities environments.

In [79], the authors proposed a solution to manage SSH public keys based on blockchain and collective signing authorities. They mainly addressed the key management problem presented between IoT devices to access to distant services. The main idea of the solution consists on adding a new block in the blockchain containing the SSH public keys whenever a key is added, rotated or updated.

Recently, Hardjono et al. [65] proposed a solution for identifying the manufacturing provenance of IoT devices while preserving the identities of the users using blockchain and cloud computing. The authors proposed a platform solution based on EPID (Enhanced Privacy Identity protocol) of Intel, a standard used for identification of IoT devices. The platform supports also device owners data selling in order to incentive IoT devices' owners to share their IoT data.

2.5.2.5 Data Sharing

In several IoT applications, a lot of data is exchanged between objects and with other entities. For that, it's very important to deal with this data and propose security solutions to share it with others. Moreover, privacy is a great issue that should be considered while addressing sharing data problem in IoT. According to [43], to ensure privacy in IoT systems, it is recommended to use peer-to-peer architecture, and specifically the blockchain technology. In addition, as all the operations handled on IoT data are controlled by the blockchain, it is easy to detect any abuse in data [43]. Blockchain might serve as a tool to deal with all these aspects. In this trend, several works have been proposed.

In [66], authors proposed a decentralized solution for sharing data in IoT environment which consists of a distributed data storage system. The proposed system uses blockchain to maintain data access control and data storage model. The main features of this system are : 1) separation of data store and data management.

This later is ensured using blockchains, 2) a decentralized access control and 3) a scalable messaging based on Publish-subscribe model to query data.

In [142], authors proposed a healthcare data sharing based blockchain architecture. The proposed architecture includes three layers to manage the access to private ERH (Electronic Medical Record) related to patients. The first layer consists of the different users that are potentially interested to access the patient's data. To do that, a user sends different requests to the Healthcare Data Gateways (second layer) in charge of the management of the access control to the data stored in the blockchain, used as a storage data layer which is based on cloud storage. Besides the immutability property of blockchain, the ERH are encrypted and signed to ensure the confidentiality, the authenticity and the integrity of data.

2.5.2.6 Main challenges of blockchain in IoT

Despite the blockchain's benefits mentioned above, it is still some challenges to be solved in order to adapt the blockchain technology in IoT. We enumerate the following challenges :

- **Computation and storage issues** : As most of IoT devices have limited capabilities in terms of computation and storage resources, the blockchain needs to be customized before its application as security solution in IoT. To address the problem of adaptability, one solution may consist to add a new application level that hides the details of blockchain implementation, namely the PoW (Proof of Work) [43]. This solution allows the resource-constrained IoT devices to involve in the system without computing the PoW.

- **Time latency** : In bitcoin blockchain, the validation of transactions takes about 10 minutes, which creates a problem for real time applications.

- **Scalability issues** : Although the remarkable success of bitcoin blockchain and the number of users that rises year after year, blockchain technology is still non scalable solution in IoT environments.

- **Bandwidth consumption** : As IoT devices generate a lot of transactions, this includes an important problem if it is necessary to validate each of those transactions that consume a lot of bandwidth.

- **The anonymity** : Actually, blockchain does't ensure a fully anonymous transactions. Indeed, the peers are identified by pseudonyms that can be tracked but they are still unlikable (impossibility of extracting identity of the person from its pseudonym) [43].

2.6 Summary and discussion

In Table 2.2, we present a comparison of security solutions implemented in IoT based on criteria that we described previously in section 2.2.

At first glance, we notice that security solutions implemented in IoT and proposed in the literature are not all efficient in all the aspects and don't fulfill all the security requirements. Indeed, traditional solutions based on cryptographic techniques which are adapted for IoT applications are, generally, efficient in terms of storage and computation. However, they are limited in terms of scalability and heterogeneity. In the other hand, blockchain based solutions deal very well with scalability and heterogeneity issues thanks to the distributed architecture offered by blockchain technology. Nevertheless, the most drawbacks of blockchain technology are the energy consumption and latency caused by the proof of work mechanism to validate transactions which is serious problem in the case of real-time and energy constrained IoT applications. In the other side, SDN approaches optimize very efficiently computation costs, energy consumption and network resources since all control tasks are dedicated to high-performance servers (called SDN controllers) which discharge constrained IoT devices from greedy operations (including the execution of cryptographic tasks). Obviously, as a centralized approach, SDN doesn't deal efficiently with scalability issues in IoT.

2.7 Conclusion

In this chapter, we surveyed security solutions proposed for IoT applications. We first categorized the different IoT applications by identifying their security requirements and their inherent challenges. Then, we discussed the IoT solutions dealing with confidentiality, privacy and availability which are based on traditional cryptographic solutions. We also reviewed some emerging technologies such as Blockchain and Software Defined Networking which are considered as efficient mechanisms to deal with scalability issues in IoT. Finally, we discussed some security solutions that take care of the context in which IoT applications involve and also the different impacts of security issues on the safety of systems and some countermeasures. Comprehensive comparison of the different approaches was provided based on some criteria, we investigated also some analysis of which techniques are suitable for each kind of IoT application. Despite the efforts that have been spent to deal with the various challenges to which Internet of things face, it is still a lot of open issues to be addressed such as scalability and dynamism issues, especially because IoT is becoming an Internet of Everything where humans, data, processes and objects are

Tableau 2.2 – Comparison of some IoT security solutions.

	Solutions \ Challenges	Computation	Communication	Memory	Mobility	Heterogeneity	Scalability	QoS
Confidentiality	Touati et al. [133]	++	–	+	+	+	-	-
	Oualha et al. [110]	+	+	-	+	-	+	+
	Guo et al. [60]	+	-	++	+	+	-	-
	Yao et al. [140]	++	+	+	+	+	-	+
	Su et al. [127]	-	+	+	+	-	-	+
	Thatmann et al. [131]	-	+	+	+	-	+	+
	Chen et al. [36]	++	+	++	-	-	-	+
	Mao et al. [99]	+	+	+	+	+	-	+
Privacy	Evans et al. [52]	+	-	++	+	-	+	+
	Zhang et al. [145]	++	-	++	+	+	-	+
	Alcaide et al. [7]	++	+	+	+	-	+	+
	Huang et al. [70]	+	-	-	+	++	+	+
	Skarmeta et al. [124]	++	+	+	-	-	+	+
	Tonyali et al. [132]	+	-	+	-	-	+	+
Availability	Maleh et al. [95]	+	+	-	+	+	-	-
	Kasinathan et al. [76]	+	-	+	+	+	+	+
	de et al. [48]	-	+	-	+	+	-	+
	Machaka et al. [93]	+	+	-	-	+	-	-
	Shreenivas et al. [120]	+	+	+	+	+	-	-
Blockchain	Hardjono et al. [65]	+	-	+	+	++	++	-
	Gaurav et al. [54]	-	-	-	++	+	++	+
	Hashemi et al. [66]	-	+	-	-	+	++	+
	Kokoris-K et al. [79]	-	+	-	-	+	++	+
	Kamanashis et al. [17]	-	-	-	+	+	++	+
SDN	Flauzac et al. [53]	++	- -	+	-	++	++	-
	Bull et al. [25]	+	-	+	+	+	-	++
	Vandana et al. [134]	+	-	+	-	++	+	+
	Gonzalez et al. [57]	+	-	+	-	+	+	++

We provide in this table a deep analysis and comparison of the solutions we presented previously in this survey according to several security challenges. We use the following notations to assess the level of satisfaction of each solution with respect to the different challenges : + + good ; + average ; - poor (limited) and - - bad.

evolving together in highly dynamic and complex system.

Fine-Grained Secure Control of Smart Actuators in IoT

In this chapter, we focus on some kind of IoT applications where we need to control remotely smart actuators over the Internet using IoT devices such as mobile devices and smartphones. This arises particularly in industrial applications to control manufacturing processes and also in smart home applications to control home appliances such as the control of home alarm systems, the monitoring and the regulation of the temperature level, turning on/off lights, etc. In SCADA based manufacturing systems, there are a tremendous number of smart actuators and sensor devices to monitor and control physical infrastructures. Basically SCADA based control systems are designed in practice in such a way actuators and robots could be accessed and controlled remotely via mobile devices [27, 147].

This problem of remote control of actuators was investigated in some previous research works [3, 46, 56, 112]. However, most of the proposed solutions have only interested in implementing remote control protocols without addressing the security issues that could occur between mobile devices and remote actuators. Indeed, there are vulnerabilities related to the underlying network between mobile devices and smart actuators that were not considered. As examples of such vulnerabilities, one can cite, eavesdropping, denial of service, replay attacks and node compromises among others. Therefore, an access control mechanism shoud be developed in order to guarantee the execution of sensitive actions by only the uthorized users. Particularly, in industrial systems as SCADA, existing security solutions are not scalable enough to meet the requirements of large systems. Indeed, most of existing solutions do not allow to define fine-grained security policies in scalable way to control the access to remote actuators.

In order to overcome this important issues discussed above, we propose in this chapter a new efficient security protocol to secure the execution of remote actions on smart actuators. In our solution, we define fine-grained privileges to the different users based on their roles in the system. To this end, we take advantage of the Attribute Based Encryption tool to define access policies in a more scalable way.

Furthermore, we use one way hash chain as a technique to authenticate IoT devices while preventing against replay attacks.

We evaluated the robustness of our solution in terms of security using AVISPA as a formal verification tool and the obtained results demonstrated that our protocol is secure under various kind of attacks. We also evaluated the performance of our solution and the obtained results show its efficiency in terms of computation and scalability.

3.1 Related works

In this section, we review some secure control based IoT solutions which are closely related to our work.

In [98], the authors proposed a solution to remotely control and monitor home appliances via internal and external mobile devices. The communication between mobile devices and home appliances is done by GSM network and using SMS messages to send commands to the home gateway to perform remote actions. The exchanged control messages are encrypted using AES standard. As a prototype, they developped a Java application to control sensorboxes in smart home environement. The main drawback of this solution is its non scalability since the architecture is not designed to support large number of mobile devices. Therefore, the solution could not be applied in large scale systems such as SCADA systems.

Mantoro et al. [2] proposed an enhanced SSP (Secure simple pairing) for Bluetooth based comunications between smartphones and home appliances in order to control and monitor IoT devices in small areas. Basically, the enhancement consists to improve SSP protocol, designed initially for Bluetooth communications, to resist against man-in-the-middle attack during the connection between smart home device and the mobile device.

Wang et al. [137] proposed a lightweight protocol to secure remote control of IoT devices by portal controllers like smart phones or tablets. The protocol involves three parts : the IoT devices, the portal controllers (smart phones in general) and trust center. The Trust Center is responsible for transfering the control process to legitimate controllers, those later are authenticated against IoT devices based on lightweight hash function and shared keys between IoT devices and controllers. The protocol is resistant to classical attacks such replay, DoS, desynchronization and man in the middle attacks and preserves the privacy of communications like non traceability of control messages between smart devices and controllers. However, the protocol has some scalability issues, since each IoT device needs to share symmetric keys with trust center and all legitimate portal controllers controlling such device.

In addition, the protocol generates an overhead regarding the number of exchanged messages.

Compared to the above schemes, our protocol ensures finegrained access control while minimizing the overhead of message exchanges and cost computation thanks to Attribute Based Encryption and One way hash chain mechanisms.

3.2 Background

In this section, we provide a brief description about Ciphertext-Policy Attribute Based Encryption mechanism and one-way hash chain, which serve as techniques to design our solution.

3.2.1 Ciphertext-Policy Attribute Based Encryption

CP-ABE is powerful encryption tool, proposed by Bethencourt et al. in [16], to ensure fine-grained access control of data shared by an owner. The idea of CP-ABE is that the encryption is done based on a policy that defines relationship between a set of attributes. So, only users that hold a set of attributes that satisfy the policy are able to decrypt ciphertexts. CP-ABE consists of the following algorithms [16] :

- **Setup**(k) : it is a randomized algorithm which is run by the authority. It takes k (security parameter) as a parameter and outputs a master key MK which is kept secret in the authority and a public key PK which is made public and used to encrypt data.

- **Encrypt**(PK, M, Γ) : it is a randomized algorithm which is run by the owner of data. It takes as parameter the public key PK, the message to encrypt M and the policy Γ. The algorithm outputs the ciphertext CT of the message M encrypted under the policy Γ.

- **Key-Generation**(MK, γ) : It is a randomized algorithm, which is run by the authority to generate secret keys D based on a set of attributes γ hold by each user.

- **Decrypt**(CT, D, PK) : it is a deterministic algorithm which is run by the user that wants to decrypt the ciphertext CT based on its secret key D. The user is able to cover the plaintext M only if its secret key D was generated based on a set of attributes γ that satisfies the policy Γ (i.e $\Gamma(\gamma) = 1$). Otherwise, the algorithm outputs $\perp$.

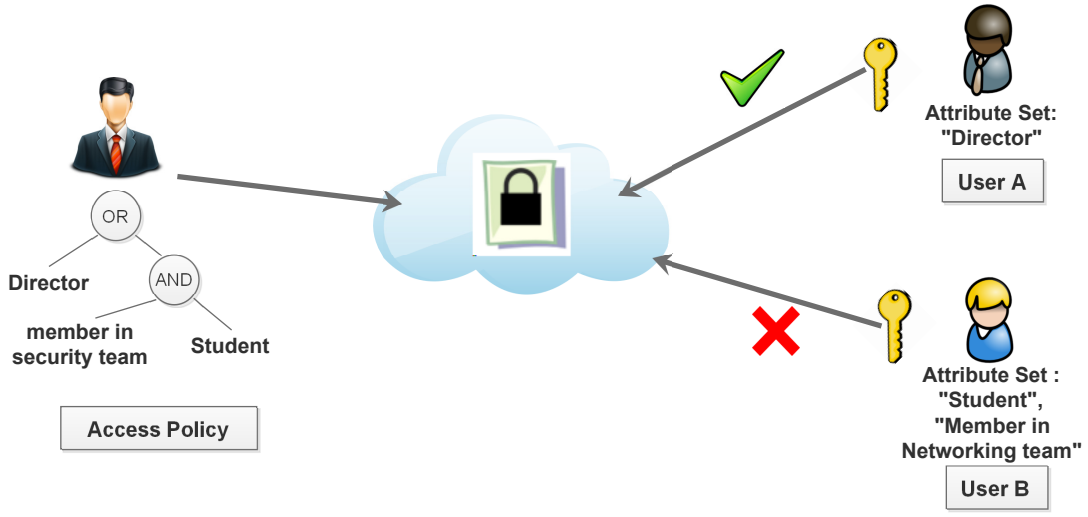


Figure 3.1 – Example of Ciphertext Policy ABE (CP-ABE).

3.2.1.1 Example of encryption with CP-ABE

Consider a simple example in the context of access control in smart building of research laboratory. As shown in figure 3.1, let assume that the director of security team encrypts a file by defining the access policy Γ = ("*Director*" OR ("*Student*" AND "*Member in security team*")), based on the encryption primitive of *CP-ABE*. If the user A holds a secret key that has been generated based on the attributes set γ_A = {"*Director*"}, it will be able to decrypt the cyphertext published by the owner, as the set of attributes γ_A satisfies the policy access Γ. On the other side, if the user B has the attributes γ_B = {"*Student*", "*Member in networking team*"}, it will not be able to decrypt the cyphertext because it does not hold the sufficient attributes that satisfy the access policy.

3.2.2 One way Hash Chain

One-Way Hash Chain is a powerful technique widely used to authenticate data sources in real-time data stream communications. In this work, we use this technique as lightweight mechanism to authenticate IoT devices and users by IoT actuators in such way we overcome replay attacks.

Considering a secure one way hash function $H : \{0,1\}^* \rightarrow \{0,1\}^l$, one way hash chain is defined as a sequence of n hash values : $h_1, h_2, , h_n$ for $n \in N$, where the hash values $h_i \in \{0,1\}^l$, for $1 \leq i \leq n$, are defined as follows :

$$h_k = \begin{cases} H(h_{k-1}) & \text{if } 1 < k \leq n \\ H(m), m \in \{0,1\}^* & \text{if } k = 1 \end{cases} \tag{3.1}$$

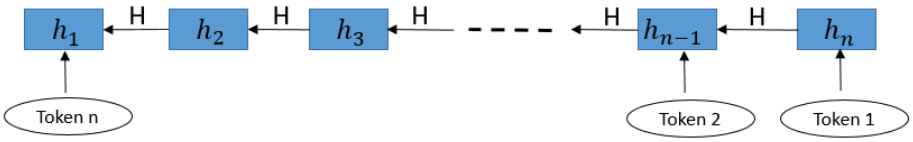


Figure 3.2 – One way hash chain.

The hash chain is designed in such way from a hash value h_i we can compute efficiently the values h_j, for $j < i$, but it is hard to compute the values h_k for $k > i$.

Each value h_i is used as disposable token to authenticate one user in one period of time. The order of tokens' usage is depicted in figure 3.2.

3.3 Models and security requirements

In this section, we present the system model and the security requirements that we consider in the developpement of our solution.

3.3.1 System model

In our system model, we consider a set of IoT smart actuators $SA = \{SA_1, SA_2, \}$ owned by some owner O_i. We note that we could have multiple owners that manage their IoT smart actuators. However for sack of simplicity, consider only one owner. As depicted in figure 3.3, the system that we consider is composed of the following entities :

- **Network of actuators** : a set of IoT smart actuators $SA = \{SA_1, SA_2, \}$ that form the control system network. These actuators are deployed in an area of interest and are internet enabled to allow outside users to control them remotely. We can give as an example an IoT based industrial control system where a set of IoT components are remotely controlled by client IoT devices which are located outside of smart actuators network. Each smart actuator SA_j allows remote execution of a set of actions $A_j = \{A_{j1}, A_{j2}, \}$. Since actuators are usually not powerful, so we have to design lightweight authentication protocol that is less energy consuming.

- **Gateway** : The gateway is deployed at the edge of the actuators' network which serves as a relay between IoT actuators in the inside of the network and other IoT devices at the outside of the network. It is also charged to manage access control to IoT actuators to execute actions remotely.

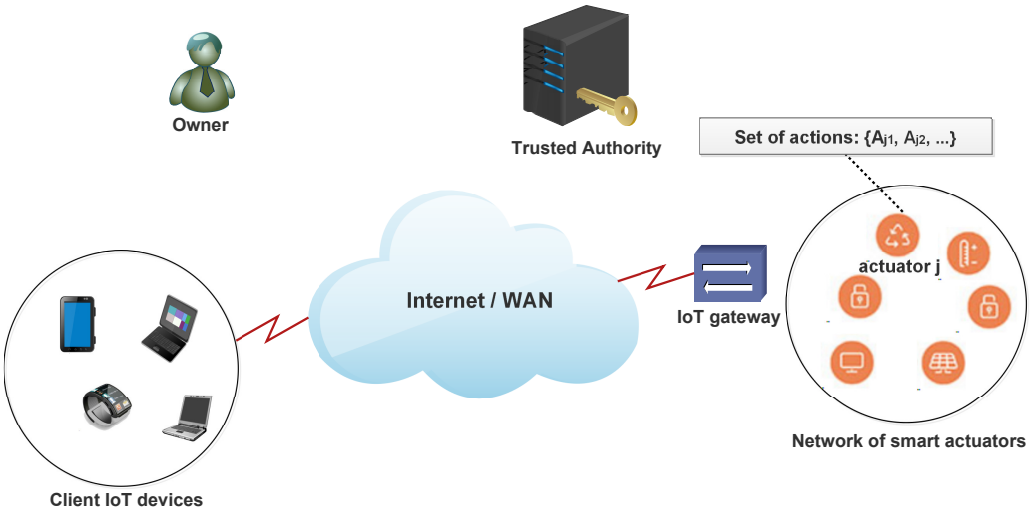


Figure 3.3 – Our system model

- **Client IoT devices** : is a set of different devices $D = \{D_1, D_2, \}$ which are used to execute remote actions on smart actuators. We note that these devices are supposed to be numerous and device D_i could control subset of actuators

3.3.2 Security model and requirements

In our security model, we consider the following assumptions :

- ◇ The central authority is completely trusted by all IoT devices (actuators and sensors) and also by owners and IoT gateways.
- ◇ IoT actuators are not powerful and are not robust against internal attacks trying to comprise them. In this work, we will not discuss mechanisms that deal with attacks trying to compromise actuators and therefore we assume that IoT actuators are available and are not compromised neither spoofed.
- ◇ IoT Gateways are supposed to be *honest-but-curious* which means that they follow the protocol properly but they may be curious about users' behaviors and their privacy.
- ◇ We note by PK_{O_i} and D_{O_i} the owner's public and private keys respectively. Likewise, PK_G and D_G are the public and the private keys of the gateway G. We assume also that the communications between gateway G and IoT actuators as well as between the owner O_i and the gateway are secure (secure channels are established between the different entities).

Under the aforementioned assumptions and the system model previously described,

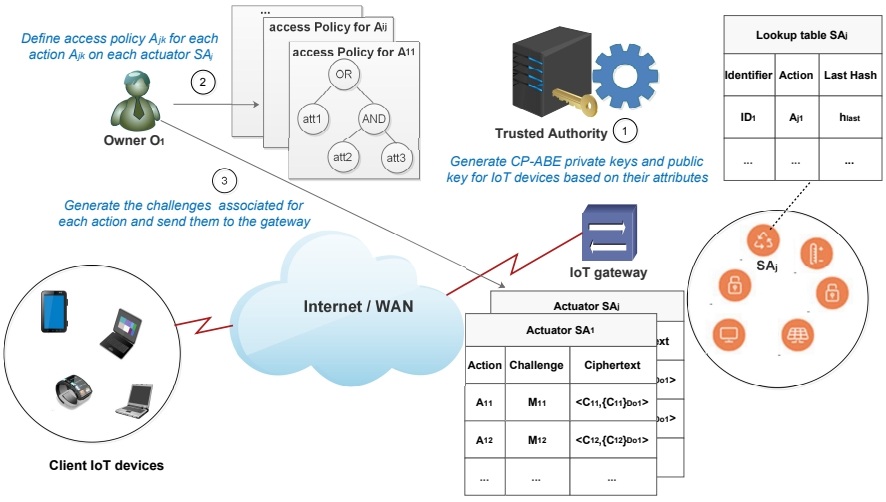


Figure 3.4 – The main steps of our solution.

our solution is a lightweight based token access control protocol to execute actions on remote actuators that satisfy the following security requirements :

- ◇ **Authentication of the IoT devices** : IoT actuators must authenticate client IoT devices to execute actions.
- ◇ **Respect of privileges** : Each IoT device $D_i \in D$ is able to execute only the permitted actions on each actuator $SA_j \in SA$. In other words, the client D_i can only execute the set of actions DA_{ij} on the actuator SA_j.
- ◇ **Replay attacks robustness** : Our authentication protocol must deal with replay attacks. This mean that an attacker or malicious user using an IoT device D_j cannot use the token $T_i(t)$ elapsed by some IoT device D_i $(i \neq j)$ at the instant t to get access to the same actuator by using the same token $T_i(t)$ at the instant $t + 1$.

