Secure authentication protocol for Internet of Things

Achraf Fayad

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAT051

Thèse de doctorat

TELECOM Paris

IP PARIS

Protocole d'authentification sécurisé pour les objets connectés

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Réseaux, informations et communications

Thèse présentée et soutenue à Palaiseau, le 14 décembre 2020, par

ACHRAF FAYAD

Composition du Jury :

Ken CHEN Professeur, Université Paris 13 Nord	Président
Pascal LORENZ Professeur, Université de Haute-Alsace (UHA)	Rapporteur
Ahmed MEHAOUA Professeur, Université Paris Descartes	Rapporteur
Lyes KHOUKHI Professeur, École Nationale Supérieure d'Ingénieurs de Caen-ENSICAEN	Examinateur
Ahmad FADLALLAH Associate Professor, University of Sciences and Arts in Lebanon (USAL)	Examinateur
Rida KHATOUN Maître de conférences, Télécom Paris	Directeur de thèse
Ahmed SERHROUCHNI Professeur, Télécom Paris	Co-directeur de thèse
Badis HAMMI Associate Professor, École pour l'informatique et les techniques avancées (EPITA)	Invité

Acknowledgments

First, I would like to thank my thesis supervisor Dr. Rida Khatoun for the excellent advice, support and encouragement he gave me over the last four years. I thank him for his good advice, his good reflexes, and his availability. I would also like to express my warmest thanks to my co-supervisor Pr. Ahmed Serhrouchni who supported me throughout my course at Télécom Paris. I would like also to express all my thanks to all the members of my thesis's jury: Pr. Ken Chen, Pr. Lyes Khoukhi, Pr. Ahmed Mehaoua, Pr. Pascal Lorenz and Dr. Ahmad Fadlallah.

During my thesis, I really enjoyed working with Dr. Badis Hammi and I hope that we will continue to collaborate in the future. He collaborated in the discussion of the results and the writing of the submitted articles.

I am deeply indebted to my Father Sami and my Mother Najwa, who have encouraged me to continue my studies and overcome all the challenges.

I would like to thank my brothers Rabih and Firas and my sister Rola, my family in law members, and all my friends and colleagues who encouraged me during the past years.

I would like to express appreciation to my beloved wife, my partner and my associate in this life, Soumaya Karaki for her patience, her encouragement, her support and her unconditional love that gave me confidence and allowed me to complete this thesis. This thesis is hers too.

I would also like to express my special appreciation to my lovely beautiful girl Rim and my beloved son Sami, who illuminated my life and push me to look forward to tomorrow.

Contents

Résumé

L'interconnexion de ressources privées sur une infrastructure publique, la mobilité des utilisateurs et l'émergence des nouvelles technologies (réseaux véhiculaires, réseaux de capteurs, Internet des objets, etc) ont ajouté des nouvelles exigences en terme de sécurité du côté serveur et celui du client. L'Internet des objets ou IoT (Internet of Things) semble répondre à des usages bien accueillis par le grand public. On trouve ainsi les applications de l'IoT dans tous les domaines de la vie du quotidien. Les objets Internet sont ainsi prévus en très grand nombre et ceci à une très grande échelle.

La sécurité est l'élément majeur qui renforcera d'une manière certaine une acceptation encore plus grande des IoT par les citoyens et les entreprises. Par ailleurs, le déploiement à grande échelle des objets connectés, sera convoité par les attaques de tout bord. Les cyberattaques opérationnelles sur les réseaux traditionnels seront projetées vers l'internet des objets. La sécurité est ainsi critique dans ce contexte au vu les enjeux sous-jacents.

Objectifs et Contributions

Les objectifs de cette thèse sont de concevoir de nouvelles méthodes sécurisées pour assurer le service d'authentification des objets connectés. La motivation pour mener cette recherche est en fait le manque de **solutions d'authentification adaptatives pour l'IoT** dans le contexte des différentes normes, technologies, infrastructures et plateformes pour l'IoT, et les différentes faiblesses et problèmes ouverts identifiés dans la littérature étudiée, qui étaient abordées dans les différentes contributions de cette thèse.
La première contribution s'appuie sur la cryptographie sur les courbes elliptiques (ECC) et les mots de passe à usage unique (OTP). Le mot de passe à usage unique est une méthode d'authentification dans laquelle un nouveau mot de passe est généré pour chaque session d'authentification et la réutilisation de ce mot de passe n'est pas possible. OTP est l'une des solutions les plus prometteuses pour l'authentification

dans l'IoT et en particulier dans de nombreux scénarios de ville intelligente tel que le contrôle à distance. Cependant, très peu de travaux ont adapté les schémas OTP à l'IoT [1] [2].
Les deuxième et troisième contributions sont basées sur des blockchains. Une blockchain est définie comme une base de données distribuée (registre) qui conserve un enregistrement permanent et infalsifiable des données transactionnelles. Une blockchain est complètement décentralisée en s'appuyant sur un réseau pair-à-pair. Plus précisément, chaque nœud du réseau conserve une copie du registre pour éviter un seul point de défaillance. Toutes les copies sont mises à jour et validées simultanément [3]. Nous pensons que les blockchains représentent une technologie très prometteuse pour répondre aux exigences de sécurité dans le contexte de l'IoT, qui sont encore loin d'être résolues par les architectures centralisées classiques qui ont atteint leurs limites en terme d'évolutivité surtout lorsque des milliers ou des dizaines de milliers d'IoT les appareils sont connectés au même réseau.