3.4 Proposed lightweight fine-grained secure control protocol

After introducing the system and security models, in this section, we present in more details our proposed protocol which allows to control remotely the execution of actions on smart actuators. Table 3.1 defines the most important notations used in this section.

In order to design an adaptative and fine-grained access control protocol, we exploit two main mechanisms which serve as the building blocks of our protocol namely :

Notation	Description
O_i	The owner i
D_i	The IoT Device i
SA_j	The smart actuator j
G	The gateway that controls access control to smart actuators
A_{ij}	The action k that could be performed on the actuator k
$H(*)$	One way hash function
$\{M\}_K$	Encryption/Decryption of message M with the key K
PK	The public key generated by CP-ABE keygen algorithm
MK	The master key generated by CP-ABE keygen algorithm. It must be kept secret in the trusted authority
SK_{D_i}	The CP-ABE secret key of the IoT device D_i
M_{jk}	The plaintext used as a challenge to execute the action A_{jk}
P_{jk}	The access policy defined by the owner to execute the action A_{jk}
C_{jk}	The ciphetext of M_{jk} obtained by CP-ABE encryption algorithm and based on the access policy P_{jk}
D_{owner}	The private key of the owner
PK_{owner}	The public key of the owner
D_G	The private key of the gateway G
P_G	The public key of the gateway G
CH_i	The hash chain used by some actuator to authenticate device a D_i

Tableau 3.1 – Table of notations

1) Cipher-Text Attribute Based Encryption and 2) One way hash chain. Basically, our proposed protocol works in three steps that we sketch in the rest of the section.

3.4.1 Initialization

In this phase, the trusted authority generates material keys (public and secret keys) for IoT devices based on the roles of the users (the attributes held by each device D_i).

Each owner O_i defines, for each smart actuator SA_j it holds, the adequate policies for all the actions $a_{jk} \in A_j$ that should be executed remotely on the actuator SA_j. In other words, we associate for each action $a_{jk} \in A_j$ a policy P_{jk} that defines fine-grained access rules based on some attributes held by the smart IoT devices. The relationships between the attributes are established based on logical **"AND"** and

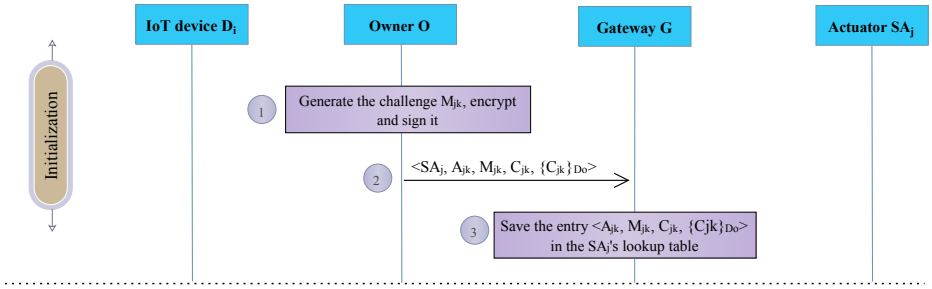


Figure 3.5 – Initialization phase.

"OR" gates as discussed in CP-ABE initial scheme [16].

Subsequently, each owner O_i generates, for each action a_{jk}, a random challenge $M_{jk} \in \{0,1\}^l$. Then, it computes the ciphertext C_{jk} of the challenge M_{jk} encrypted under the policy P_{jk} using the encryption primitive of CP-ABE scheme. All the ciphertexts are also signed by the private key D_{O_i} of the owner O_i. Finally, the owner makes all the encrypted challenges available by sending them under the form of a set of Fivepet $\{< SA_j, A_{jk}, M_{jk}, C_{jk}, \{C_j k\}_{D_{O_i}} >$ where $SA_j \in SA, A_j k \in A_j\}$ to the IoT gateway G that controls smart actuators. The gateway G constructs, for each smart actuator SA_j, two columns table that contains, for each action A_{jk}, its corresponding encrypted challenge $\{C_{jk}\}_{D_{O_i}}$ as illustrated in the figure 3.4.

3.4.2 Token generation

In this phase, client IoT devices negotiate tokens from the IoT gateways to execute actions on smart actuators. As depicted in figure 3.6 (from step 5 to step 10), many steps are carried out by three entities : the client IoT device D_i, the gateway G and the smart actuator SA_j as follows :

- ◇ The client IoT device D_i, which wants to execute action A_{jk} on the actuator SA_j, sends a request message containing the pair $< SA_j, A_{jk} >$ to the gateway G (step 4 in Fig. 3.6).
- ◇ By consulting the table related to the actuator SA_j, the gateway G generates randomly a nonce value $V_j \in \{0,1\}^l$ and responds to the smart object D_i by sending the challenge $< \{C_{jk}\}_{D_{owner}}, C_{jk}, V_j >$, where C_{jk}, $\{C_{jk}\}_{D_{owner}}$ are the ciphertext related to the action A_{jk} and its signature respectively (steps 5 and 6).
- ◇ The IoT device D_i, which holds the private key SK_{D_i} generated by the trusted authority, verifies the signature $\{C_{jk}\}_{D_{owner}}$ using the owner's public key PK_{owner}. If D_i holds the private key SK_{D_i} generated based on a

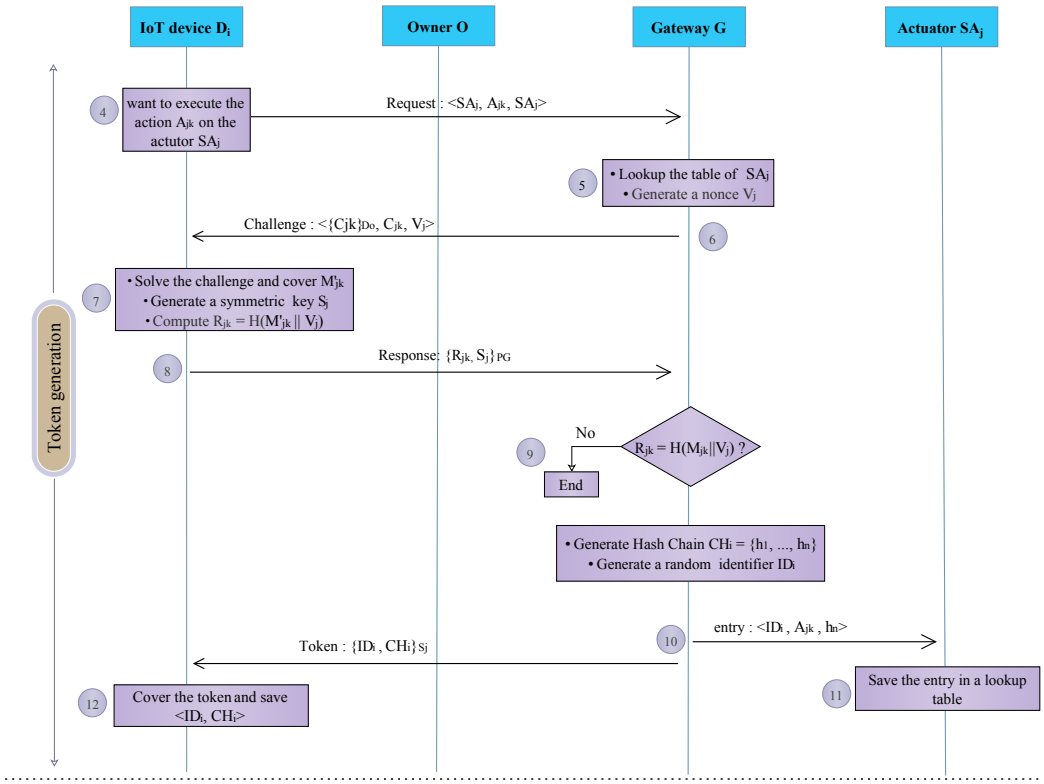


Figure 3.6 – Token generation phase.

set of attributes that satisfies the policy P_{jk}, then it will be able to decrypt the ciphertext C_{jk} using CP-ABE decryption primitive and thereby covers the plaintext M'_{jk}. Then, the device D_i compute the hash of the message M'_{jk} concatenated to the nonce value V_j. Let $R_j k = H(M'_{jk}||V_j)$ the resulting hash value. Finally D_i sends the response R_{jk} to the gateway G (step 7 and 8).

- ◇ The gateway G, upon receiving the response R_{jk}, checks out if $H(M_{jk}||V_j) = R_{jk}$. If so, the gateway G generates an access token for the device D_i to execute the action $A_j k$ which consists on the hash chain $CH_i = \{h_{i1}, h_{i2}, , h_{in}\}$ and sends the triplet $< D_i, A_{jk}, h_{in} >$ to the actuator SA_j along with the whole chain CH_i to the device D_i. The actuator saves, in a lookup table, the triplet $< D_i, A_{jk}, h_{in} >$ which is used to authenticate the device D_i (steps 9 and 10).

3.4.3 Action execution

In this phase, the actuator SA_j can authenticate real time access control of remote IoT devices. The process of authentication is executed as follows :

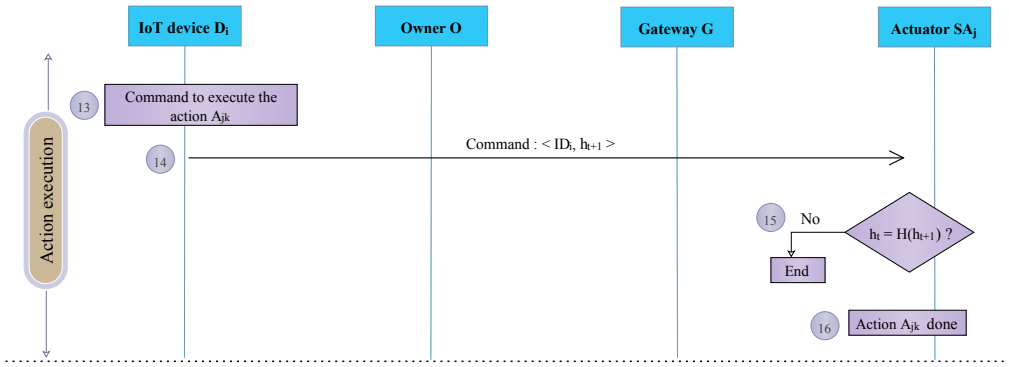


Figure 3.7 – Action execution phase.

- ◇ Device D_i sends a request to the actuator SA_j to execute some action A_{jk}. The request includes a quadruplet of information $< D_i, SA_j, A_{jk}, h_{i(n-t)} >$, where $h_{i(n-t)}$ is the hash value that is used in t^{th} access. For the first access, D_i uses the $(n-1)th$ hash value $h_{i(n-1)}$. For the second access, it uses the value h_{n-2} and son on.
- ◇ Smart actuator SA_j looks for the entry $< D_i, A_{jk} >$ in its lookup table. If this entry is found, the actuators checks out if $H(h_{i(n-t)}) = h_{i(n-t+1)} = h_{last}$. If this condition holds, the actuator SA_j authenticate the device D_i to execute the action A_{jk}. Then, SA_i updates also the value $h_{i(n-t+1)}$ by the received one $h_{i(n-t)}$.

3.5 Security evaluation

In this section, we evaluate the security of our scheme. For that, we give some security analysis and we provide formal verification using the AVISPA tool.

3.5.1 Security analysis

In this section, we present a discussion about the main security analysis and the proof of properties that our proposed protocol ensures.

3.5.1.1 Authentication

In our protocol, there are two levels of authentication. First, the gateway G authenticates the legitimate IoT devices based on a challenge-response authentication technique. Once the first authentication is done, each smart actuator SA_j

authenticates IoT device D_i based on one way hash chain CH_i every time an action A_{jk} is perfomed.

3.5.1.2 Respect of privileges

In our scheme, the execution of each action A_{jk} could be performed only by the legitimate devices. Since, the challenge C_{jk} is encrypted in such a way that only the legitimate devices have the required keys to decrypt the ciphertext as defined in the access policy P_{jk}. Furthermore, CP-ABE scheme does not allow users' collision, which means that two or many illegitimate IoT devices can not cooperate to construct a secret key that allows them to decrypt the ciphertext [16].

3.5.1.3 Replay attacks

During the token generation process, the gateway generates a random nonce value V_i and sends it to the IoT device D_i with the challenge C_{jk} . The response message contains the value $H(V_i||M_{jk})$ that reveals no information about the plaintext M_{jk}. The value $H(V_i||M_{jk})$ cannot be used by another IoT device D_j (i) to get an authorized token to execute the same action since a fresh random nonce value V_i is generated for each token generation request.

In addition, the execution of each remote action on smart actuator elapses one hash value from the hash chain. Therefore, even thought, an intruder intercepts the hash value $h_{i(n-t)}$ at instant t, it cannot deduce the next token $h_{i(n-t-1)}$ thanks to the irreversible mathematical property of the hash function.

3.5.2 Formal verification

3.5.2.1 AVISPA tool

We described our protocol using the AVISPA's High-Level Protocol Specification Language (hlpsl) [130]. The AVISPA tool allows the designers of security protocols to detect potential attacks and verify if their protocols meet the attended security services.

3.5.2.2 The protocol specifications

In our protocol, we defined four roles in HLPSL language. Namely : the owner (O), the gateway (P), the device (D) and the actuator (A) roles which correspond to the

```
role proxy (O,P,D,A : agent,
                Kop, Kap : symmetric_key,
                Ko : public_key,
                Hash : function,
                Pkp : public_key,
                Snd, Rcv : channel(dy)) played_by P
def=
  local State  : nat,
        Nd, Rd, Rep, H, Msg, Cipher, SigCipher, IdAction : text,
        HashChain : (text) set,
        Sk : symmetric_key

  init  State := 0 /\ IdAction := ox

  transition
   1. State   = 0 /\ Rcv({Msg'.Cipher'}_Kop.{Cipher'}_inv(Ko)) =
      State' := 1 /\ SigCipher' := {Cipher'}_inv(Ko)
                  /\ secret(Mo', sec_mo, {O,P,D})

   2. State   = 1 /\ Rcv({Rd'}_Pkp) /\ Rd' = IdAction =|>
      State' := 2 /\ Nd' := new() /\ Snd(Cipher.SigCipher.Nd')
                  /\ request(P,D,auth_po,Hash(Msg.Nd'))

   3. State   = 2 /\ Rcv({Rep'.Sk'}_Pkp) /\ Rep' = Hash(Mo.Rd) =
      State' := 3 /\ H' := Hash(new()) /\ IdAction'= new()
                  /\ HashChain' := {H', Hash(H'), Hash(Hash(H')
                  /\ Snd((HashChain'.IdAction')_Sk)
                  /\ Snd({Hash(Hash(H')).IdAction'}_Kap)
                  /\ secret(HashChain',sec_hc1,{O,P})
                  /\ secret(Hash(Hash(H')), sec_hc2, {O,P,A})
end role
```

Figure 3.8 – **Hlpsl** specifications of the role P (the gateway)

differents agents in our system. The figure 3.8 shows the example of the gateway role in which we specify the different exchanged messages as explained previously. The channel (dy) is modeled in our specifications based on Dolev-Yao intruder model which means that all the exchanged messages between all the agents are intercepted by the intruder. This last can analyze, modify the intercepted messages or eventualy decrypts them if he knows the requiered keys. In our protocol, we analyze some security properties, which are specified in the goal section of hlpsl specifications as shown in figure 3.8. Basically, we verify the following properties :

- ◇ P authenticates P on $H(V_i||M_{jk})$: P generates a nonce value V_i and sends the challenge C_{jk}. If D is able to construct $H(V_i||M_{jk})$ from the challenge C_{jk} and the nonce V_i, P authenticates D.
- ◇ A authenticates D on $H(h_{(n-t)})$: The agent A disposes of the hash value $h_{(n-t+1)}$. If A receives a hash value $h_{(n-t+1)}$ from the agent D such that $H(h_{(n-t)}) = h_{(n-t+1)}$, A authenticates D.
- ◇ Secrecy of CH_i : P generates the hash chain CH_i. It sends the hole chain to the agent D. This information must be kept secrete between P and D.
- ◇ Secrecy of h_n : P generates the hash chain CH_i. It sends the last generated

```
goal
 secrecy_of sec_mo
 secrecy_of sec_ch
 authentication_on auth_po
 authentication_on auth_oa
 authentication_on auth_pa
end goal
```

Figure 3.9 – The hlpsl security goals of our protocol

hash value h_n in the hole chain CH_i to the agent A. This value must be kept secrete between S, D and A.

3.5.2.3 The obtained results

We can see clearly from the figure 3.10 that the obtained results demonstrate the security of our protcol under the test we performed using AVISPA.

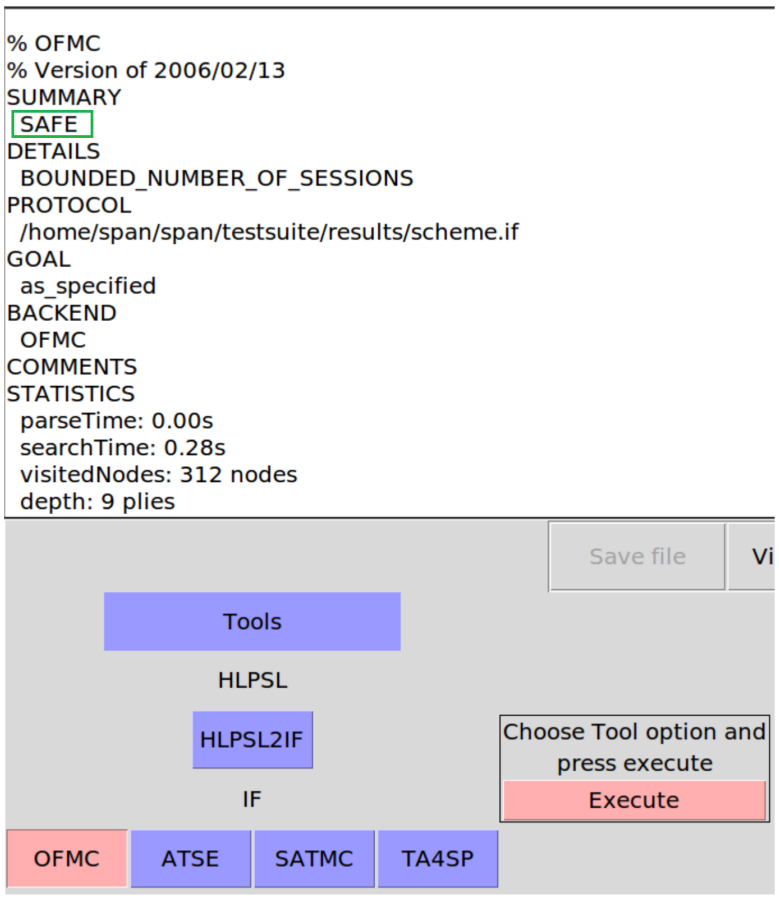


Figure 3.10 – Results reported by the OFMC back-end

3.6 Performance alalysis

In this section, we present the performance analysis of our scheme. The table 3.2 shows the evaluation of our proposed scheme in terms of number of execution of cryptographic operations (enrcyption/ decryption and hash) and the storage occupation with respect to the number of performed actions p by an IoT device D_i on an actuator SA_j. We consider 256 bits as the lenght of each hash value (usage of SHA-256 hash function). Furthermore, we consider the size of each challenge equale to 512 bytes. The size of the identifier of each action is set to 32 bits which allows the owners and the gateway to manage about 4.2 billions of actions on the actuators.

	Computation			**Storage**
	Public key enc/dec	**Secret key en/dec**	**Hash**	
gateway G	1	4	p	$1540 \sum_{j \in SA} \lvert A_j \rvert$
IoT device D_i	3	1	$p+1$	$32\lvert CH_i \rvert + 4$
Actuator SA_j	0	1	p	$36\lvert CH_i \rvert$
Owner O_k	$\sum_{j \in SA} \lvert A_j \rvert$	$\lvert SA \rvert$	-	-

Tableau 3.2 – Computation and storage analysis

For IoT devices and actuators, we notice that our protocol minimises the number of encryption/decryption operations against an increase of the number of hash computations. Indeed, the number of executions of encryption/decryption algorithms does not increase with respect to p. The number of hash computations increases proportionally to p, but their cost is damn negligible compared to the high cost of encryption/decryption operations.

3.6.1 Experiment settings

We performed these simulations in a virtual machine 64 bits ubuntu 16.04 with 2GB of RAM and with Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz Processor. We used the cpabe toolkit implemented by [1], which is based on PBC- 0.5.4 pairing library [2] to implement algebraic operations. We used AES as a secret key encryption scheme, SHA256 as a Hash function, and RSA as a public key encryption scheme. We developped the different simulation scenarios based on Python language.

Subsequently, we discuss furthermore the evaluation of the computation cost of our solution, especially at the IoT device level. Basically, we performed two test scenarios

[1] http ://acsc.cs.utexas.edu/cpabe/
[2] https ://crypto.stanford.edu/pbc/download.html

Challenge Lenght (bytes)	Mean execution time (seconds)
128	0.10343159533
192	0.10304590583
256	0.10351666125
320	0.10426878214
384	0.10431057986
448	0.10558491747
512	0.10569450596
1024	0.10583648784

Tableau 3.3 – Time execution of challenge response

with the variation of several parameters in order to evaluate their impacts on our solution.

3.6.2 Scenario 1 : The evaluation of token generation cost

In this first scenario, we focus on the token generation phase of our protocol. For that, we consider one IoT device D that wants to get a token to execute a certain action on a given actuator. In the test, we compute the execution time that the device D takes to cover the plaintext associated to the challenge and produces the response message. We evaluate especially the impact of the lenght of challenges to be decrypted by D. We assume also that the challenges are encrypted based on CP-ABE encryption algorithm using special access policies that consist of a tree Γ that contains 10 attributes linked with one **"AND"** gate, ie. $\Gamma = AND(att1, ..., att10)$.

The table 3.3 shows the mean execution time of challenge response, performed by one device during the token generation phase, with respect to different lenghts of the challenge.

We notice that the execution time elapsed during the token generation phase is not largely influenced by the length of challenges. Basically, with challenges of 512 and 1024 bytes, token generation phase gets a little more time than with the challenges of 128 bytes ; therefore, we recommend the usage of challenges of 256 bytes as it's still unbreakable by force brute attacks while maintaining a good efficiency.

3.6.3 Scenario 2 : The impact of the action execution rate and the number of IoT devices

In this scenario, we vary the rate of execution of actions arrivals for several IoT devices. Each device D_i periodically perfoms one action in one actuator. To simplify,

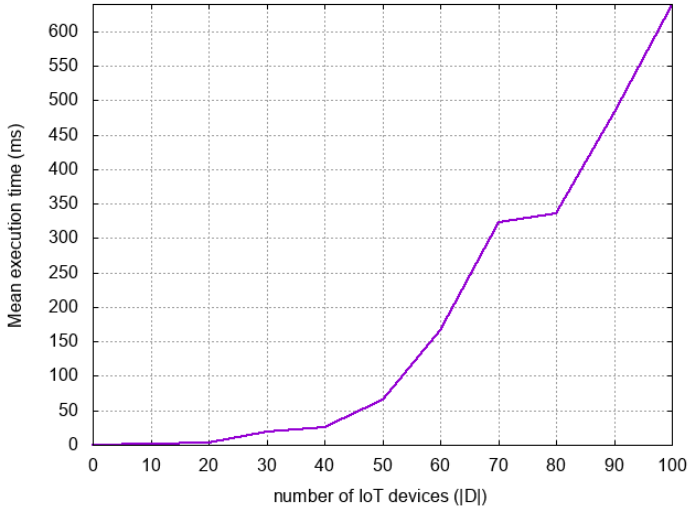


Figure 3.11 – The mean execution time of action execution with respect to λ ($|D| = 10$).

we assume that for each two devices D_i and D_j ($i \neq j$), they execute two actions on two different actuators. In this test, we assume that the gateway can handle one request for only one IoT device at each time, which is the worse case that we can consider [3]. We simulate the rate of action execution requests arrivals according to poisson process of parameter λ. We measure the execution time from event arrival until the execution of the action. Note that, one device D_i executes the token generation algorithm only in the first time in order to get the required token to execute the same action subsequently. Figure 3.11 shows the average of execution time of token generation obtained by varying the number of IoT devices $|D|$ and fixing the parameter λ to the value 0.5 (ie. 2 actions per second for each device). Figure 3.11 shows that the mean execution time of token generation and action execution phases is not hugely impacted by the number of IoT devices $|D|$ neither by the rate of the execution of actions. Indeed, with $|D| = 100$ IoT devices that execute different actions simultaneously and with a rate of one action per 2 seconds for each device, the execution is about 640 ms.

Similarly, in figure 3.12, we plot the average of execution time of token generation with respect to the rate of action execution arrivals λ, we note that the execution time increase with respect to the parameter λ. The increasing shape of the curve 3.12 is close to the linear form, which is satisfactory under the assumption that the gateway can handle only one request of one device each time.

Therefore, our scheme scales very well with complex systems and it could be useful for real time control systems.

[3] In the practice, the gateway can handle several requests simultaneously thanks to multi-threading feature

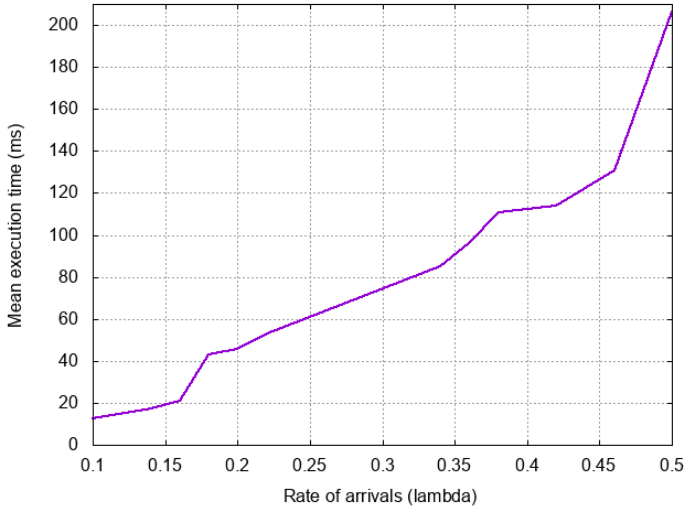


Figure 3.12 – The mean execution time of action execution with respect to $|D|$ ($\lambda = 0.5$).

3.7 Conclusion

In this chapter, we have proposed a distributed protocol to control the execution of remote actions on smart actuators using smart objects and mobile devices. Our protocol has the advantage of being efficient, scalable and allows fine-grained access control. Furthermore, the security analysis, using AVISPA toolkit, showed that our protocol is robust against various attacks. Moreover, we have demonstrated in this paper that our protocol is less energy consuming and scales very well with the number of IoT devices and actuators. The proposed protocol is very suitable for many applications, particularly in resource-limited environments.

Mutual-authentication in IoT based fog computing architecture

According to Cisco [103], more than 50 billion devices are going to be connected to the internet by 2020, with an amount of data that will reach 500 zettabytes. Thus, efficient data management and processing technologies are needed to cover this huge data quantity. Cloud-computing technology has shown its efficiency regarding data processing, high computational power, and data storage and management tasks. Indeed, the cloud already handles many applications, which exploit the strength of this technology, as a service repository. Therefore, it may be an important player in the supply /demand equation that the world is about to face in the coming years.

Cloud computing has been known as a centralized paradigm because all the data need to be processed/stored into one of cloud data centers. However, this operating mode can become problematic when the processed data raise up since it will endure more latency and so a lack of efficiency, especially for real time applications. Consequently, a new paradigm called fog computing has appeared recently to overcome these challenges [20, 125].

Fog computing paradigm aims to extend cloud-computing services to the edge of the network while ensuring interaction with the cloud [69]. Therefore, computation, communication, storage and control operations are performed closer to end users and IoT devices by pooling network's local resources. Indeed, this paradigm adds a resource-rich extra layer composed of a large number of edge devices such as routers, base stations, etc. to ensure an interoperable, low latency and a highly reliable service supply space [144]. With the new fog-computing paradigm, new challenges appear in prospect. Data security is one of the most important challenges of this architecture. Indeed, the fully distributed and untrustworthy nature of this architecture makes data security as one of the main users' concerns [97]. Authentication service is the entry point of any security system which consists of verifying users' identities.

In this chapter, we propose a novel and efficient authentication protocol which ensures mutual authentication at the edge of IoT network. Our scheme performs a

first registration in the cloud level, and then it uses credentials provided by the cloud to realize any eventual mutual authentication between IoT devices and fog nodes without any resort to the cloud. In addition, our solution takes into consideration the eventual authentication between fog nodes. We base our construction on blockchain technology and secret sharing technique. The Blockchain is maintained by fog nodes and it allows IoT devices to authenticate any fog node in the architecture. In addition, it allows fog nodes to establish mutual authentication with end IoT devices. On the other side, end IoT devices are authenticated through secret sharing mechanism. Using blockchain and secret sharing scheme, IoT devices store only a few and a fixed number of information in order to authenticate any fog node in the architecture.

4.1 Related work

Fog computing is a new paradigm, which extends cloud computing services to the edge of the network. This new architecture integrates network edge devices to overcome several cloud computing limitations related to bandwidth and latency. Fog computing architecture introduces a new rich service layer able to interact with any end device with any connection mode such as 4G/5G, WIFI, etc. In addition, it can impressively reduce latency, and provides the expected interoperability and reliability [126, 34], while ensuring the connection with cloud data centers for any eventual control operations and data aggregation. The objectives of fog computing paradigm are not recent and have several similarities with previous proposals such as cloudlet and mobile edge computing (MEC) which has the same goals. Yet, fog computing architecture is the most adequate proposal, which overcomes cloud-computing limitations while ensuring a better reliability compared to cloudlets and mobile edge computing proposals [4, 49].

However, adopting fog computing architecture introduces several challenges. Security is one of the most important challenges, which needs to be addressed in order to attract devices to use this new computing model. Generally, authentication is the first service which needs to be addressed in any security system. As far as we know there have been only one scientific paper [47] which addresses mutual authentication in fog computing. In [47], the author proposed an authentication scheme, which allows any Fog user to authenticate mutually with any Fog server under the authority of a Cloud service provider. In this scheme, a Registration Authority (RA) is deployed in the cloud and defines a random master key for each user. This master key is used to generate secret keys for each fog server in order

to allow them to verify the authenticity of the devices. Thus, each fog server will maintain a secret key for each user in the network. Moreover, each time a user joins the network, the RA generates and sends a secret key to each fog server. Otherwise, the fog servers are not going to be able to authenticate that new user and thus the user will not be able to access the fog server services. In addition, the author in [47] did not consider authentication between fog servers.

Several authentication solutions have been proposed for similar architectures as fog computing. In [135] the authors proposed an authentication scheme based on near field communication (NFC) technologies, which relies on physical contact for pre-authentication in a location-limited channel. Similarly, NFC-based solutions have been used as an authentication model for Cloudlet in [73]. However, this solution cannot always be applied, since there is no guarantee that the devices and the fog nodes are located in a near area. Similarly, password based solutions have been proposed in several architectures [90, 111]. The problem with these solutions is their low entropy since are vulnerable to dictionary attacks. Moreover, due to the untrustworthy nature of fog architecture, the fog nodes cannot be trusted with devices login and passwords. In addition, solutions based on passwords cannot ensure mutual authentication by themselves. Likewise, Biometric authentication techniques are complex and cannot always be applied in fog computing due to the heterogeneous nature of fog architecture, in which several devices do not possess biometric information.

4.2 Background

In this section, we present some cryptographic tools that we will use in our authentication solution.

4.2.1 Review on Shamir's secret sharing scheme

In cryptography, secret sharing refers to a method for distributing a secret amongst a group of participants by giving each one of them a part of that secret. These parts are called shares. The distributed secret can be reconstructed if all the shares are combined together. Otherwise, individual shares are of no use on their own.

Based on the fact that the collection of at least k different points can reconstruct a polynomial of degree $k-1$, Shamir [13] introduced the secret sharing scheme by

dividing a secret S into pieces $(x_i, S_i = q(x_i))$ using a randomly chosen polynomial :

$$q(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}$$

In which a_0 represents the secret S and $(1, S1 = q(1)), (2, S2 = q(2))(k, Sk = q(k))$ are the shares. The polynomial q can be reconstructed using Lagrange interpolation as :

$$q(x) = \sum_{i=1}^{k} Y_i \times \prod_{j=1, j \neq i}^{k} \frac{x - x_j}{x_i - x_j}$$

Where $Y_i = S_i$

Consequently, the secret S can be calculated as $S = q(0)$

4.3 Our solution

In this section, we present our proposed solution, which ensures mutual authentication in fog computing architecture. Table 1, defines the most important notations used in this section.

Notation	Description
$H(*)$	Hash Function
$P(x)$	Polynomial of degree m
(PK_i, SK_i)	Broker B_i's public and private keys respectively
(PK, SK)	Validation public and private keys respectvely, shared between all the brokers
n	a public parameter which defines the group Z_n
(X_{ui}, Y_{ui})	User ui's coordinates generated by one of the brokers
X_{ui}^{-1}	private key related to the public key X_{ui}
F_{si}	Fog node i's share
(FPK_i, FSK_i)	Fog node i's public and private keys
B_i	Broker i
CS	Session key
$H[]$	The Hash chain
σ	Cryptographic digital signature
$\{M\}_K$	The encryption of the message M with the public key K

Tableau 4.1 – Table of notations

In our solution, we consider an architecture (figure 2) composed by the following components :

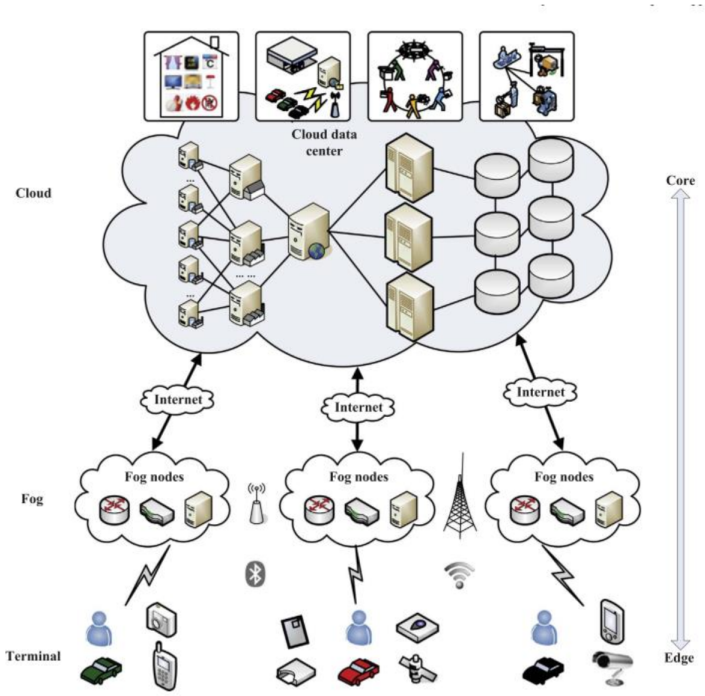


Figure 4.1 – Fog-computing architecture [4]

- ◇ Several cloud brokers responsible for the verification of devices and fog nodes' identities. In addition, these entities distribute authentication credentials for both devices and fog nodes in order to allow them to be authenticated at the edge of the network.
- ◇ Fog nodes, which provide computational services at the edge of the network.
- ◇ End devices, which request services from the fog nodes.

The main idea of our solution is as follows :

An application called *broker* is setup in each cloud in order to verify the authenticity of both the devices and the fog nodes. Thus, each user should perform a first authentication nearby one of the Cloud brokers which verifies the validity of the user's identity. If it succeeds, then it generates credentials and sends them to that user. This allows any fog node to authenticate him at the edge of the network. To authenticate a user, the the concerned fog node performs a first authentication using its certificate at the cloud as well. The aim of this step is to verify the authenticity of the fog nodes and provide some information which allows them to verify devices' credentials at the edge of the network, without contacting the cloud brokers. In addition, the brokers set up a mechanism which allows the devices to authenticate

the fog nodes by using blockchain technology. Indeed, after the verification of fog node's certificate, the cloud broker generates a transaction which contains the node's public key and signs it using its private key. Then, it broadcasts that transaction, so one of the other brokers can validate and insert it into the blockchain. We note that our blockchain is private and it is stored in each fog node. Furthermore, it does not just allow the devices to authenticate any fog node in the architecture, but it also allows fog nodes to authenticate each other.

The mutual authentication starts when a user requests a service from a fog node. First, that user should authenticate the fog nodes from which he requests a service. Thus, it verifies the part of the blockchain where one of the cloud brokers has signed the transaction which contains the public key of that fog node. Once the user verifies the authenticity of the fog node, he should send his credentials to that fog node. Then, based on secret sharing scheme, the fog node combines the user's credential and the information provided by the cloud broker, during its initial authentication, to verify the authenticity of that user.

We note that in our scheme, we do not consider further access control issues with respect to whether the user has the right to run any application in the fog node, or which services he has the right to exploit.

4.3.1 Implementation

In what follows, we show how we can achieve our proposed authentication scheme which allows to verify the authenticity of both devices and fog nodes at the edge of the network.

We note that our solution uses public key cryptography in some points of the authentication process, thus, for sake of illustration in what follows, we consider RSA [119] as a model of public key cryptography.

Our scheme achieves mutual authentication based on secret sharing scheme and blockchain technology, and it works as follows :

4.3.1.1 Setup phase

In this phase, the brokers set up the system parameters that will be used in the eventual registration and authentication phases. We note that it is suffucient if only one of the brokers runs this setup phase and shares the setup parameters with the other brokers. Thus, in what follows, one of the brokers is running this phase as :

- ◇ Initialize a blockchain, which will contain the public key of each legitimate fog node. We assume that each cloud broker B_i already has a pair of keys (public key PK_i and private key SK_i). The key PK_i should be known by the other brokers since the broker B_i uses SK_i to sign each transaction that it generates. In addition, the brokers should share in common another pair of keys (PK, SK). SK will be used to sign valid transactions, while PK will be used by the devices to verify the SK signature at the edge authentication level.
- ◇ Choose a polynomial P of degree m, as follows :

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^m$$

The degree m of the polynomial can be randomly chosen and does not depend neither on the number of devices nor the number of fog nodes. a_0 is considered as the secret token that is going to allow the fog nodes to verify the authenticity of the devices, and $a_i, i \in [1, m]$ are randomly chosen coefficients from Z_p.

- ◇ Choose two primes q_1, q_2 and compute two values $\phi(n) = (q_1 - 1) \times (q_2 - 1)$ and $n = q_1 \times q_2$.
- ◇ Generate m points $Pi(X_i, Y_i)$ randomly from Z_p, and set the verification parameters VP as :

$$VP = \{(token = S), \{P_i(X_i, Y_i)\}, n\} \tag{4.1}$$

4.3.1.2 Fog registration phase

Fog nodes should perform a first registration in the cloud level. Thus, it provides its certificate to one of the cloud brokers. Then the broker runs the following actions :

- ◇ Verify the certificate given by the fog node.
- ◇ Prepare a transaction signed by the secret key SK_i and which contains the public key of the fog node, its state "legitimate".
- ◇ Insert the transaction into a new block (figure 3), and most importantly fill the difficulty field which defines the mathematical problem that should be solved in the validation step.
- ◇ Broadcast the transaction between the blockchain peers (the other brokers) so it can be validated.

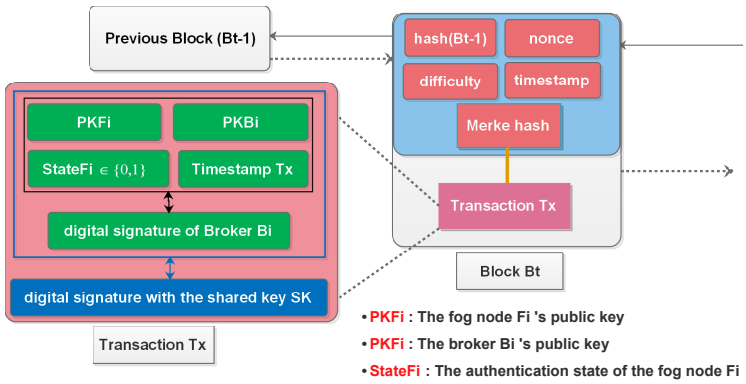


Figure 4.2 – The structure of block in our solution.

To validate the transaction, the brokers run the proof of stack algorithm, which designates one of the brokers B_j to verify and validate the transaction as follows :

- ◇ Verify the signature of the transaction using the broker's B_i public key PK_i
- ◇ If the signature has been successfully verified, solve the mathematical problem defined through the difficulty field in B_i's block.
- ◇ Fill the solution of the mathematical problem in the nonce field, then sign the transaction using the validation key S_k
- ◇ Insert the new block into the blockchain.

We note that the verification in this step has no relation with the certificates' verification that the broker ran before. It only consists of verifying that one of the valid brokers has generated the transaction. Figure 3, shows the structure of our proposed blockchain.

Once the fog node's public key FPK_i has been inserted into the blockchain, the broker B_i, who verified its certificate, sends to that legitimate fog node the verification parameters that were computed in the setup phase, in order to allow to authenticate the devices without resorting to the cloud.

4.3.1.3 Devices registration phase

The devices should also perform a first registration in the cloud level. To confirm its identity, a user needs to successfully be authenticated using the already adopted authentication system in the cloud. After that, the cloud broker generates new credentials to that user in order to allow him to perform any eventual authentications at the edge of the network (fog node level) as folllows :

- ◇ Choose a unique and random X_{ui} from Z_p which is coprime with $\phi(n)$. Then, it computes a X_{ui}^{-1} which is the modular multiplicative inverse of X_{ui} (modulo $\phi(n)$).
- ◇ Generate a unique point $P_{ui}(X_{ui}, Y_{ui})$, where X_{ui} and Y_{ui} in Z_p. We note that P_{ui} has to be different from the P_i points generated in the setup phase.
- ◇ Combine the user's specific point with the m points P_i generated in the setup phase as follows :

$$L_{u,i} = \prod_{j=1}^{m} \frac{X_{ui}}{X_{ui} - X_j}$$

$$Us_{ui} = Y_{u,i} \times L_{u,i}$$

- ◇ Prepare the user's credential as :

$$\text{User's credential} = \{PK, Us_{ui}, X_{ui}, X_{ui}^{-1}, n\} \tag{4.2}$$

Where : PK is the validation public key.

We note that the operations to compute the $L_{u,i}$ and the user's share Us_{ui} are realized in Z_p and do not have any relation with Z_n, where n has been defined in the setup phase. In addition, by sending the public key (PK), we aim to allow the user to verify that the information given by the fog node, in the mutual authentication phase, comes from the valid blockchain and not a falsified one.

4.3.1.4 Mutual authentication phase

Using the credentials given by the cloud broker and the information in the blockchain, both the devices and the fog nodes can mutually authenticate each other at the edge of the network as follows :

Fog node authentication : the device starts by authenticating the fog node through the following steps :

- ◇ Request the transaction from the blockchain in which the cloud broker inserted and signed the fog node's public key and its current state.
- ◇ As soon as the fog node sends back its transaction block, from the blockchain, verify that the received transaction comes from the valid blockchain that the brokers use to publish legitimate fog nodes. Therefore,

given a transaction block defined as :

$$Bc_i = (header, Tx, H(Tx)\sigma_{SK})$$
$$Tx = (Fn_i, H(Fn_i)\sigma_{SK_i})$$
$$Fn_i = (FPK_i, state, timestamp)$$

Where :

✓ *header* the block Bc_i header in the blockchain

✓ *state*= valid or not valid.

The user computes :

$$H_1 = H(T_x)$$

Where : H is a hash function

◇ Verify the signature of the block using the validation public key PK, received as part of its credentials as follows :

$$H_2 = (H(T_x)\sigma_{SK_i})^{PK}.$$

◇ If H_1 is equal to H_2, then the fog node transaction is verified. Otherwise, the user notices that the fog node did not provide a block from the valid blockchain since the signature does not match with the public key PK provided by the broker.

User authentication : once the user verifies the transaction presented by the fog node, it starts its authentication process. Therefore, it sends its credentials encrypted with the fog node's public key as :

$$Credentials = \{Us, X_{ui}\}FPK_i$$

Where :

$$Us = L_{u,i} \times Y_{u,i}$$

On the other side, the fog node verifies the user's authenticity as follows :

◇ Decrypt the received verification parameters using the private key FSK_i.

◇ Perform a polynomial interpolation in Z_p using the values (Us_{ui}, X_{ui}) provided by the user, and the (X_i, Y_i) coordinates provided by the cloud

broker (eq.1) as follows :

$$Lfn_k = \frac{-X_{ui}}{X_k - X_{ui}} \times \prod_{j=1, j\neq k}^{m} \frac{-X_j}{X_k - X_j}$$

$$Fs = \sum_{i=1}^{m} Y_i \times Lfn_i$$

$$\text{Computed token} = Us_{ui} + Fs = S'$$

◇ Compare the two values S' and the token S. If the token S' is valid, the fog node generates a hash chain $H[]$ and a session key CS, which will be used in the eventual further data exchange between the fog node and the authentic user. Then, it encrypts them using X_{ui} as follows :

$$\text{access credential} = \{CS, H[]\}X_{ui}$$

Where :

✓ $\{*\}X_{ui}$ is a public encryption method in Z_n, which uses X_{ui} as a public key.

Note that we linked both secret sharing scheme and public key encyption through X_{ui} value. Thus, this value is not just an importatnt part in the process which prooves that the user is valid member of the group, it also respresents an insurance that the access credentials given by the fog node can only be decrypted an entity which really possesses credentials (eq.4.2) given by the broker.

◇ Finally, the fog node sends the access credentials, and triggers a timer which defines the period of time that the fog node should wait until the first service request from the user.

We note that, if the user does not send any service request, encypted with the session key and contains the first element of the hash chain $H[0]$. Then, by the end of the timer, the authentication session expires. Figure 4, describes the mutual authentication process.

4.4 Threat model

In our protocol, we distinguish two different adversarial models where each model reflects a specific situation defined as follows :

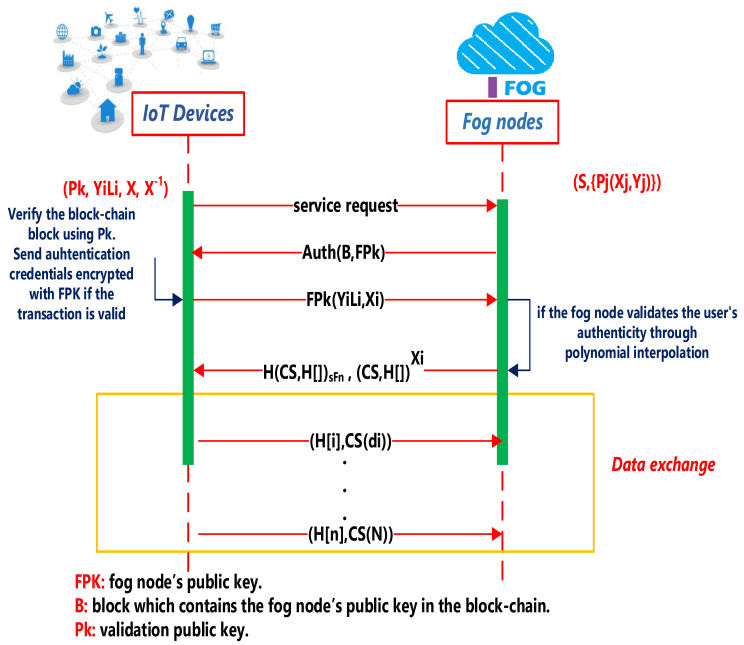


Figure 4.3 – Edge network mutual authentication

1) **The case where an attacker impersonates fog nodes :** Let A be a polynomial time adversary which interacts with a signature oracle. Thus, it submits arbitrary messages m_i to the oracle in order to get the signature of these messages. Finally, the adversary outputs a message m that has never been submitted to the oracle along with its signature.

Adversary A wins the security game if he outputs a valid signature for the message m.

2) **The case where an attacker impersonates end-d :** Let A be a polynomial time adversary which interacts with a random encryption oracle. A submits two messages $\{m_0, m_1\}$ to the oracle. Then, the oracle picks a random coin $b \in \{0, 1\}$ and replies by sending $E(m_b)$, where E is a public key encryption function. Finally, the adversary outputs a guess b' about which one of the two submitted messages $\{m_0, m_1\}$ has been encrypted with E.

The advantage of adversary A in this game is expressed as :

$$Adv = Pr[b = b'] - 1/2$$

4.5 Security analysis

In this section, we prove that our authentication solution ensures the expected security requirements.

4.5.1 Replay/impersonation attack

In our authentication scheme, the fog node verifies the user's credentials, then it sends him the session key with the hash chain encrypted using X_{ui} as a public key. Finally, it sets a timer and waits for the user's request. On the other side, the user needs to get the session key and sends a service request to the fog node before the achievement of the timer. Otherwise, the authentication session will expire. As we can notice, the user needs to send a service request in a limited period of time. Thus, it will be useless for any party to try to replay the user's authentication request since any party, which wants to successfully perform this attack, needs to recover the user's private key X_{ui}^{-1} and get the session key to use it in the eventual service request. Since X_{ui}^{-1} is a secret key generated through one of proven secure public key schemes, as RSA [119], its security is preserved. Likewise, it remains useless to impersonate the user's identity and use its credentials to be authenticated in the fog node, since it also requires the attacker to recover the user's private key X_{ui}^{-1}.

On the other side, if an attacker impersonates an existing fog node identity, it will need to recover the private key of that fog node, which is used to sign access credentials (the session key and the hash chain). Likewise, if an attacker tries to convince a user that he is a legitimate fog node, it needs to provide a valid blockchain transaction signed by one of the known brokers and which contains its public key. Therefore, the attacker has to forge the validation signature key used by the brokers. In the case of RSA signature, a formal proof about its security has been provided in [40].

4.5.2 Man in the middle

If an intermediate node tries to perform man in the middle attack to get access to one of the legitimate fog node, it will need to guess the user's private key X_{ui}^{-1}, in order to find out the session key and send back a service request to the fog node before the achievement of the timer. Thus, this attack will also fail since the probability of guessing the user's private key in a limited time is negligeable.

Tableau 4.2 – Transactions' verification and validation time

	High	**Low**	**Average**
Transaction validation time(ms)	**29000**	**3000**	**10487**
Transaction verification time(ms)	**0.0481**	**0.0211**	**0.0482**

4.5.3 User/ Fog compromise

If a fog node has been compromised, it will not affect the authentication of the users nearby other fog nodes since fog nodes possess only verification parameters and have no knowledge about the devices' private keys X_{ui}^{-1}. Thus, a compromised fog node cannot perform any kind of attack which aims to use any user's credential to get access in other fog nodes. On the other side, a user which has been compromised, can still be authenticated in any other fog node. Thus, it is important that the devices ask for a revocation nearby one of the cloud brokers in case they were compromised. If the system detects misbehavior in any user/fog node, one of the cloud brokers needs to revoke them. Using the blockchain as a repository of the revocation list for both revoked fog nodes/devices can be an adequate solution to manage this situation.

4.6 Performance evaluation

In this section, we evaluate the performance of our authentication scheme. Our experimentations have been realized in a real wireless adhoc network, using two laptops (an HP, i7 laptop with a CPU frequency of 2.7 GHZ and a Samsung i5 laptop with a CPU frequency of 2.6 GHZ). We first measure the computational time that the broker spends in the generation of devices' credentials during the devices' registration phase. Then, we provide the measurements of our edge authentication level and compare it with multi-level certificate-based solution. We note that all arithmetic operations are realized in Z_n or Z_p where p and n are encoded in 1024 bits (128 bytes).

4.6.1 Registration in the Cloud

The registration algorithm in the cloud broker level verifies the user's identity, then it generates credentials for that user. The verification of devices' identities depends

on the authentication algorithm adopted in the cloud. Whereas, the credential generation step consists only of computing some multiplications. Thus, we conclude that the complexity of this phase is linear in the order of $O(n)$, where n is the number of registration requests that the broker receives at the same time. Figure 5, illustrates the computational time of credential generation according to the number of registration requests, received in parallel.

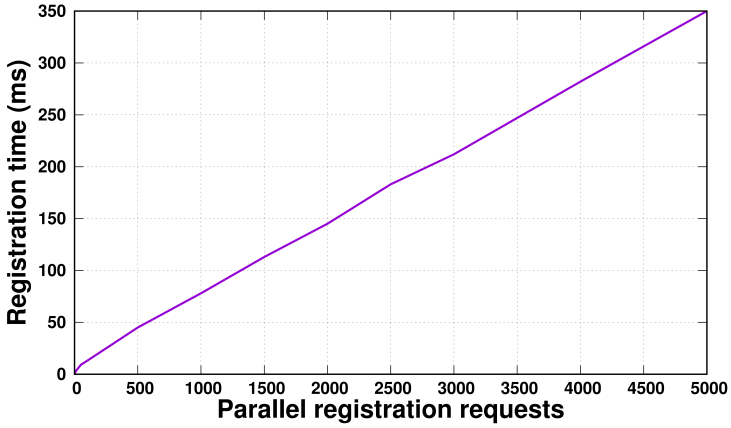


Figure 4.4 – Registration phase

4.6.2 Edge level authentication

As shown in Figure 6, the edge level authentication does not take much time. At this authentication level, the fog node verifies the user's credential. In our solution, almost all computation operations are performed by the fog node, which has a considerable computation power. In addition, the authentication process is performed at the edge level of the network, so it does not occur a considerable latency. In our solution, the fog node will only perform a constant number of multiplication and addition operations to verify the authenticity of the user. Similarly, to authenticate a fog node, the user will only verify the transaction provided by that node. This operation consists of verifying that one of the authorized brokers signed the provided transaction, which is not a time consuming task as shown in Table 4.2.

In existing certificate-based solutions, the fog node will search and verify a set of intermediate certificates going up to a certificate issued from one of the root authorities that the fog node trusts. This operation endures an important latency since the verification depends on the number of intermediate authorities going up to a root authority (certificate level). Note that in our experiments, the intermediate

authorities are in the same network. Thus, the latency can be higher if the authorities are on another network.

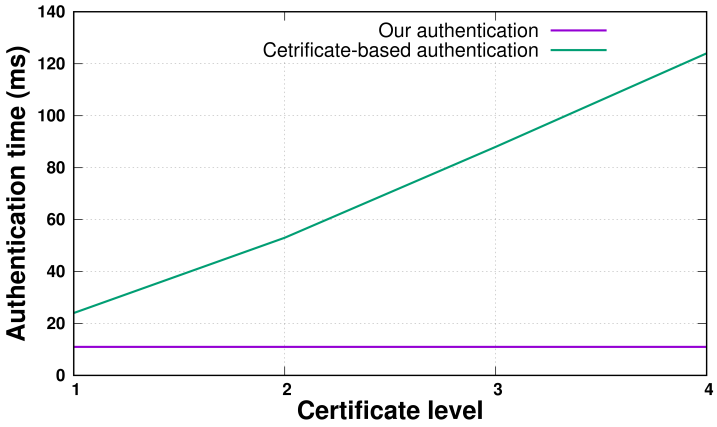


Figure 4.5 – Our solution Vs certificate-based authentication

Tableau 4.3 – Storage and computation cost in our scheme

	Storage (bytes)	Computation		
		user registration	fog registration	mutual authentication
Cloud broker	/	1 Inv + $2m$ Mult + $2m$ Add	1 Sign + Val	/
Fog node	$nbF \times TS +$ $(2m+3) \times 128$	/	/	1 Asm-Dec + $2m$ Add + $2m$ Mult
End user	5×128	/	/	1 Sign-Verif +1 Asm-Enc

Table 5.4 shows the computation and storage overhead in each step of our protocol. Note that, (Inv, Mult, Add) refer to modular inverse, multiplication and addition resp. (Sign, Sign-Verif, Asm-Enc/Dec) refer to RSA signature, signature verification and RSA asymmetric encryption/decryption respectively. (m, nbF, TS) are the predefined polynomial degree, number of fog nodes and trasanction size respectively. *Val* refers to the validation process in the blockchain. As we can notice, all the components perform lightweight arithmetic operations during the different steps of our protocol, except Cloud brokers which sometimes validate blocs in the blockchain. In terms of storage, the devices store few credentials while the fog nodes store the blockchain and verification parameters. Note that the size of the blockchain depends on the number of fog nodes.

4.6.3 Blockchain Performance evaluation

In order to evaluate the performance of validation and verification of transactions, which are part in the process of fog nodes registration, we have measured the average time to validate one transaction as well as the time that the user takes to verify the signature and the content of one transaction. In our evaluation, we use go-ethereum platform [1] which is one of the official implementations of ethereum blockchain protocol. In table 4.2, we present the average time of transaction's validation and signature verification. In this test, we also measure the average memory and CPU occupations. We note that the average memory usage for running the mining process is around 27.9 MO. During the transactions' validation, the percentage of miner's CPU overhead reaches 92.65. We note that the mining operation is done by the cloud brokers, which have an important computation power far away from what we use to evaluate our scheme's performance. Therefore, better results can be achieved as much as we use more computational power.

4.7 Conclusion

In this chapter, we have proposed a new secure authentication scheme based on secret sharing and blockchain technology in fog computing architecture. In our scheme, both the devices and the fog nodes perform one registration in the cloud level. Then, they will be able to mutually authenticate each other at the edge of the network without resorting to the cloud. The devices hold some information which allow them to verify the authenticity of any legitimate fog. Moreover, fog nodes in our solution do not need to store any devices' identifiers and any digital certificates : they only hold a couple of values that are going to allow them to verify the authenticity of any user in the system. In addition, fog nodes can also authenticate each other at the edge of the network using the blockchain. Furthermore, our scheme deals efficiently with situations where an entity from the system tries to impersonate another one in order to get services from fog nodes. Finally, our experimental results show that our proposal realizes mutual authentication in a short time comparing to certificate based authentication approaches. In the future, we intend to address more intensively the revocation problem in fog computing architecture.

[1] https ://github.com/ethereum/go-ethereum

Decentralized Blockchain-Based Trust Management Protocol for IoT

Internet of Things can be viewed as service centric architecture where each device, or thing in general, can request services from other devices and it may also provide services for other devices (service provider). Service centric based IoT applications face several security challenges such as trust management. Indeed, IoT service providers may behave dishonestly and maliciously for the purpose of promoting IoT devices (service requesters) to select them for one or many services on behalf of other trusted service providers. Furthermore, dishonest IoT service providers may perform discriminatory, bad-mouthing and ballot-stuffing attacks to disrupt the network and monopolize many provided services. Therefore, it is clear that a trust management protocol to evaluate the trustworthiness of IoT service providers, in a scalable and efficient way, is more than necessary.

To date, there is a large number of trust management protocols that have been developed for Wireless Sensor Networks (WSN), Social networks and P2P systems in general (eg.[14, 32, 37, 128, 35, 100, 63]). In these protocols, trust evaluation is often based on some information that includes : 1) the direct observations of each node regarding the others (which is gathered whenever the node encounters the IoT service providers) and 2) the indirect recommendations received from other nodes against the service providers. These solutions are still not scalable and are not suitable in high mobility based applications. Indeed, in most solutions, a node needs to communicate with a large number of IoT devices so it would be able to accurately compute trust levels of IoT service providers. Moreover, other questions still arise on how trust information (direct observations and indirect recommendations) is disseminated and shared in a scalable way among different IoT objects in order to speed up the process of trust computation and make it more accurate. In addition, each node has to store this whole trust information about every encountered service provider. Hence, this raises an important question : how we can ensure a fully distributed and scalable trust management protocol with mobility support, in which

IoT devices can evaluate trustworthiness of any service provider in Internet, without the presence of any pre-trusted entity ?

In this chapter, we propose a new scalable trust management solution, named ***BC-Trust***, to address the aforementioned limitations. Our solution is based on blockchain technology and fog computing paradigm, and allows highly mobile IoT devices to accurately assess and share trust recommendations about other devices in a scalable way without referring to any pre-trusted entity.

5.1 Related work

In this section, we review some trust management protocols for IoT which are closely related to our work.

Very recently, Guo et al. [62] provided a comprehensive survey about the most recent works in the trust managements and computational trust models in IoT. They focused basically on service management in IoT dealing with the choice of IoT devices as service providers according to their trustworthiness. They discussed the five fundamental components of each trust management system, namely : trust composition, trust propagation, trust aggregation, trust update and trust formation.

Chen et al. [31] proposed a trust management model based on fuzzy reputation concept for IoT. However, they considered only some specific WSN applications where nodes can establish limited trust relationships with other nodes. Actually, compared to WSN nodes, IoT devices are internet-enabled and can establish complex relationships with other IoT devices and owners.

Saied et al. [118] proposed a multi-service and context-aware trust management protocol for IoT systems, which deals efficiently with different malicious attacks. However, their protocol is based on centralized trusted servers that collect trustworthiness from IoT devices which is not viable in IoT. Similarly, Guo et al. [61] proposed a 3-tier hierarchical architecture based on cloudlets to disseminate trust information to a central cloud. Their architecture allows IoT devices to report trust information and also query trustworthiness of other devices directly from the local cloudlets. However, the proposed architecture refers always to the central cloud which is responsible to disseminate the trustworthiness information gathered from one cloudlet to the other cloudlets which can involve latency issues. Moreover, their trust model is still limited, since distributed cloudlets are assumed to be honest in their architecture and they maintain only trust data in their geographical area.

The concept of social Internet of Things has been developed recently in many

works. This concept consists on extending the world of IoT in such away, IoT devices will be able to establish autonomously social relationships between other devices and users. Many works have investigated the trust management problem in the context of social IoT [33, 75, 87, 105]. Chen et al. [33] proposed an adaptive trust management protocol for social and dynamic IoT systems. The main idea consists on distributing the computation of trust information among IoT devices. In their computational model, each device maintains its own trust assessment toward other users and devices. The trust assessment is based on the recommendations of the other devices, the direct observations and also the history of the interactions. The authors considered different classes of trust properties such as QoS, honesty and cooperativeness depending on the social relationships between IoT devices. However, their protocol is not scalable enough since each device must save all the trust pieces of information (that include its history and the recommendations of the other devices, etc.) related to its social friends (IoT devices and owners) in a lookup table. In [105], the authors proposed two trustworthiness computational models. 1) A subjective model which basically consists on the combination of the local trust parameters (direct observations) and also the received indirect recommendations. And 2) An objective model, where they proposed to disseminate trust assessments in a distributed Hash table maintained by a subset of trusted IoT devices. However, this last assumption is not actually practical in IoT environments. Moreover, their solution is still limited and it is applicable only in social based IoT applications.

Recently, Lu et al. [139] proposed a new blockchain based trust management solution for vehicular networks (VANETs). The idea of their solution is to use the blockchain as a platform to share the reputation opinion reported by different vehicles. The blockchain is maintained by the road side units (RSU), which are also the miners. The authors proposed a new consensus algorithm that favors blocks containing a large variation of trust values. However, the proposed consensus method is vulnerable to some kind of collaborative attacks aiming to report high or low trust values to generate priority blocks and then disrupt blockchain trust values. Similarly, Yang et al. [91] proposed a privacy-preserving trust model for VANETs that combines blockchain and public key infrastructure to deal with tracking attacks while broadcasting forged messages. However, the authors did not discuss the security of their trust management protocol against trust attacks like bad mouthing and ballot stuffing attacks.

5.2 Security model

In this Section, we define our security model by highlighting the main security attacks that may occur in our system. In our model, we assume that every IoT device may provide services for other devices and it may simply behave as service requester. Moreover, we consider dishonest service providers that act for their own benefits in order to be selected as service providers by other IoT service requesters. Thus, each malicious service provider can perform the following trust-related malicious attacks [62] :

- ◇ *Self-promotion attacks* : a malicious service provider can promote its importance to other service requesters by sending good recommendations about itself, then it may act maliciously by providing bad services.
- ◇ *Bad-mouthing attacks* : a malicious service provider can distrust the trustworthiness of other trusted service providers by providing bad recommendations about them to service requesters and therefore decrease their chances to be selected as service providers. These attacks could be performed in a collaborative way by a set of malicious nodes to ruin well-behaved nodes.
- ◇ *Ballot-stuffing attacks* : a malicious service provider can consolidate other malicious service providers and boost their trustworthiness by providing good recommendations. Therefore, this may increase their chances to be selected as service providers. Similarly to Bad-mouthing attacks, this attack could be performed in collaborative way by malicious nodes to recommend each other.
- ◇ *Opportunistic service attacks* : a malicious service provider can decide to provide opportunistically a good service to attract the service requesters and enhance its reputation regarding them. This malicious node could exploit this good opportunistic reputation to perform successful Ballot-stuffing and Bad-mouthing attacks.
- ◇ *On-off attacks* : in this kind of attacks, one node can decide to provide good and bad services in a random way to avoid the risk of not being selected as a service provider. Once again, with good reputation, this malicious node can perform Ballot-stuffing and Bad-mouthing attacks with the collaboration of other malicious nodes.

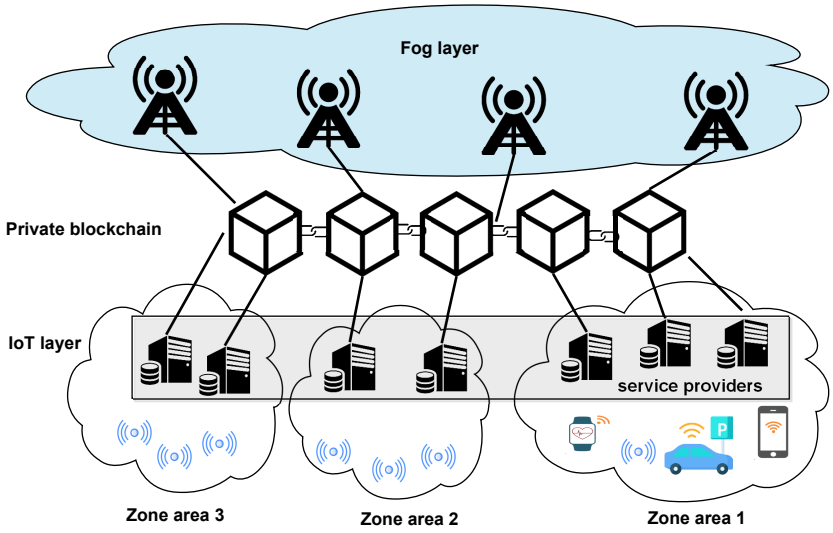


Figure 5.1 – Our system architecture

5.3 Our Trust management solution

In this section, we prensent our architecture, then we define the main steps of our protocol which allows any entity in our architecture to measure the trustworthiness of any service provider.

5.3.1 Our architecture

In our solution, we consider a trust management architecture, composed of the following components :

- ◇ **IoT service requesters** are nodes which communicate with any other component in the architecture, via Internet or other network protocols. Each service requester can request services from other service providers in the architecture. We denote the set of service requesters by $D = \{O_1, O_2, ..., O_N\}$.
- ◇ **Service providers** are nodes which offer services to other devices. We denote the set of service providers by $Sp = \{Sp_1, Sp_2, ..., Sp_M\}$. Note that in our architecture, we assume that service providers are powerful nodes that have enough computational power and storage capacity to validate and maintain blocks into the blockchain.
- ◇ **Fog nodes** are responsible for a reliable management of trustworthiness in the system. Indeed, the set of fog nodes $FN = \{FN_1, FN_2, ..., FN_P\}$ maintains a blockchain which stores the various trust values related to

service providers. In addition, fog nodes provide to service requesters a global view on the trustworthiness of each service provider. Note that these fog nodes are not assumed to be trusted. Indeed, since our solution is based on blockchain there is no need to trust any node as far as the whole blockchain is trusted. Moreover, we assume that fog nodes layer covers all the IoT network scope and can manage the high mobility of IoT devices.

- ◇ **Blockchain layer** which is maintained between fog nodes and service providers. The blockchain layer is responsible for the management of trustworithness data reported by IoT devices. In our architecture design we use private blockchain as we restricted the validation process to only service providers which are known in our system. Thus, we believe that consortium blockchain is relevant solution for our protocol. Moreover, we combined PBFT and PoS protocols to design a scalable and secure consenus protocol (see section 5.3.5).

We illustrate in Figure 5.1 our architecture on which we base to propose our trust management protocol.

5.3.2 Our Trust model

In our trust model, we usually use the following appellations that we define as :

- ◇ Trust value $T_{ij}^{S}(t)$: is a real number in the range $[0, 1]$ which expresses the trust level of IoT device O_i toward IoT service provider Sp_j with respect to the service S at instant t. The max value 1 means that the node Sp_j (trustee) is full trusted with respect to the node O_i (trustor) and 0 indicates that service provider Sp_j is a bad or malicious node.
- ◇ Recommendation $R_{ij}^{S}(t)$: is a real number in the range $[0, 1]$ computed by a fog node based on the trust values, which concern service provider Sp_j, reported by IoT devices. This value is sent to IoT device O_i.
- ◇ Direct Observation $D_{ij}^{S}(t)$: is a real number in the range $[0, 1]$. It represents the mean of satisfactions against the service S during the interactions between device O_i and service provider Sp_j.

Figure 5.2 illustrates our trust model, in which we define trust parameters used in our protocol. In addition, Table 1 summarizes the main notations used in this paper.

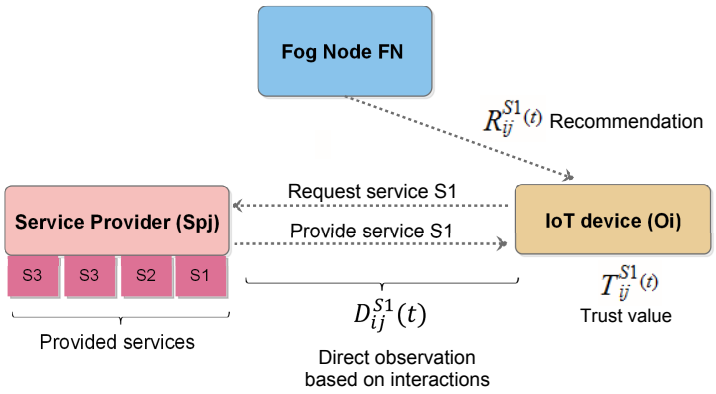


Figure 5.2 – Our trust model

Notation	Description
O_i	The IoT device i
PK_i	The public key of IoT device O_i
SK_i	The private key of IoT device O_i
S_k	The service k
$T_{ij}^{S_k}(t)$	Trust of O_i toward Sp_j w.r.t. service S_k at time t
$D_{ij}^{S_k}(t)$	Direct observation of O_i toward Sp_j w.r.t. service S_k at time t
$R_{rj}^{S_k}(t)$	Recommendation of O_r toward Sp_j w.r.t. service S_k at time t
S_{ij}	Satisfaction level of O_i toward Sp_j
α_{ij}	Accumulated satisfaction level of O_i toward Sp_j
β_{ij}	Accumulated dissatisfaction level of O_i toward Sp_j
α	Weight on previous experiences
β	Weight on direct observation
γ	Weight on indirect recommendations
ΔT_R	The period of time that separates two transactions
$Tx_R(O_i, Sp_j, S_k)$	Transaction that contains the recommendation of node O_i toward node Sp_j

Tableau 5.1 – Table of notations

5.3.3 Our protocol BC-Trust for trust management

BC-Trust is a real time, evolutionary and encounter-based assessment process, which provides trust information about any service provider. Indeed, in our protocol, *"honest"* IoT devices continuously evaluate and update trust information about encountering IoT service providers whenever they request a service. In what follows, we explain the different steps of our protocol ***BC-Trust***.

5.3.3.1 Setup phase

The setup phase of our protocol is completed in two following steps :

1. **Identification step** : in a massively distributed system of a very large number of heterogeneous IoT devices, the identification of IoT devices is one of the major challenges that must be addressed before developing a trust management protocol [62]. In our system, we assume that there is a public key infrastructure which is responsible for cryptographic key generation. Therefore, PKI authority generates a public and private key pair for each IoT device and fog node in the architecture. The public keys are maintained in the blockchain by the fog nodes. Thus, once PKI authority generates the pair (PK_A, SK_A) for each entity A, it sends a transaction containing PK_A to the blockchain. PK_A (public key) serves as an identifier of entity A. Hence, at the end of identification step, all entities in the architecture are able to identify each other via the blockchain.
2. **Service indexing step** : in order to allow IoT devices to discover available services, service providers register their proposed services into their closest fog nodes. Thus, we propose to use a distributed hash table (DHT) to store the different services provided by different service providers. This DHT table, maintained by the fog nodes, is synchronized and updated via a distributed protocol similar to structured P2P networks [42].

5.3.3.2 Trust Dissemination Phase

During the execution of the protocol, each *"honest"* IoT device O_i periodically reports its recommendations toward the encountered service providers every ΔT_R time units (ΔT_R is a system parameter). Device O_i's recommendations are reported to the closest fog nodes. For sake of optimization, each device O_i reports only the most fresh recommendations that have been updated during the last ΔT_R. Therefore, at the end of ΔT_R, the reported trust values are structured in separate transactions, where each transaction $Tx_R(O_i, Sp_j, S_k)$ contains the following information :

- ◇ The trustor node identifier : which is the public key PK_{O_i} of service requester O_i.

- ◇ The trustee node identifier : which is the public key PK_{Sp_j} of service provider Sp_j.
- ◇ The service S_k that has been provided by node Sp_j to device O_i during the last ΔT_R.
- ◇ A set of criteria $C' \subset C = \{C_1, C_2, ..., C_N\}$: that represents the criteria on which O_i has based its evaluation of service S_k.
- ◇ The trust value $T_{ij}^{S_k}$ that refers to the level of trustworthiness of the service provider Sp_j assessed by the device O_i with respect to the service S_k and criteria C'.
- ◇ The timestamp $tsp_{ij}^{S_k}$ of the last updated trust value $T_{ij}^{S_k}$.
- ◇ The previous $\{R_{ij}, \Delta T = [t_1, t_2]\}_{SK_{FN_l}}$ signed by FN_l and computed based on trust values reported by IoT devices regarding service provider Sp_j. The computation of R_{ij} takes in consideration only the reported trust values in the interval $\Delta T = [t_1, t_2]$. Further explanations about the computation of R_{ij} are provided in phase 5.3.3.5.
- ◇ The approval of service S_k signed by the service provider Sp_j as : $\{approval, S_k, timestamp\}_{SK_{Sp_j}}$. This information is used as a proof that service S_k has been accomplished and provided by Sp_j and thus it prohibits that O_i can report a recommendation about the service provider Sp_j without requesting any service from it.

The device O_i signs the transaction $Tx_R(O_i, Sp_j, S_k)$ by its private key SK_{O_i} and sends it to all its closest fog nodes. In order to avoid some security threats related to dishonesty of some fog nodes[1], each device must send the transaction to many fog nodes (the closest ones) to increase the probability of its insertion into the blockchain. Upon receiving the transactions, the fog node periodically performs the following steps :

1. It first verifies these transactions by verifying the signature of both service provider Sp_j (the approval signature) and service requester O_i (the transaction signature).
2. It gathers only the valid transactions in one single block.
3. It broadcasts the block to be validated to the whole service providers that maintain the blockchain.

[1]The fog nodes can also be compromised. They can drop, delay, modify and redirect the received messages.

4. Finally, once the validation is done, the block will be added to the blockchain by all fog nodes and service providers.

5.3.3.3 Trust assessment process

When a node O_i requests the service S_k from service provider Sp_j at time t, it first queries for the available services from the distributed hash table (maintained by the fog nodes) to identify the potential IoT service providers it should interact with them. Node O_i will choose one service provider Sp_j among others based on the trustworthiness level of each service provider at time t. The trustor IoT device O_i assesses or updates the trustworthiness of service provider Sp_j (trustee) as follows :

$$T_{ij}^{S_k}(t) = \begin{cases} \alpha T_{ij}^{S_k}(t-\Delta t) + \beta D_{ij}^{S_k}(t) + \gamma R_{ij}^{S_k}(\Delta t), \text{ if } P(i,j) \\ R_{ij}^{S_k}(\Delta t), \text{ otherwise} \end{cases} \tag{5.1}$$

where $0 \leq \alpha, \beta, \gamma \leq 1$ and $\alpha + \beta + \gamma = 1$, are used to weigh the importance of each trust parameter. These weighs are adjusted dynamically by the trustor in order to maximize the accuracy of trust assessment as well as make the protocol more resilient to bad-mouthing and ballot-stuffing attacks. In equation (5.1), $P(i,j)$ is a predicate that is equal to *true* if the device O_i has interacted previously with the service provider Sp_j, otherwise $P(i,j) = false$.

In equation (5.1), we distinguish two main cases depending on the experience of device O_i with the encountered IoT service provider Sp_j :

1. **Case 1** : if the device O_i has previously encountered the service provider Sp_j, it will assess its trustworthiness level based on $T_{ij}^{S_k}(t-\Delta t)$, $D_{ij}^{S_k}(t-\Delta t)$ and $R_{ij}^{S_k}(\Delta t)$. $T_{ij}^{S_k}(t-\Delta t)$ represents the last trustworthiness of service provider Sp_j. $D_{ij}^{S_k}(t-\Delta t)$ represents the direct observation measured till instant t. The last parameter denoted by $R_{ij}^{S_k}(\Delta t)$ refers to the indirect recommendations of the other IoT devices toward Sp_j.

2. **Case 2** : if the device O_i has not interacted previously with the service provider Sp_j and it does not dispose of any previous trustworthiness level $T_{ij}^{S_k}(t-\Delta t)$ about Sp_j, then it considers only the indirect recommendation $R_{ij}^{S_k}(\Delta t)$ as trustworthiness value $T_{ij}^{S_k}(t)$.

5.3.3.4 Computation of $D_{ij}^{S_k}(t)$

When a device O_i requests one service S_k from Sp_j, it measures the satisfaction level of the provided service. Let $S_{ij}(t)$ be the current satisfaction level, which is a real number in the range $[0, 1]$. The direct observation $D_{ij}^{S_k}(t)$ is :

$$D_{ij}^{S_k}(t) = \frac{\alpha_{ij}}{n} = \frac{\sum_{t_i \in \{t_1,..,t_n\}} S_{ij}(t_i)}{n} \tag{5.2}$$

where :

- α_{ij} is the cumulative of the satisfaction levels and is continuously updated by $\alpha_{ij} = \alpha_{ij} + S_{ij}(t)$.
- $t_1 < t_2 < ... < t_n = t$ represent the instants where service S_k was requested.
- n is the number of experiences regarding the service S_k.

Algorithm 1 summarizes the different steps of trust assessment protocol, executed by IoT devices.

Algorithm 1 Trust assessment-IoT devices level

Input : O_i : IoT device, Sp_j : IoT service provider
procedure ComputeAndReportTrust
    [t]*Requests a recommendation about Sp_j from the home fog node*
    Fog node sends the recommendation $R_{ij}^{S_k}$ to O_i
    if $(T_{ij}^{S_k}, D_{ij}^{S_k}) \in$ *loockup(O_i)* **then**
        $T_{ij}^{S_k} \leftarrow \alpha \times T_{ij}^{S_k} + \beta \times D_{ij}^{S_k} + \gamma \times R_{ij}^{S_k}$
    else
        $T_{ij}^{S_k} \leftarrow R_{ij}^{S_k}$
    end if
    if $T_{ij}^{S_k} <$ *Threshold* **then**
        Ignore the service provider Sp_j
        return false
    else
        Service S_k Done
        Evaluate the satisfaction $S_{ij}(t) \in [0, 1]$
        $\alpha_{ij} \leftarrow \alpha_{ij} + S_{ij}(t)$; $n \leftarrow n + 1$;
        $D_{ij} \leftarrow \frac{\alpha_{ij}}{n}$
        Update the entry (D_{ij}, T_{ij}) in the loockup table
        Construct and send transaction $Tx_R(O_i, Sp_j, S_k)$
        return True
    end if
end procedure

5.3.3.5 Computation of $R_{ij}^{S_k}(\Delta t)$

As previously explained, our trust assessment is also based on recommendations provided by fog nodes. These recommendations are computed using trust values stored in the blockchain.

To provide indirect recommendation $R_{ij}^{S_k}(\Delta t)$, fog node FN_l starts by filtering out the most recent transactions, which have been occurred during the last Δt time units, available in the blockchain. We denote by L the list of IoT objects which have reported the filtered transactions. Next, from the list L, we distinguish two cases :

1) **Case 1 ($L \neq \emptyset$)** : fog node FN_l computes $R_{ij}^{S_k}(\Delta t)$ as follows :

$$R_{ij}^{S_k}(\Delta t) = sp \times Rs_{ij}^{S_k}(\Delta t) + (1 - sp) \times Ro_{ij}^{S_k}(\Delta t) \tag{5.3}$$

where :

- sp : the rate of service providers in the list L ($0 \leq sp \leq 1$)
- $Rs_{ij}^{S_k}(\Delta t)$: the average of the recommendations provided by service providers.
- $Ro_{ij}^{S_k}(\Delta t)$: the weighted average of the recommendations provided by IoT devices.

Overall, in equation (5.3), the computation of $R_{ij}^{S_k}(\Delta t)$ depends upon two different values $Rs_{ij}^{S_k}(\Delta t)$ and $Ro_{ij}^{S_k}(\Delta t)$. Indeed, in our solution, service provider Sp_j could be recommended by both IoT devices or other service providers.

Therefore, in the list L, fog node FN_l selects the subset L_O of IoT devices. Then, it computes $Rs_{ij}^{S_k}(\Delta t)$ as follows :

$$Ro_{ij}^{S_k}(\Delta t) = \frac{1}{(1 - sp)|L|} \sum_{k \in L_O} T_{kj}^{S_k} \tag{5.4}$$

where :

$L_O \subset L$: is a subset of L that contains only service requesters.

Equation (5.4) represents the average of all recommendations ($T_{kj}^{S_k}$) that were reported by all devices $O_k \in L_O$ and stored in the blockchain during the last period ΔT.

Likewise, fog node FN_l, selects the subset $L_S (L_S \subset L)$ of IoT devices. Then, it

computes $RS_{ij}^{S_k}(\Delta t)$ as follows :

$$Rs_{ij}^{S_k}(\Delta t) = \sum_{k \in L_s} \frac{T_{ik}^{S_k}}{\sum_{k \in L_s - \{j\}} T_{ik}^{S_k}} \times T_{kj}^{S_k} \tag{5.5}$$

Equation (5.5) represents the weighted average of all recommendations $T_{kj}^{S_k}$ that were reported by all devices $Sp_k \in L_S$.

In fact, each recommendation value $T_{kj}^{S_k}$ provided by Sp_k is weighted by the ratio of the trust value reported by O_i toward Sp_k, to the sum of all trust values given by O_i toward each service provider in L_S. Hence, if trust value $T_{ik}^{S_k}$ of O_i toward Sp_k is high, then the fog node will attribute a high weight to the recommendation $T_{kj}^{S_k}$. For sake of optimization, the fog node only considers the recommendation coming from service providers that device O_i grants them a minimum trust value *Threshold*. As an example, fog node considers the recommendations provided by the service providers if their trust value regarding O_i exceed 0.7 (i.e. $T_{ik}^{S_k} > 0.7$).

Finally, fog node FN_l computes the recommendation $R_{ij}^{S_k}(\Delta t)$ and responds to the device O_i by sending $\{R_{ij}^{S_k}(\Delta t), \Delta t = [t_1, t_2]\}_{SK_{FN_l}}$ signed by its private key SK_{FN_l}. The device O_i will integrate this information in the next transaction as explained previously in Section 5.3.3.2. It allows the other fog nodes to detect any misbehavior from fog node FN_l during block validation step.

2) **Case 2** $(L = \emptyset)$: this case means that there have been no device which recommended Sp_j during the last ΔT time units. If service provider Sp_j has never been recommended by any IoT object in the architecture, then fog node FN_l returns a recommendation $R_{ij}^{S_k}(T) = 0.5$. Otherwise, fog node FN_l searches the most recent transaction Tx_R that has been reported prior interval $[t - \Delta T, t]$. Since Tx_R has not been reported in the last ΔT, it is still considered as an old transaction. Therefore, fog node FN_l will consider recommendation reported in transaction Tx_R with a small penalty *Pnl*. In our solution, we consider a constant penalty *Pnl* equal to 0.05. Thus, let $R_{kj}^{S_k}(t')$ be the recommendation reported in Tx_R such that $t' < t - \Delta T$, fog node FN_l computes the recommendation $R_{ij}^{S_k}(\Delta t)$ as follows :

$$R_{ij}^{S_k}(\Delta t) = (1 - Pnl) \times R_{ij}^{S_k}(t') \text{ , where } t' < t - \Delta T$$

Algorithm 2 summarizes the different steps performed by fog nodes while computing recommendations.

We illustrate in Figure 5.3 the different steps of our protocol.

Contermesure against selfish service providers : it is possible that some service

Algorithm 2 Trust assessment-Fog nodes level

procedure COMPUTERECOMMENDATION
  Init1 : $L_S \leftarrow \{\}$; $L_O \leftarrow \{\}$
  [t] *Init2* : $L \leftarrow$ *the* T*-th most recent recommenders that reported transactions in* $[t - \Delta T, t]$
  if $L = \emptyset$ **then** $R_{ij}^{S_k} \leftarrow 0.5$
    if $\exists R_{kj}^{S_k}(t^{'}) \in$ *Blockchain* && $t^{'} < t - \Delta T$ **then**
      $R_{ij}^{S_k} \leftarrow (1 - Pnl) \times R_{ij}^{S_k}(t^{'})$
    end if
    Send the recommendation R_{ij}^{Sk} *to the device* O_i
    return $R_{ij}^{S_k}$
  end if
  for $O_k \in L$ **do**
    if O_k *is a service provider* **then**
      $L_S \leftarrow Sp \bigcup \{O_k\}$
    else
      $L_O \leftarrow Sr \bigcup \{O_k\}$
    end if
  end for
  Compute $Ro_{ij}^{S_k}$ *//recommendation of* L_O *(equation 4)*
  Compute $Rs_{ij}^{S_k}$ *//recommendation of* L_S *(equation 5)*
  $R_{ij}^{S_k} \leftarrow Sp \times Rs_{ij}^{S_k} + (1 - Sp) \times Ro_{ij}^{S_k}$
  Send $R_{ij}^{S_k}$ *to the device* O_i
  return $R_{ij}^{S_k}$
end procedure

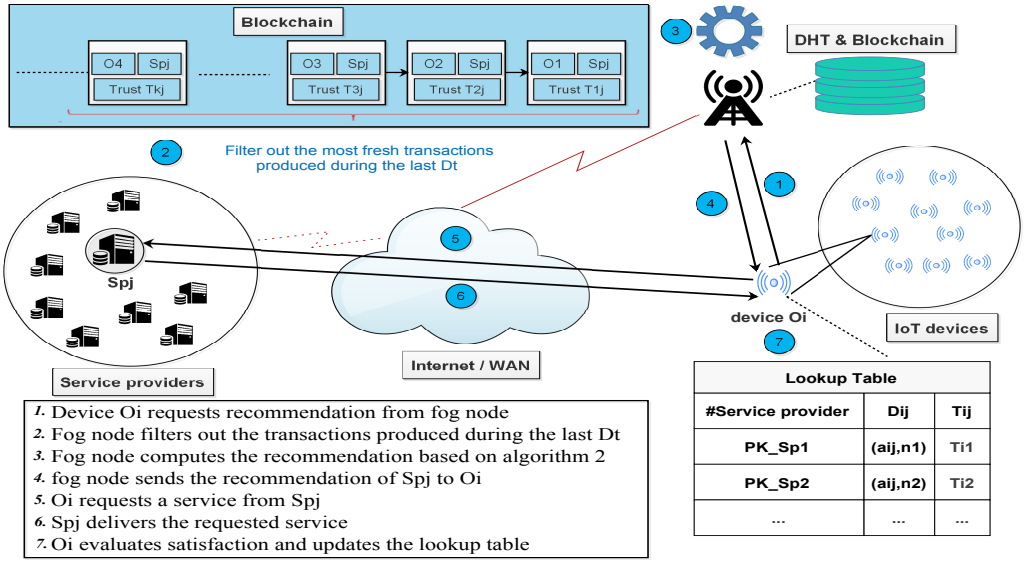


Figure 5.3 – work-flow of our trust management protocol ***BC-Trust***

providers don't participate in the process of block validation in order to save their computational power or break down the whole system. In order to avoid the problem of selfishness, the protocol uses a penalty in the computation of $R_{ij}^{S_k}(\Delta t)$. Therefore, fog node FN_l must check if Sp_j has participated in the process of validation during the last N generated blocks. In the case of non participation, the value $R_{ij}^{S_k}(\Delta t)$ is mutiplied by $1 - Pnl'$ where $Pnl' = 0.1$.

5.3.4 Countermeasure against cooperative attacks

The most common attacks that are performed in IoT based trust management systems are basically bad-mouthing and ballot stuffing attacks. In these attacks, malicious nodes tend to report bad recommendations for honest service providers or good recommendations for malicious ones. For more effectiveness, in general, this kind of attacks is cooperatively performed by several attackers in order to promote each other or target some honest service providers. Cooperative bad-mouthing and ballot-stuffing attacks involve great damages on the whole IoT system. Moreover, these attacks are very hard to detect and overcome, at least for the following reasons :

- ◇ **Risk of false negative** : when a group of nodes give bad recommendations for one particular node A repetitively, it is hard to say for sure whether this group of nodes is malicious or because the node A is really malicious.
- ◇ **Risk of false positive** : it could be possible in some cases that a group of nodes request periodically one particular service from one service provider (the case for example of data aggregation). Therefore, reporting periodically the same recommendations for one service provider (the aggregator node for example) does not necessarily mean that this group of nodes is conducting a cooperative attack against this service provider.

5.3.5 Block generation and consensus protocol

In this section, we propose a consensus algorithm for block generation and validation. Our main goal is to achieve a good tradeoff between security and scalability. Therefore, we propose an enhanced version of the protocol PBFT implemented in Tendermint [2]. Our consenus is a combination of PBFT (Practical Byzantine Fault Tolerance) and PoS (Proof of stack) algorithms. To ensure a high security level,

[2]Tendermint blockchain framework consists of a set of tools for achieving consensus on a distributed network, execution of smart contracts, and creation of blocks

we adopt as stack concept the trust value of each service provider. Thus, service providers with high trust values are more likely to be selected as validators.

The *PBFT* mechanism is capable of supporting up to f Byzantine *"faults"* if and only if the network is composed of at least $3f + 1$ nodes. The particular implementation of this algorithm works with a reasonably communication overhead in small networks. Unfortunately, the size of the necessary communication overhead increases drastically with the number of nodes, making the mechanism inefficient in large scale (bad scalability) [109].

To enhance the scalability of the *PBFT* protocol, we propose to choose a subset of $log(n)$ validators (instead on n) among the list of validators (service providers). Our consensus protocol is summarized in the following stages :

Validators selection stage : as soon as a block is formed, each block proposer (fog node) needs to randomly choose $log(n)$ validators from the set of n candidate service providers having a trust value higher than a treshold (0.7 in our protocol). The random selection is weighted to the trust values (stack) of the candidate service providers. So, each service provider Sp_j has a known probability of selection which correponds to the ratio $\frac{T_j}{\sum_{k=1}^{n} T_k}$. Thus, service providers with high trustworthiness have high probabity to be selected. Once the selection of validators is done, the block proposer sends a *validate* message to these selected validators and broadcasts the block to all blockchain nodes.

Pre-vote stage : once receiving the validate request, each validator checks the transactions forming the block. Mainly, it verifies the correctness of the signatures and the timestamp in each transaction. If the block is correct, the validator sends pre-vote messages to the other validators. Then, it waits for reception of $\frac{2}{3} * log(n)$ pre-vote messages.

Pre-commit stage : once the validator node receives $\frac{2}{3} * log(n)$ the pre-vote messages from the other validator nodes, it broadcast a pre-commit message to all the blockchain participants.

Commit phase : Each node in the blockchain (fog nodes and service providers) commit the block if it receives at least $2/3$ of $log(n)$ pre-commit messages.

In order to protect the protocol from fault byzantine errors in the propagation of the messages and blocks, we introduce *time-outs* after sending each message or block. Therefore, each node waits for a certain time-out related to each stage (pre-vote, pre-commit). In case the expiration of the time-out, the node deletes the block to be validated from its memory. Besides, all the exchanged messages are digitally signed by the private keys of message senders.

Algorithm 3 Countermeasure against cooperative attacks

Input : O_i : IoT device, Sp_j : IoT service provider
procedure ONLINE COUNTERMEASURE
    Init : $Sp \leftarrow \{\}$; $Sr \leftarrow \{\}$; $Nbocc[T] \leftarrow \{0\}$
    [t] $L \leftarrow$ *the most recent recommenders that reported transactions in* $[t - \Delta T, t]$
    $min_j(t) \leftarrow \min_{i \in L}\{T_{ij}^{S_k}(t)\}$
    $max_j(t) \leftarrow \max_{i \in L}\{T_{ij}^{S_k}(t)\}$
    if $max_j(t) - min_j(t) < Thr$ **then**
        $History \leftarrow$ *transactions produced during* $[t - n \times \Delta T, t]$
        for $O_k \in TopR$ **do**
            for $i := 1$ to n **do**
                if $T_{kj}^{S_k}(t - i\Delta T) \in History$ **then**
                $Nbocc[k] \leftarrow Nbocc[k] + 1$
                end if
            end for
        end for
        for $O_k \in L$ **do**
            if $\frac{Nbocc[k]}{n} > 0.8$ **then**
            $L \leftarrow L - \{O_k\}$
            end if
        end for
    end if
    $R_{ij}^{S_k} \leftarrow$ COMPUTEANDREPORTTRUST(L)
    return $R_{ij}^{S_k}$
end procedure

In this section, we propose a countermeasure solution to reduce the impact of cooperative attacks in the system. Our mitigation technique takes advantage of the history of the recommendations reported to the blockchain and works as follows : 1) Analyze the history of the received recommendations to detect if there is a cooperative attack. 2) Trigger a mitigation technique to eliminate the recommendations provided by the group of malicious nodes in the case of any eventual cooperative attack.

In order to deal with cooperative attacks, each fog node, during the computation of trust recommendations, executes the algorithm 2. This later works in the following steps :

1. First, the fog node selects all the recommendations for one particular IoT service provider S_k (as discussed previously in our protocol). Let $L = \{O_1, O_2, ..., O_l\} \cup \{Sp_1, Sp_2, ..., Sp_m\}$ be the subset of IoT devices and service providers that have recommended Sp_k during the last ΔT.
2. The fog node computes $min_k(t) = \min_{i \in L}\{T_{ik}^{S_k}(t)\}$ and $max_k(t) = \max_{i \in L}\{T_{ik}^{S_k}(t)\}$ which are respectively the minimum and the maximum of the recommendations provided by the devices of the list L. If the difference $\max_k(t) - min_k(t)$ is bigger than Thr, then the sevice provider Sp_k may

be subject of a cooperative attack. Indeed, having a large difference between $max_k(t)$ and $\min_k(t)$ is a suspicious situation. In fact, there is at least one node who did not grant a good recommendation to Sp_k contrariwise to others. Thus, one of these sub-groups is malicious (see from step 3 to step 7 in algorithm 2).

3. If an anomaly has been detected, the fog node consults the history of recommendations, available in the blockchain, which concern service provider Sp_k in the last n time slots ΔT. The fog node ignores the recommendation of each node who frequently appears in the history (see from step 7 to step 16 in algorithm 2).

5.4 Theoretical Analysis

In this section, we will study the convergence of our protocol ***BC-Trust*** with respect to the parameters of our system. In this theoretical analysis, we give lower and upper bounds of trust values obtained by our protocol under bad-mouthing and ballot stuffing attacks, showing that our protocol is highly resilient to these attacks. We recall in Table 2 the symbols used in this section.

Notation	Description
N	The number of IoT devices (service providers and service requesters)
sp	The rate of service providers
λ	The rate of honest devices
m	The minimum satisfaction value that can be attributed to one honest service provider Sp_j by honest device O_i
T_n^{ij}	The trust value attributed to service provider Sp_j by an honest IoT device O_i at time n
H, M	The subsets of honest and malicious devices respectively
$E_h(T_j)$	The mean trust value of honest service provider Sp_j, measured by all IoT devices
$E_m(T_j)$	The mean trust value of malicious service provider Sp_j, measured by all IoT devices

Tableau 5.2 – Table of symbols

Trust values $T_{ij}^{S_k}$ are updated each time that device O_i requests a service S_k from service provider Sp_j. We define $\{t_0, t_1, t_2, ...\}$ as an ordered set of instants when O_i requests S_k. Hence, each t_n refers to the n^{th} service request. For sake of simplicity, we consider only one service in what follows. Thus, we note $T_{ij}^{S_k}(t_n)$ by T_n^{ij}.

We define the sequence $\mathcal{S} = (T_n^{ij})_{n\in\mathbb{N}}$ by the set of trust values $T_{ij}(t)$, $t \in [t_n, t_{n+1}]$, $n \in \mathbb{N}$.

We define the sequence $\mathcal{R} = (R_n^{ij})_{n\in\mathbb{N}}$ by the set of recommendation values $R_{ij}(t)$ reported by fog nodes at each instant $t \in [t_n, t_{n+1}]$, $n \in \mathbb{N}$.

We define the sequence $\mathcal{D} = (D_n^{ij})_{n\in\mathbb{N}}$ by the set of direct observations $D_{ij}(t)$, $t \in [t_n, t_{n+1}]$, $n \in \mathbb{N}$.

5.4.1 Study of the convergence of $\mathcal{S} = (T_n^{ij})_{n\in\mathbb{N}}$

Given a network of N devices. For each honest device O_i and honest service provider Sp_j, we have :

$$\forall i, j \in \{1, ..., N\}, i \neq j, m \leq D_n^{ij} \leq 1 \tag{5.6}$$

Démonstration. From equation (5.2), we have :

$$D_n^{ij} = \frac{\alpha_{ij}}{n} = \frac{\sum_{t=1}^{n} S_{ij}(t)}{n} \tag{5.7}$$

Since Sp_j is a honest service provider, the satisfaction value $S_{ij}(t)$ at time t is at least equal to m and at most equal to *1*. Therefore, we obtain from equation (5.7) :

$$m \leq D_n^{ij} \leq 1$$

□

Given a network of N devices with a rate *sp* of service providers and λ ($\lambda > 0$) the rate of honest devices. Under bad-mouthing attacks, for each honest device O_i and honest service provider Sp_j such that $i, j \in \{1, ..., N\}, i \neq j$, we have :

$$\forall n \geq 1, R_n^{ij} \geq \lambda_m \times T_{n-1}^{min} \tag{5.8}$$

where

$$T_n^{min} = \min\{T_n^{kj}\ , k, j \in \{1, ..., N\}, \text{ and } O_k, O_j \in H\}$$
$$\lambda_m = sp + (1 - sp) \times \lambda$$

Démonstration. From equation (5.3), we have :

$$R_n^{ij} = sp \times Rs_{ij}(n) + (1 - sp) \times Ro_{ij}(n)$$

Given a set $L^{'} = L_S^{'} \cup L_O^{'}$ composed of two subsets $L_S^{'}$ (service providers) and $L_O^{'}$ (IoT devices) that have recommended Sp_j. We distinguish two cases for each subset :

1) For the subset $L_O^{'}$, recommendation $Ro_{ij}(n)$ is expressed as follows :

$$Ro_{ij}(n) = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'}} T_{n-1}^{kj} = R_H + R_M$$

where :

$$R_H = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap H} T_{n-1}^{kj}$$
$$R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj}$$

In what follows, we study the lower bounds of (R_H and R_M).

Case 1 : the sum R_H

$$R_H = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap H} T_{n-1}^{kj}$$

By definition, for each $n \geq 0$, we have :

$$\forall i, j \in \{1, ..., N\}, T_n^{kj} \geq T_n^{min}$$

Hence, given that λ is the rate of honest devices in $L_O^{'}$, we can simplify R_H as follows :

$$\forall n \geq 1, R_H \geq \lambda \times T_{n-1}^{min} \tag{5.9}$$

Case 2 : the sum R_M

Under bad-mouthing attacks, malicious devices report bad recommendations T_n^{ij} which are equal to 0 in the worst case. Therefore :

$$R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj}$$
$$T_n^{ij} \geq 0 \implies R_M = \frac{1}{|L_O^{'}|} \times \sum_{k \in L_O^{'} \cap M} T_{n-1}^{kj} \geq 0 \tag{5.10}$$

From inequalities (5.9) and (5.10), we have :

$$Ro_{ij}(n) \geq \lambda \times T_{n-1}^{min} \tag{5.11}$$

2) For the subset $L_S^{'}$, recommendation $Rs_{ij}(n)$ is expressed as follows :

$$Rs_{ij}(n) = \sum_{k \in L_S^{'}} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S^{'}} T_{n-1}^{ik}} \times T_{n-1}^{kj}$$

Since our protocol considers only service providers that device O_i grant them a trust value $T_{n-1}^{ik} > Threshold$ (refer to Section 5.3.3.5 about the computation of $Rs_{ij}^{S_k}(\Delta t)$). Hence, the set $L_S^{'}$ will be reduced to the new set L''_S :

$$L''_S = L'_S - \{Sp_k, T_{ik} \leq Threshold\}$$

$$Rs_{ij}(n) = \sum_{k \in L_S''} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S''} T_{n-1}^{ik}} \times T_{n-1}^{kj}$$

Given $T_{n-1}^{kj} \geq T_{n-1}^{min}$ for each $n \geq 1$, we have :

$$Rs_{ij}(n) \geq \sum_{k \in L_S''} \frac{T_{n-1}^{ik}}{\sum_{k \in L_S''} T_{n-1}^{ik}} \times T_{n-1}^{min}$$

Let $a_k = \frac{T_{n-1}^{ik}}{\sum_{k \in L_S''} T_{n-1}^{ik}}$. We can easily check that $a = \sum_{k \in L_S''} a_k = 1$. Therefore, the sum $Rs_{ij}(n)$ can be simplified as :

$$Rs_{ij}(n) \geq T_{n-1}^{min} \tag{5.12}$$

From inequalities (5.11) and (5.12), we find out :

$$R_n^{ij} \geq \lambda_m \times T_{n-1}^{min}$$

where $\lambda_m = sp + (1 - sp) \times \lambda$ □

5.4.1.1 Resiliency against malicious attacks

Given a network of N devices with sp the rate of service providers and λ ($\lambda > 0$) the rate of honest devices. Under bad-mouthing attacks, for each honest device O_i

and honest service provider Sp_j, such that $i, j \in \{1, ..., N\}, i \neq j$, we have :

$$T_h = \lim_{n\to\infty} T_n^{ij} \geq \frac{m \times \beta}{1 - \alpha - \gamma \times \lambda_m} \tag{5.13}$$

where : $\lambda_m = sp + (1 - sp) \times \lambda$

Démonstration. Given O_i and Sp_j are honest. By definition, we have : $\forall n \geq 0$, $T_n^{ij} \geq T_n^{min}$. Thus, we only need to study the convergence of the sequence $(T_n^{min})_{n\in\mathbb{N}}$. Based on the result of lemma 1 and lemma 2, we have :

$$T_n^{min} \geq \alpha \times T_{n-1}^{min} + \beta \times m + \gamma \times \lambda_m \times T_{n-1}^{min}$$

Hence, we get :

$$\lim_{n\to\infty} T_n^{min} \geq (\alpha + \gamma \times \lambda_m) \times \lim_{n\to\infty} T_{n-1}^{min} + \beta \times m$$

Therefore :

$$\lim_{n\to\infty} T_n^{min} \geq \frac{\beta \times m}{1 - \alpha - \gamma \times \lambda_m}$$

Since $\forall n \geq 0$, $T_n^{ij} \geq T_n^{min}$, we have :

$$\lim_{n\to\infty} T_n^{ij} \geq \lim_{n\to\infty} T_n^{min}$$

Therefore,

$$T_h = \lim_{n\to\infty} T_n^{ij} \geq \frac{\beta \times m}{1 - \alpha - \gamma \times \lambda_m}$$

□

Given a network of N devices with sp the rate of service providers and λ ($\lambda > 0$) the rate of honest devices. Under ballot-stuffing attacks, for each honest device O_i and malicious service provider Sp_j, such that $i, j \in \{1, ..., N\}, i \neq j$, we have :

$$T_m = \lim_{n\to\infty} T_{ij}(n) \leq 1 - \frac{m \times \beta}{1 - \alpha - \gamma \times \lambda_m} \tag{5.14}$$

where : $\lambda_m = sp + (1 - sp) \times \lambda$

Démonstration. The proof is similar to the proof of proposition 1. □

Theorem 5.1. *Given a network of N devices with sp the rate of service providers and λ (λ > 0) the rate of honest devices. Under bad-mouthing attacks, the mean*

trust $E_h(T_j)$ of honest service providers measured by all network devices is :

$$E_h(T_j) \geq \lambda \times T_h \geq \frac{\lambda \times \beta \times m}{1 - \alpha - \gamma \times \lambda_m} \tag{5.15}$$

Démonstration. Let Sp_j be a honest service provider, we have :

$$\begin{aligned} E_h(T_j) &= \lim_{n\to\infty} \frac{1}{L} \sum_{i=1}^{L} T_n^{ij} \\ &= \lim_{n\to\infty} \frac{1}{L} \sum_{i=1}^{L} [Pr(O_i \text{ is honest}) \times T_n^{ij} + \\ &Pr(O_i \text{ is malicious}) \times T_n^{ij}] \\ &= \lim_{n\to\infty} \lambda \times T_n^{ij} + (1 - \lambda) \times \varepsilon \end{aligned}$$

Since $\varepsilon \geq 0$, the worst value of ε given by bad-mouthing attacker is 0. Hence, we have :

$$\begin{aligned} E_h(T_j) &\geq \lim_{n\to\infty} \lambda \times T_n^{ij} + (1 - \lambda) \times 0 \\ &\geq \lambda \times T_h \geq \frac{\lambda \times \beta \times m}{1 - \alpha - \gamma \times \lambda_m} \end{aligned}$$

□

Theorem 5.2. *Given a network of N devices with sp the rate of service providers and λ ($\lambda > 0$) the rate of honest devices. Under ballot-stuffing attacks, the mean trust $E_m(T)$ of dishonest service providers measured by all network devices is :*

$$E_h(T_j) \leq \lambda \times T_m + 1 - \lambda \leq 2 - \lambda - \frac{\lambda \times \beta \times m}{1 - \alpha - \gamma \times \lambda_m} \tag{5.16}$$

Démonstration. Let Sp_j be a dishonest service provider, we have :

$$\begin{aligned} E_h(T_j) &= \lim_{n\to\infty} \frac{1}{N} \sum_{i=1}^{N} T_n^{ij} \\ &= \lim_{n\to\infty} \frac{1}{N} \sum_{i=1}^{N} [Pr(O_i \text{ is honest}) \times T_n^{ij} + \\ &Pr(O_i \text{ is malicious}) \times T_n^{ij}] \\ &= \lim_{n\to\infty} \lambda \times T_n^{ij} + (1 - \lambda) \times \varepsilon \end{aligned}$$

Since $\varepsilon \leq 1$, the best value of ε given by a ballot-stuffing attacker is 1. Hence, we

have :

$$E_h(T_j) \leq \lim_{n\to\infty} \lambda \times T_n^{ij} + (1-\lambda) \times 1$$
$$\leq \lambda \times T_m + 1 - \lambda \leq 2 - \lambda - \frac{\lambda \times \beta \times m}{1 - \alpha - \gamma \times \lambda_m}$$

□

5.5 Performances evaluation

In this section, we evaluate the effectiveness, resiliency and the benefits of our proposed ***BC-Trust*** approach through different experiments. We demonstrate how our experimental results fit with the theoretical analysis presented in the previous section. Basically, we performed three initial experiments. The first one evaluates the effectiveness of our solution in terms of convergence time with respect to different parameters (α, β, γ). The second one evaluates the resiliency of our protocol against bad-mouthing and ballot-stuffing attacks. Finally, we evaluate the effectiveness of our countermeasure approach against cooperative attacks. Table 5.3 summarizes the main setting parameters related to our experiments.

parameters	**values**
Number of IoT devices (N)	*100*
Rate of service providers (sp)	*20%*
Default values of (α, β, γ)	$\alpha = \beta = \gamma = \frac{1}{3}$
Number of services	*1*
Number of criteria	*5*
Δt	*5* seconds

Tableau 5.3 – Test settings

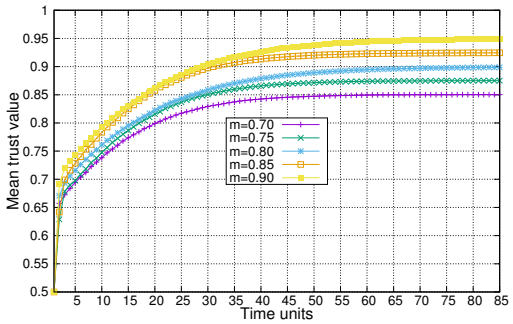


Figure 5.4 – Mean trust of honest service providers w.r.t. m.

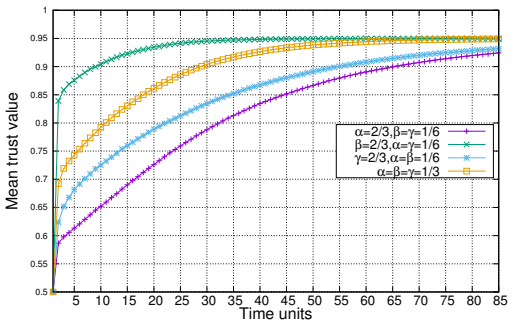


Figure 5.5 – Mean trust of honest service providers w.r.t. α, β, γ.

5.5.1 Evaluation of the convergence of our protocol

The first bunch of experiments aims to measure the convergence time of our protocol, and to study the impact of parameters α, β, γ and m on both convergence value and time. In order to get a clear view on the behavior of our protocol, this first sequence of experiments is done in a safe area where all the nodes are assumed to be honest.

Figure 5.4 illustrates the evolution of the mean trust value of all the service providers seen by all IoT devices during the lifetime of the simulation. We clearly notice that the limit trust value depends on the parameter m (the minimum satisfaction level that can be attributed to honest service providers). Besides, this limit trust value converges to the value $\frac{m+1}{2}$ which exactly fits with the result of proposition 1. However, we notice that the convergence time does not depend on the parameter m. Indeed, even with two different m values, our protocol converges to almost the same time (convergence after about *70* time units).

Figure 5.5 depicts the mean trust value with respects to the parameters : α, β, γ. As we notice, these three parameters have an impact only on the convergence time of the mean trust value. However, these parameters do not affect the convergence value. Moreover, parameter β (the weight of direct observation) enhances significantly the convergence time compared to parameters α and γ. Indeed, with $\beta = \frac{2}{3}$ and $\alpha = \gamma = \frac{1}{6}$, the convergence time is reduced to around *40* time units, whereas with smaller value of β (i.e. $\beta = \frac{2}{3}$) the convergence time is significant (> 80 time units).

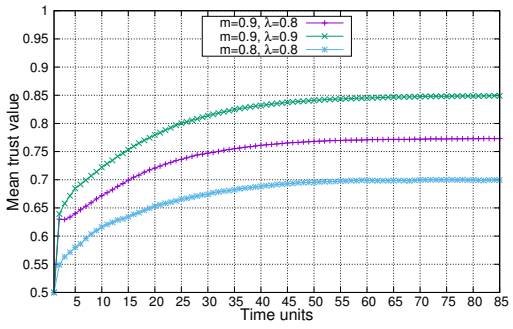


Figure 5.6 – Mean Trust under bad-mouthing attacks.

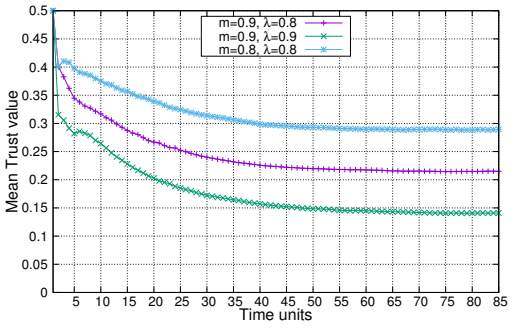


Figure 5.7 – Mean Trust under ballot-stuffing attacks.

5.5.2 Effectiveness of our protocol against malicious attacks

After studying the behavior of our protocol in normal circumstances, we evaluate in what follows its effectiveness under malicious attacks. We mainly focus on two kind of attacks : bad-mouthing and ballot-stuffing attacks.

As illustrated in Figure 5.6, the robustness of our protocol against bad-mouthing attacks has been evaluated with respect to the rate of honest nodes (λ). To do so, we vary the rate of honest nodes λ and the parameter m while the other parameters are kept constant and take their default values as shown in Table 3. Overall, we notice that the limit of mean trust value for honest service providers is reduced compared the result obtained in the case where there is no attack. As trivially expected, this limit value decreases with respect to the rate of malicious nodes ($1 - \lambda$). However, even with 20% of malicious nodes and $m = 0.9$, our protocol converges to a mean trust value which exceeds 0.75. This is due to our strategy of the computation of recommendations which favors trust values coming from honest nodes. Moreover, it is straightforward to see that the limit mean trust value is always bigger than the lower bound obtained in the theoretical analysis (see proposition 1) with a small gap which is up to 4%.

On the other side, we evaluated the impact of ballot stuffing attacks on our protocol by varying the rate of honest nodes λ. Figure 5.7 illustrates the mean trust value of malicious service providers (evaluated by honest nodes) with respect to different values of λ and m. Despite the presence of significant malicious nodes ($1-\lambda = 20\%$), we notice that the limit trust value is still small and reflects a correct reputation on these malicious nodes. Moreover, it is worth nothing that the theoretical analysis discussed in proposition 2 (upper bound limit of mean trust value of malicious nodes under ballot-stuffing attacks) are confirmed in the Figure 5.7.

Overall, the above results exhibit that ***BC-Trust*** shows its effectiveness and robustness to deal with bad-mouthing and ballot-stuffing attacks.

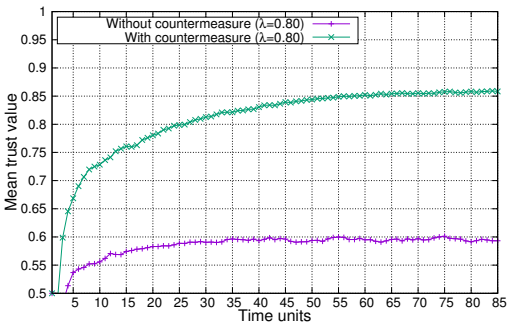


Figure 5.8 – Trust under cooperative bad-mouthing attacks.

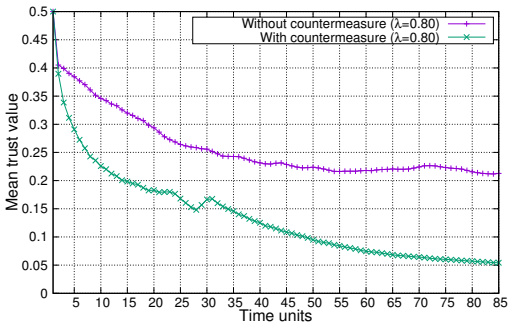


Figure 5.9 – Trust under cooperative ballot-stuffing attacks

5.5.3 Effectiveness against cooperative attacks

In order to evaluate the efficiency and robustness of the countermeasure approach of our solution against cooperative attacks, we performed a set of experiments using the following scenarios :

- ◇ **Scenario 1** : we perform a cooperative bad-mouthing attack in which, all the malicious nodes target one service provider Sp_j and periodically report bad recommendations about it. The other honest nodes behave naturally, where they choose the service provider Sp_j randomly among other service providers and report real recommendations about it.
- ◇ **Scenario 2** : we perform a cooperative ballot-stuffing attack in which all malicious nodes periodically report good recommendations about the target malicious service provider Sp_k, whereas honest nodes provide real recommendations about Sp_k.

In both scenarios, we vary the rate of malicious nodes $(1 - \lambda)$ to show the resiliency of our approach.

In figure 5.8, we show the evolution of mean trust value of target service provider Sp_j under bad-mouthing attacks. We notice that our online countermeasure algorithm significantly reduces the effect of collaborative attacks compared to the case where there is no countermeasure. Indeed, despite the presence of $1 - \lambda = 20\%$ of malicious nodes conducting bad-mouthing attacks, the mean trust value of service provider Sp_j reaches the limit value 0.87. This last is significantly bigger than the reached limit value in the case where there is no countermeasure (0.59).

Similarly, in Figure 5.9, we show the results of experiments conducted on *BC-Trust* with the presence of ballot stuffing attacks by varying the rate λ. Our countermeasure algorithm also mitigates the trust computation process performed by fog nodes and it significantly reduces the impact of cooperative ballot-stuffing attacks. Indeed, with a rate $1 - \lambda = 20\%$ of malicious nodes, the limit trust value of the target malicious node reaches the value 0.05. This value is small comparing to the limit value 0.27 obtained in the case where there is no countermeasure.

5.5.4 BC-Trust vs Existing solutions

In table 5.4, we show a comparison of our solution and two other solutions (presented in related works section) in terms of storage, computation and communication overhead. We notice that our protocol ***BC-Trust*** reduces storage related to trust

Tableau 5.4 – comparison in terms of trust evaluation cost

	Storage	Computation			Communication
		#Mult	#Add	#Exp	
[33]	$O(N^2)$	$O(N)$	$O(N)$	0	$O(N)$
[105]	$O(F)$	$O(1)$	$O(F)$	0	$O(F)$
Ours	$O(N)$	$O(1)$	$O(1)$	0	$O(1)$

In this table, we provide comparison in terms of computation, storage and communication. Note that N is the number of devices and F is the average number of friends in the social graph as presented in the work of Nitti et al. [105].

values compared to other solutions. Indeed, in our protocol, IoT devices store only trust data related to service providers which are basically its own direct observations. The amount of this data is at most equal to $8 \times N$ which depends linearly on the number of IoT devices N if we assume that trust values are encoded on 4 bytes. However, in other approaches, the storage overhead depends quadratically on the number of IoT devices N since each device must keep the recommendation of other nodes against each service provider. Moreover, contrary to other approaches, ***BC-trust*** reduces computation overhead (few additions and multiplication) which is independent of the number of IoT devices. Finally, the communication overhead, measured as the amount of data exchanged during Δt, is also reduced in our protocol. Indeed, IoT devices need to exchange only with fog nodes to get recommendation about one service provider, whereas in other solutions IoT devices must exchange the recommendations between each others.

Note that to reach the consensus at blockchain layer, the communication complexity of fog nodes and service providers is $O(((sp \times N + l) * log(sp \times N))$, where sp is the percentage of service providers, l is the number of fog nodes in the architecture. This communication complexity is better than the complexity of PBFT consenus protocols, which work with $O(n^2)$ for each validation event. We believe that this complexity is reasonable when it comes to powerful service providers and fog nodes. Regarding the storage complexity, the storage of blockchain transactions can be optimized by maintaining only the most recent transactions used in the computation of recommendation values and replace the old blocks by their merkle hash [102].

We present in Table 5.5 a qualitative comparison of our proposal with some previously presented related works. Our solution is very convenient with high mobility scenarios and resists against node failures. Furthermore, our solution is QoS-aware protocol since it reduces the latency during the computation of trust values and allows IoT devices to filter out service providers with respect to some QoS metrics thanks to fine-grained based service property. Contrary to other approaches, ***BC-Trust*** introduces other original properties such as global view of

	Scalability	Mobility	Node-failure	QoS	Convergence time	Global view
[33]	-	-	+	-	-	-
[105]	+	+	-	-	+	+
[75]	-	-	+	-	+	-
[31]	-	-	-	-	+	-
[118]	-	-	+	+	+	-
[37]	+	-	-	+	+	-
[32]	+	-	-	+	+	-
Ours	+	+	+	+	+	+

Tableau 5.5 – Comparison between trust management protocols

trustworthiness information and scalability support which are very important in IoT.

In addition, we were interested in comparing our approach with the work of chen et al. [33] in terms of successful detection rate under high mobility scenario with the presence of malicious nodes. For this aim, we adopted the same mobility scenario based on random walk model. In this model, we generated initial random positions for 100 devices in a range of 100x100m. Each round (5 seconds), all devices must change their positions based on the random walk model and try to discover the service providers in their neighborhoods. We considered a communication range equal to 20m for all devices. Each device requests the same service from all its neighbors and performs the trust management protocol. We fixed the number of rounds by 100 rounds. We assume that there are sufficient fog nodes that can cover the whole region. We varied the percentage of malicious nodes from 0% to 30% and we observed the rate of successful detection. Fig. 9 shows that our protocol outperforms the protocol of chen et al. in terms of successful detection rate even with the presence of 30% of malicious nodes. BC-Trust achieves a high performance with 100% of successful detection rate if $\lambda \leq 0.25$ and 90% if $\lambda = 0.3$. These results are achieved thanks to our 2-layer architecture that ensures a global view of trustworthiness in the whole network with only few exchanges, and so it deals very efficiently with high mobility scenarios.

5.5.5 Blockchain scalability evaluation

We implemented our protocol using the framework tendermint [24]. Tendermint framework is a tool used to build a private and consortium blockchains based on PBFT consensus protocol. It allows developers to build any application on the top of the underlying PBFT protocol using Application Blockchain Interface (ABCI). We developed our protocol using Python as programming langage. BC-Trust was

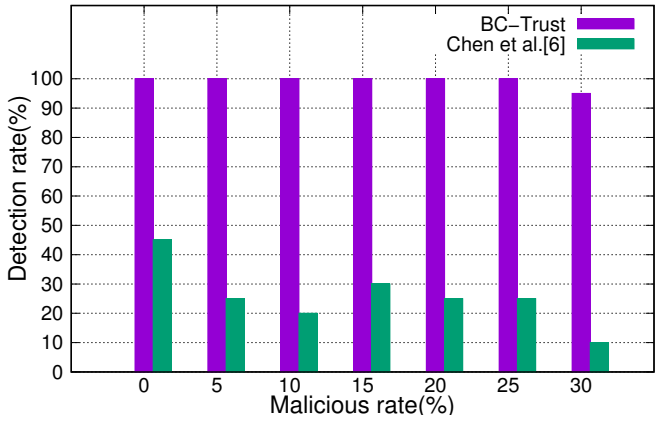


Figure 5.10 – Successful detection rate under high mobility

evaluated using 20 docker nodes, which were created locally in Ubuntu 18.04 machine with 16GB of RAM and Intel(R) Core(TM) i5-6200 CPU. Our evaluation was carried out on 1, 5, 10, 15 and 20 nodes where all of them act as validator nodes. We performed two scenarios of test to see the impact of the transactions rate and the number of validator nodes on transaction validation latency.

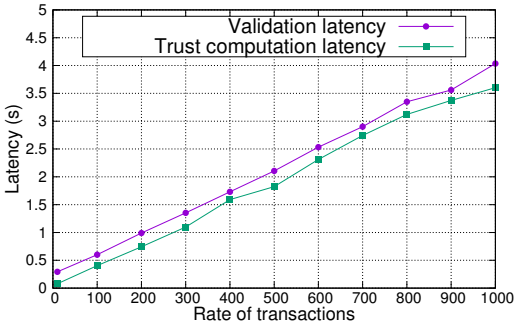


Figure 5.11 – Validation latency w.r.t. transactions rate

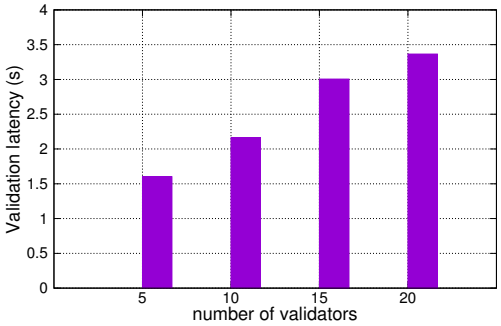


Figure 5.12 – Validation latency w.r.t. the number of validators

5.5.5.1 Impact of transactions' rate

In the following, we study the performance of BC-Trust with respect to the number of transactions received in parallel. In this scenario, we fixed the number of validators to 1 and we varied the number of transactions between 1 to 1000 transactions. We measured both the validation time of all the transactions and the time needed to compute recommendation values from the trust values stored in the blockchain (query time). As shown in Fig. 10, the latency of validation and request operations increase linearly with respect to the number of transactions and queries respectively. Overall, the results are very satisfactory in terms of scalability, given that the time that takes to validate 1000 transactions is 4s.

5.5.5.2 Impact of the number of validator nodes

In order to evaluate the impact of the number of validator nodes in our solution, we used the Tendermintâs PBFT implementation as consensus method. In these experiments, we varied the number of validator nodes by creating several docker containers that execute the same ABCI application (our BC-Trust). We fixed the number of parallel transactions by 100 transactions for all the experiments. Fig. 11 shows the validation time with respect to the number of validator nodes. As expected, the validation latency increases accordingly. The maximum delay is roughly 3.3 seconds for 20 validator nodes. Note that Tendermint uses all validators to validate each block and it is mandatory that at least 2/3 of nodes commit the block to reach the consensus. Therefore, it is straightforward that, if we choose to reduce the number of validator nodes to $log(n), n \in \{5, 10, 15, 20, ...\}$ as we have discussed in section 5.3.5, we can obtain better performance. This does not affect the security of the protocol as the selection of validator nodes is random and it is based on the trust level of each candidate service provider.

5.6 Conclusion

In this chapter, we proposed a new decentralized trust management protocol for IoT in fog computing architecture. Our protocol is distributed and each IoT object can assess trustworthiness of service providers and share it among IoT devices in a scalable way. Based on blockchain technology, our protocol offers a global view on the trustworthiness of each service provider in the architecture. Moreover, contrary to most existing works, our proposal deals efficiently with high mobility scenarios thanks to blockchain technology. Besides, we demonstrated through experiments the resiliency and robustness of our solution in front of malicious attacks. Then, we showed that our solution outperforms the existing ones, especially in terms of saving computation and storage resources. In addition, we confirmed our experimental result through an advanced theoretical analysis about the convergence of trust values under different malicious attacks. Furthermore, we shed the light on cooperative attacks where we proposed an efficient countermeasure based on the analysis of recommendations' history reported by IoT devices to the blockchain.

For future work, we plan to extend our proposed mitigation approach by developing more efficient offline algorithms for malicious nodes detection using machine learning techniques.

Conclusions and future directions

The evolution of the Internet, wireless communication technologies, as well as the development of new smart devices allowing the collection of environmental data such as temperature, motion, pressure etc. allows the emergence of the Internet of Things (IoT) paradigm. IoT has a wide variety of applications in many fields such as healthcare, industry, logistics, etc. This technology is considered as enabling technology for Systems of Systems (SoS).

In this thesis, we focused on the security of Internet of Things for Systems of Systems. The aim of our work is to make IoT based SoS more secure, and hence more trusworthy to build and execute sensitive applications. For this purpose, we investigated different security problems that threaten the IoT by using cryptographic techniques and new emerging technologies such as blockchain.

First, we were interested in the first step of the thesis on the study of the state of the art of security solutions In IoT. For this purpose, we surveyed security solutions proposed for IoT applications. We classified the different IoT applications by identifying their security requirements and their inherent challenges. Then, we discussed the IoT solutions dealing with confidentiality, privacy and availability, which are based on traditional cryptographic solutions and new emerging technologies such as Software Defined Networking (SDN) and blockchain.

Next, on the first hand, we focused on the access control and authentication issues in the context of remote control of IoT actuators. We tackled the problem of remote secure control of IoT actuators. For this purpose, we proposed a distributed lightweight fine-grained access control protocol based on Ciphertext Policy Attribute Based Encryption mechanism and one-way hash chain. We demonstrated through formal security analysis based on AVISPA tool that our scheme is secure against various attacks. Moreover, we demonstrated through simulations the scalability and the efficiency of our solution, which saves substantially energy consumption and computation costs.

On the other hand, we focused mainly on the application of blockchain technology

to address security problems in the Internet of Things. We combined blockchain and fog computing paradigms to propose new solutions for the authentication and trust management in IoT.

We tackled the problem of authentication in heterogeneous IoI systems. We considered in particular the mutual authentication problem between IoT devices and fog nodes. The main aim is to develop a new efficient protocol that takes into consideration the resource limitation of IoT devices. Moreover, the mutual authentication should be performed only at the edge of the network without referring to external servers and cloud computing layer. To achieve this goal, we have proposed an efficient mutual authentication scheme, named MASFOG, which is based on public blockchain and secret sharing techniques. We use the blockchain, maintained at the fog layer, to allows IoT devices to authenticate fog nodes. In addition, it allows fog nodes to establish mutual authentication with each others. We used also secret sharing technique to authenticate lightweight devices. We showed through experimentations the efficiency of our authentication scheme which provides a low overhead in terms of storage capacity and computation.

We were also interested on the problem of trust management based on the blockchain technology. After studying the existing solutions that deal with trust management in IoT, we found out that they are not scalable enough when it comes to massively distributed systems such as IoT. Moreover, other questions still arise on how trust information is disseminated and shared in a scalable way among different IoT objects in order to speed up the process of trust computation and make it more accurate and secure. To deal with these issues, we have proposed a new scalable trust management protocol, named *BCTrust*, that allows IoT devices to accurately assess and share trust recommendations about other devices in a scalable way without referring to any pre-trusted entity. We have confirmed the efficiency and scalability of our protocol through theoretical analysis and extensive simulations.

6.1 Open Issues and Future Directions

Hereafter, we shed the light on some perspectives and future directions relating to security in IoT :

- ◇ Regarding our authentication scheme, *MASGOG*, there are still some important issues concerning the revocation problem in fog computing. *MASFOG* is based on the blockchain technology where there is no central entity that can manage the whole system. Therefore, the decentralization

nature of *MASFOG* makes the revocation process very hard to achieve since it is necessary to update the cryptographic keys of the revoked nodes in the whole blockchain. Beside, we need to develop efficient revocation mechanism that should be as less expensive as possible. This task is also very tricky as use in MASFOG design a public blockchain based on Proof of Work (PoW). As future work, we intend to extensively address the problem of revocation in *MASFOG*.

- ◇ Regarding **BC-Trust**, we plan to enhance this protocol by developing more efficient algorithms for malicious node detection (malicious fog nodes or IoT devices) based on trust data maintained on the blockchain and reported by devices. In this direction, it could be interesting to explore and use machine learning approaches to improve the process of malicious node detection. Beside, in future, we will consider other trust models such as Bayesian and probabilistic models to propose more efficient and robust trust management protocols.

- ◇ The need to develop context-awareness solutions in the environment in which the smart objects and humans evolve is a fundamental approach to address the security in highly distributed and dynamic environments such as IoT. For example, in the context of trust management, it could be interesting to take into account information like : number of surrounding objects, the energy level of service providers and devices, the geographic localization of devices, etc. in order to make relevant decisions about the trustworthiness of service providers. We can exploit the context in order to develop efficient authentication approaches. This goal could be achieved by taking into account the physical locations of devices and their activities to make decisions about the authenticity of devices without referring to credential keys and heavy cryptographic approaches.

- ◇ As we have shown in this thesis, the blockchain technology can offer a high level of security of IoT transactions in many applications. A lot of researchers believe that this technology could change the world of IoT in terms of security and services. However, this technology is still just in its early stage, and therefore a lot of research must be devoted in order to optimize some of its important features such as consensus mechanisms used to validate transactions. In the context of IoT, it is important to develop a consensus approach that establish a tradeoff between security and efficiency. In addition, blockchain solutions suffer also from some privacy issues and are still vulnerable to anonymity attacks [43]. In

the Blockchain, pseudonyms are used as users' identifiers to send and receive transactions. In fact, the pseudonym does not ensure the privacy of transactions, it is possible to de-anonymize a user and disclose its identity by analyzing transactions' inputs and outputs. Therefore, it should be interesting to take advantage of the state of art in privacy based blockchain and public ledger solutions and adapt them in the field of IoT.

References

[1] M. M. Abdel-Aziz and A. T. Abdel-Hamid. Hardware low power implementation of attribute-based encryption. In *2016 28th International Conference on Microelectronics (ICM)*, pages 273–276, Dec 2016.

[2] R. Acker and M. Massoth. Secure ubiquitous house and facility control solution. In *2010 Fifth International Conference on Internet and Web Applications and Services*, pages 262–267. IEEE, 2010.

[3] M. T. Ahammed and P. P. Banik. Home appliances control using mobile phone. In *2015 International Conference on Advances in Electrical Engineering (ICAEE)*, pages 251–254, Dec 2015.

[4] A. Ahmed and E. Ahmed. A survey on mobile edge computing. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–8, Jan 2016.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things : A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4) :2347–2376, 2015.

[6] S. Alanazi, J. Al-Muhtadi, A. Derhab, K. Saleem, A. N. AlRomi, H. S. Alholaibah, and J. J. Rodrigues. On resilience of wireless mesh routing protocol against dos attacks in iot-based ambient assisted living applications. In *17th International Conference on E-health Networking, Application & Services (HealthCom)*, pages 205–210. IEEE, 2015.

[7] A. Alcaide, E. Palomar, J. Montero-Castillo, and A. Ribagorda. Anonymous authentication for privacy-preserving iot target-driven applications. *Computers & Security*, 37 :111–123, 2013.

[8] M. Ali, J. C. Nelson, R. Shea, and M. J. Freedman. Blockstack : A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference*, pages 181–194, 2016.

[9] M. Ambrosin, M. Conti, and T. Dargahi. On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, pages 49–54. ACM, 2015.

[10] M. Atzori. Blockchain-based architectures for the internet of things : A survey. *Available at SSRN : https ://ssrn.com/abstract=2846810*, January 2017.

[11] A. Aydeger, N. Saputro, K. Akkaya, and M. Rahman. Mitigating crossfire attacks using sdn-based moving target defense. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pages 627–630, Nov 2016.

[12] A. Bahga and V. K. Madisetti. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10) :533, 2016.

[13] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers : Authentication in ad-hoc wireless networks. In *NDSS*. Citeseer, 2002.

[14] F. Bao, R. Chen, M. Chang, and J.-H. Cho. Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. *IEEE Trans. on network and service management*, 9(2) :169–183, 2012.

[15] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy, 2007. SP'07*, pages 321–334. IEEE, 2007.

[16] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, May 2007.

[17] K. Biswas and V. Muthukkumarasamy. Securing smart cities using blockchain technology. In *2016 IEEE 18th International Conference on High*

Performance Computing and Communications ; IEEE 14th International Conference on Smart City ; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pages 1392–1393. IEEE, Dec 2016.

[18] R. Blom. An optimal class of symmetric key generation systems. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 335–338. Springer, 1984.

[19] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Annual International Cryptology Conference*, pages 213–229. Springer, 2001.

[20] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[21] S. Bouzefrane, A. F. B. Mostefa, F. Houacine, and H. Cagnon. Cloudlets authentication in nfc-based mobile computing. In *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pages 267–272. IEEE, 2014.

[22] S. Brief. Sdn security considerations in the data center, 2013.

[23] P. J. Bruening and K. K. Waterman. Data tagging for new information governance models. *IEEE Security Privacy*, 8(5) :64–68, Sept 2010.

[24] E. Buchman. *Tendermint : Byzantine fault tolerance in the age of blockchains*. PhD thesis, 2016.

[25] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson. Flow based security for iot devices using an sdn gateway. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Future Internet of Things and Cloud (FiCloud),*, pages 157–163. IEEE, July 2016.

[26] C. Cachin and M. Vukolić. Blockchain consensus protocols in the wild. *arXiv preprint arXiv :1707.01873*, 2017.

[27] A. A. Cárdenas, S. Amin, and S. Sastry. Research challenges for the security of control systems. In *HotSec*, 2008.

[28] M. Castro, B. Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[29] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings. 2003 Symposium on Security and Privacy, 2003*, pages 197–213. IEEE, 2003.

[30] I. Chatzigiannakis, A. Pyrgelis, P. G. Spirakis, and Y. C. Stamatiou. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In *2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 715–720. IEEE, 2011.

[31] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang. Trm-iot : A trust management model based on fuzzy reputation for internet of things. *Comput. Sci. Inf. Syst.*, 8(4) :1207–1228, 2011.

[32] I. R. Chen, F. Bao, M. Chang, and J. H. Cho. Trust management for encounter-based routing in delay tolerant networks. In *IEEE Global Telecommunications Conf. GLOBECOM*, pages 1–6, Dec 2010.

[33] I. R. Chen, F. Bao, and J. Guo. Trust-based service management for social internet of things systems. *IEEE Trans. on Dependable and Secure Computing*, 13(6) :684–696, Nov 2016.

[34] M. Chen, Y. Hao, Y. Li, C.-F. Lai, and D. Wu. On the computation offloading at ad hoc cloudlet : architecture and service modes. *IEEE Communications Magazine*, 53(6) :18–24, 2015.

[35] R. Chen, J. Guo, and F. Bao. Trust management for soa-based iot and its application to service composition. *IEEE Trans. on Services Computing*, 9(3) :482–495, 2016.

[36] W. Chen. An ibe-based security scheme on internet of things. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, volume 3, pages 1046–1049. IEEE, 2012.

[37] J.-H. Cho, A. Swami, and R. Chen. Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks. In *Int. Conf. on Computational Science and Engineering*, volume 2, pages 641–650. IEEE, 2009.

[38] T. Choi, H. B. Acharya, and M. G. Gouda. The best keying protocol for sensor networks. *Pervasive and Mobile Computing*, 9(4) :564–571, 2013.

[39] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4 :2292–2303, 2016.

[40] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *Ieee Access*, 4 :2292–2303, 2016.

[41] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages 360–363. Springer, 2001.

[42] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[43] M. Conoscenti, A. Vetrò, and J. C. De Martin. Blockchain for the internet of things : a systematic literature review. pages 1–6, November 2016.

[44] B. Cusack, Z. Tian, and A. K. Kyaw. Identifying dos and ddos attack origin : Ip traceback methods comparison and evaluation for iot. In *International Conference on Interoperability in IoT*, pages 127–138. Springer, 2016.

[45] F. Dalipi and S. Y. Yayilgan. Security and privacy considerations for iot application on smart grids : Survey and research challenges. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 63–68. IEEE, Aug 2016.

[46] S. R. Das, S. Chita, N. Peterson, B. A. Shirazi, and M. Bhadkamkar. Home automation and security for mobile devices. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 141–146. IEEE, 2011.

[47] A. V. Dastjerdi and R. Buyya. Fog computing : Helping the internet of things realize its potential. *Computer*, 49(8) :112–116, 2016.

[48] F. M. de Almeida, A. de RL Ribeiro, E. D. Moreno, and C. A. Montesco. Performance evaluation of an artificial neural network

multilayer perceptron with limited weights for detecting denial of service attack on internet of things. *training*, 11 :12.

[49] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing : architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18) :1587–1611, 2013.

[50] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on dependable and secure computing*, 3(1) :62–77, 2006.

[51] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM, 2002.

[52] D. Evans and D. M. Eyers. Efficient data tagging for managing privacy in the internet of things. In *2012 IEEE International Conference on Green Computing and Communications (GreenCom)*, pages 244–248. IEEE, 2012.

[53] O. Flauzac, C. GonzÃ¡lez, A. Hachani, and F. Nolot. Sdn based architecture for iot and improvement of the security. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pages 688–693. IEEE, March 2015.

[54] K. Gaurav, P. Goyal, V. Agrawal, and S. L. Rao. Iot transaction security. In *5th International Conference on the Internet of Things (IoT)*, Seoul, S. Korea, october 2015.

[55] M. Gerla, E. K. Lee, G. Pau, and U. Lee. Internet of vehicles : From intelligent grid to autonomous cars and vehicular clouds. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 241–246. IEEE, March 2014.

[56] O. Ghabar and J. Lu. Remote control and monitoring of smart home facilities via smartphone with wi-fly. IARIA, 2015.

[57] C. Gonzalez, S. M. Charfadine, O. Flauzac, and F. Nolot. Sdn-based security framework for the iot in distributed grid. In *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–5. IEEE, July 2016.

[58] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm, 2006.

[59] J. Granjal, E. Monteiro, and J. S. Silva. Security for the internet of things : a survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials*, 17(3) :1294–1312, 2015.

[60] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan. Cp-abe with constant-size keys for lightweight devices. *IEEE transactions on information forensics and security*, 9(5) :763–771, 2014.

[61] J. Guo, I. R. Chen, and J. J. P. Tsai. A mobile cloud hierarchical trust management protocol for iot systems. In *5th IEEE Int. Conf. on Mobile Cloud Computing, Services, and Engineering*, pages 125–130, 2017.

[62] J. Guo, R. Chen, and J. J. Tsai. A survey of trust computation models for service management in internet of things systems. *Computer Communications*, 97 :1–14, 2017.

[63] L. Guo, C. Zhang, and Y. Fang. A trust-based privacy-preserving friend recommendation scheme for online social networks. *IEEE Trans. on Dependable and Secure Computing*, 12(4) :413–427, July 2015.

[64] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, 58(5) :1189–1205, 2013.

[65] T. Hardjono and N. Smith. Cloud-based commissioning of constrained devices using permissioned blockchains. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 29–36. ACM, 2016.

[66] S. H. Hashemi, F. Faghri, P. Rausch, and R. H. Campbell. World of empowered iot users. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 13–24. IEEE, April 2016.

[67] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen. Distributed access control with privacy support in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 10(10) :3472–3481, 2011.

[68] P. Hu. A system architecture for software-defined industrial internet of things. In *2015 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5. IEEE, Oct 2015.

[69] P. Hu, S. Dhelim, H. Ning, and T. Qiu. Survey on fog computing : architecture, key technologies, applications and open issues. *Journal of network and computer applications*, 98 :27–42, 2017.

[70] X. Huang, R. Fu, B. Chen, T. Zhang, and A. Roscoe. User interactive internet of things privacy preserved access control. In *2012 International Conference for Internet Technology And Secured Transactions*, pages 597–602. IEEE, 2012.

[71] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle. 6lowpan fragmentation attacks and mitigation mechanisms. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 55–66. ACM, 2013.

[72] W. Huo-wang and Z. Cheng. Parallel clustering-based k-anonymity algorithm in internet of things. *Information Technology*, 12 :003, 2013.

[73] M. H. Ibrahim. Octopus : An edge-fog mutual authentication scheme. *IJ Network Security*, 18(6) :1089–1101, 2016.

[74] Y. Imine, D. E. Kouicem, A. Bouabdallah, and L. Ahmed. Masfog : An efficient mutual authentication scheme for fog computing architecture. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 608–613. IEEE, 2018.

[75] U. Jayasinghe, N. B. Truong, G. M. Lee, and T. W. Um. Rpr : A trust computation model for social internet of things. In *in Int. IEEE Conf. on Smart World Congress*, pages 930–937, July 2016.

[76] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits. Denial-of-service detection in 6lowpan based internet of things. In *2013*

IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 600–607. IEEE, 2013.

[77] H. M. Kim and M. Laskowski. Towards an ontology-driven blockchain design for supply chain provenance. 2016.

[78] B. Klugah-Brown, J. B. A. Kanpogninge, and X. Qi. A signcryption scheme from certificateless to identity-based environment for wsns into iot. *International Journal of Computer Applications*, 120(9), 2015.

[79] L. Kokoris-Kogias, L. Gasser, I. Khoffi, P. Jovanovic, N. Gailly, and B. Ford. Managing identities using blockchains and cosi. In *9th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2016)*, number EPFL-TALK-220210, 2016.

[80] D. E. Kouicem, B. Abdelmadjid, and L. Hicham. Distributed fine-grained secure control of smart actuators in internet of things. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 653–660. IEEE, 2017.

[81] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef. An efficient architecture for trust management in ioe based systems of systems. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pages 138–143. IEEE, 2018.

[82] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef. Internet of things security : A top-down survey. *Computer Networks*, 141 :199–221, 2018.

[83] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef. Decentralized trust management based blockchain protocol for internet of things. *IEEE Transactions on dependable and secure Computing*, Under Revision.

[84] Y. Lee, W. Lee, G. Shin, and K. Kim. Assessing the impact of dos attacks on iot gateway. Bangkok, Thailand, December 2016.

[85] Y. Leng and L. Zhao. Novel design of intelligent internet-of-vehicles management system based on cloud-computing and internet-of-things. In *Proceedings of 2011 International Conference on Electronic Mechanical*

Engineering and Information Technology, volume 6, pages 3190–3193. IEEE, Aug 2011.

[86] F. Li, Y. Han, and C. Jin. Practical access control for sensor networks in the context of the internet of things. *Computer Communications*, 89 :154–164, 2016.

[87] Z. Lin and L. Dong. Clarifying trust in social internet of things. *arXiv preprint arXiv :1704.03554*, 2017.

[88] J. Liu, Y. Xiao, S. Li, W. Liang, and C. P. Chen. Cyber security and privacy issues in smart grids. *IEEE Communications Surveys & Tutorials*, 14(4) :981–997, 2012.

[89] A. Loibl. Namecoin. *namecoin. info*, 2014.

[90] R. Lu, Z. Cao, Z. Chai, and X. Liang. A simple user authentication scheme for grid computing. *IJ Network Security*, 7(2) :202–206, 2008.

[91] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu. A privacy-preserving trust model based on blockchain for vanets. *IEEE Access*, 6 :45655–45664, 2018.

[92] C. Ma, K. Xue, and P. Hong. Distributed access control with adaptive privacy preserving property for wireless sensor networks. *Security and Communication Networks*, 7(4) :759–773, 2014.

[93] P. Machaka, A. McDonald, F. Nelwamondo, and A. Bagula. Using the cumulative sum algorithm against distributed denial of service attacks in internet of things. In *International Conference on Context-Aware Systems and Applications*, pages 62–72. Springer, 2015.

[94] M. Mainelli and C. von Gunten. Chain of a lifetime : How blockchain technology might transform personal insurance. *Long Finance*, 2014.

[95] Y. Maleh, E. Abdellah, and M. Belaissaoui. Dos attacks analysis and improvement in dtls protocol for internet of things. In ACM, editor, *ACM International conference on Big Data and Advanced Wireless technologies (BDAW'2016)*, Nov. 2016.

[96] L. Malina, J. Hajny, R. Fujdiak, and J. Hosek. On perspective of security and privacy-preserving solutions in the internet of things. *Computer Networks*, 102 :83 – 95, 2016.

[97] K. Manoj. An enhanced remote user authentication scheme with smart card. *International Journal of Network Security*, 10, 01 2010.

[98] T. Mantoro, M. A. M. Adnan, and M. A. Ayu. Secured communication between mobile devices and smart home appliances. In *2013 International Conference on Advanced Computer Science Applications and Technologies*, pages 429–434. IEEE, 2013.

[99] Y. Mao, J. Li, M.-R. Chen, J. Liu, C. Xie, and Y. Zhan. Fully secure fuzzy identity-based encryption for secure iot communications. *Computer Standards & Interfaces*, 44 :117–121, 2016.

[100] X. Meng and D. Liu. Getrust : A guarantee-based trust model in chord-based p2p networks. *IEEE Trans. on Dependable and Secure Computing*, 15(1) :54–68, Jan 2018.

[101] S. Müller, S. Katzenbeisser, and C. Eckert. Distributed attribute-based encryption. In *International Conference on Information Security and Cryptology-ICISC 2008*, volume 5461, pages 20–36, Berlin, Heidelberg, 2009. Springer, Springer Berlin Heidelberg.

[102] S. Nakamoto. Bitcoin : A peer-to-peer electronic cash system, 2008.

[103] C. V. Networking. Cisco global cloud index : Forecast and methodology, 2016–2021. *White paper. Cisco Public, San Jose*, 2016.

[104] K. T. Nguyen, M. Laurent, and N. Oualha. Survey on secure communication protocols for the internet of things. *Ad Hoc Networks*, 32 :17–31, 2015.

[105] M. Nitti, R. Girau, and L. Atzori. Trustworthiness management in the social internet of things. *IEEE Trans. on Knowledge and Data Engineering*, 26(5) :1253–1266, May 2014.

[106] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li. Achieving k-anonymity in privacy-aware location-based services. In *INFOCOM, 2014 Proceedings IEEE*, pages 754–762. IEEE, 2014.

[107] H. Noura. *Adaptation of Cryptographic Algorithms According to the Applications Requirements and Limitations : Design, Analyze and Lessons Learned*. HDR dissertation, UNIVERSITY of PIERRE MARIE CURIE -Paris VI, 2016.

[108] F. Olivier, G. Carlos, and N. Florent. Sdn based architecture for clustered wsn. In *2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 342–347. IEEE, July 2015.

[109] D. Ongaro and J. Ousterhout. In search of an understandable consensus algorithm. In *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, pages 305–319, 2014.

[110] N. Oualha and K. T. Nguyen. Lightweight attribute-based encryption for the internet of things. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE, 2016.

[111] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig. Vanet-based approach for parking space availability. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 75–76. ACM, 2007.

[112] F. Pellarin. Communication method and device for remote control of an actuator for mobile equipment in a building, Feb. 23 2016. US Patent 9,269,261.

[113] A. Rghioui, A. Khannous, and M. Bouhorma. Denial-of-service attacks on 6lowpan-rpl networks : Threats and an intrusion detection system proposition. *Journal of Advanced Computer Science & Technology*, 3(2) :143, 2014.

[114] S. S. N. Roby. Application of blockchain technology in online voting. 2017.

[115] A.-R. Sadeghi, C. Wachsmann, and M. Waidner. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE on Design Automation Conference (DAC)*, pages 1–6. IEEE, June 2015.

[116] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer, 2005.

[117] S. Sahraoui and A. Bilami. Efficient hip-based approach to ensure lightweight end-to-end security in the internet of things. *Computer Networks*, 91 :26 – 45, 2015.

[118] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent. Trust management system design for the internet of things : A context-aware and multi-service approach. *Computers & Security*, 39 :351–365, 2013.

[119] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11) :612–613, 1979.

[120] D. Shreenivas, S. Raza, and T. Voigt. Intrusion detection in the rpl-connected 6lowpan networks. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 31–38. ACM, 2017.

[121] S. Sicari, A. Rizzardi, L. Grieco, G. Piro, and A. Coen-Porisini. A policy enforcement framework for internet of things applications in the smart health. *Smart Health*, 2017.

[122] S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini. Security policy enforcement for networked smart objects. *Computer Networks*, 108 :133–147, 2016.

[123] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini. Internet of things : Security in the keys. In *Q2SWinet MSWiM*, pages 129–133, 2016.

[124] A. F. Skarmeta, J. L. Hernandez-Ramos, and M. V. Moreno. A decentralized approach for security and privacy challenges in the internet of things. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 67–72. IEEE, 2014.

[125] I. Stojmenovic and S. Wen. The fog computing paradigm : Scenarios and security issues. In *2014 Federated Conference on Computer Science and Information Systems*, pages 1–8. IEEE, 2014.

[126] I. Stojmenovic and S. Wen. The fog computing paradigm : Scenarios and security issues. In *2014 Federated Conference on Computer Science and Information Systems*, pages 1–8. IEEE, 2014.

[127] J. Su, D. Cao, B. Zhao, X. Wang, and I. You. epass : An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things. *Future Generation Computer Systems*, 33 :11 – 18, 2014. Special Section on Applications of Intelligent Data and Knowledge Processing Technologies ; Guest Editor : Dominik ÅlÄzak.

[128] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou. Servicetrust : Trust management in service provision networks. In *IEEE Int. Conf. on Services Computing*, pages 272–279, June 2013.

[129] L. Sweeney. k-anonymity : A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05) :557–570, 2002.

[130] T. Team et al. Avispa v1. 1 user manual. *Information Society Technologies Programme (June 2006), http ://avispa-project. org*, 2006.

[131] D. Thatmann, S. Zickau, A. Förster, and A. Küpper. Applying attribute-based encryption on publish subscribe messaging patterns for the internet of things. In *2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS)*, pages 556–563. IEEE, 2015.

[132] S. Tonyali, O. Cakmak, K. Akkaya, M. M. Mahmoud, and I. Guvenc. Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid ami networks. *IEEE Internet of Things Journal*, 3(5) :709–719, 2016.

[133] L. Touati, Y. Challal, and A. Bouabdallah. C-cp-abe : Cooperative ciphertext policy attribute-based encryption for the internet of things. In *2014 International Conference on Advanced Networking Distributed Systems and Applications (INDS)*, pages 64–69. IEEE, 2014.

[134] C. Vandana. Security improvement in iot based on software defined networking (sdn). *International Journal of Engineering and Technology Research (IJSETR)*, 5(1) :291–295, january 2016.

[135] P. Varshney and Y. Simmhan. Demystifying fog computing : Characterizing architectures, applications and abstractions. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pages 115–124. IEEE, 2017.

[136] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang. A survey on mobile edge networks : Convergence of computing, caching and communications. *IEEE Access*, 5 :6757–6779, 2017.

[137] Z. Wang, H. Ding, J. Han, and J. Zhao. Secure and efficient control transfer for iot devices. *International Journal of Distributed Sensor Networks*, 9(11) :503404, 2013.

[138] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin. Storj a peer-to-peer cloud storage network. 2014.

[139] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung. Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet of Things Journal*, 2018.

[140] X. Yao, Z. Chen, and Y. Tian. A lightweight attribute-based encryption scheme for the internet of things. *Future Generation Computer Systems*, 49 :104 – 112, 2015.

[141] S. Yu, K. Ren, and W. Lou. Fdac : Toward fine-grained distributed data access control in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(4) :673–686, April 2011.

[142] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang. Healthcare data gateways : found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10) :218, 2016.

[143] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1) :22–32, Feb 2014.

[144] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han. Fog computing in multi-tier data center networks : A hierarchical game approach. In *2016 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2016.

[145] R. Zhang, Y. Zhang, and K. Ren. Distributed privacy-preserving access control in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(8) :1427–1438, Aug 2012.

[146] S. Zhu, S. Setia, and S. Jajodia. Leap+ : Efficient security mechanisms for large-scale distributed sensor networks. *ACM Trans. Sen. Netw.*, 2(4) :500–528, Nov. 2006.

[147] S. Żółkiewski and K. Galuszka. Remote control of industry robots using mobile devices. In *New Contributions in Information Systems and Technologies*, pages 323–332. Springer, 2015.