Les contributions peuvent être résumées comme suit:

1. Concevoir, mettre en œuvre et évaluer une nouvelle méthode d'authentification légère pour l'Internet des objets: l'approche proposée est une méthode d'authentification par mot de passe unique (OTP) basée sur la cryptographie à courbe elliptique (ECC) et un schéma d'établissement de clés pour l'IoT. Dans le schéma proposé, OTP n'est pas seulement utilisé pour l'authentification, mais également pour générer une clé à usage unique (OTK) qui sera utilisée dans différents services de sécurité tels que le chiffrement et l'intégrité des données. Nous évaluons l'efficacité de notre approche avec une implémentation réelle et comparons ses performances à deux autres approches à savoir, Hash Message Authentication Code (HMAC-based) One Time Password (HOTP) [4] et mot de passe à usage unique basé sur le temps (TOTP) [5]. Les résultats de performance obtenus démontrent l'efficience et l'efficacité de notre approche en termes de sécurité et de performance.

2. Concevoir, implémenter et évaluer une solution d'authentification simple et légère basée sur la blockchain pour les systèmes IoT: **Nous avons fourni une implémentation réelle de notre méthode proposée en nous appuyant sur la blockchain Ethereum et en utilisant différents appareils afin de confirmer sa faisabilité et d'évaluer ses performances**. Les résultats obtenus confirment son adéquation à de tels environnements.

3. Proposer un mécanisme d'authentification décentralisé efficace et léger pour les appareils IoT pour résoudre les limites de l'utilisation des passerelles telles que la non-prise en charge de l'hétérogénéité des approches d'authentification, la non-mobilité des nœuds et la nécessité d'une intervention physique en phase d'initialisation. Le mécanisme proposé s'adapte à des techniques d'authentification hétérogènes afin de garantir une sécurité flexible et plus résiliente. L'approche est une solution de sécurité évolutive qui permet la mobilité des nœuds tout

en assurant leur authentification au niveau de la passerelle avec une intégration aisée de nouveaux appareils ainsi que de nouveaux services. **Nous fournissons une véritable implémentation basée sur Java de notre approche. L'évaluation approfondie fournie montre clairement la capacité de notre système à répondre aux différentes exigences, avec un coût très léger**.

Conclusion

La première contribution est une méthode d'authentification légère pour les objets IoT basée sur ECC et Isogenie offrant deux services de sécurité: l'authentification et la génération de clés. Notre méthode repose sur One Time Password (OTP), une solution très prometteuse pour l'IoT compte tenu de sa robustesse et de sa simplicité. Fondamentalement, la méthode OTP est utilisée pour garantir l'authentification sans avoir à utiliser un serveur tiers. La méthode n'utilise pas l'OTP uniquement pour l'authentification, mais également pour générer une clé à usage unique (OTK) dérivée d'OTP afin d'être utilisée par des algorithmes de chiffrement. Ainsi, notre méthode améliore la confidentialité des données en plus d'assurer l'authentification tout en étant plus efficace en terme de calcul que les solutions existantes. De plus, cette approche a l'originalité de ne pas s'appuyer sur un compteur ou un horodatage comme pour les approches de type OTP synchrones, ni sur une génération et une gestion de challenge côté serveur comme pour les solutions de type OTP asynchrones. De plus, une attention particulière a été accordée au compromis entre les performances et les ressources disponibles sur les appareils IoT. Des évaluations approfondies par le biais de la mise en œuvre ont montré des résultats prometteurs faisant du nouveau système un bon candidat pour une méthode d'authentification légère pour les applications IoT.

La deuxième contribution et la troisième contribution de cette thèse sont basées sur la blockchain. Ce choix est basé sur les caractéristiques inhérentes à la blockchain (par exemple, l'évolutivité, la robustesse, la résilience) pour répondre aux exigences des solutions d'authentification ainsi qu'aux limites de certaines solutions existantes. La deuxième contribution est une méthode d'authentification légère, prenant en charge des scénarios d'authentification simples et mutuels, sans nécessiter un matériel spécifique, ce qui la rend applicable dans une multitude de cas d'utilisation de l'IoT. La méthode proposée a été évaluée à travers une implémentation réelle s'appuyant sur la blockchain Ethereum et utilisant différents dispositifs afin de confirmer sa faisabilité et d'évaluer ses performances initiales. Les résultats obtenus confirment son adéquation à de tels environnements.
La troisième contribution a fourni l'authentification et l'autorisation pour les appareils IoT. Elle présente un avantage important d'être adaptatif en ce sens qu'elle s'adapte à des techniques d'authentification hétérogènes afin d'assurer une sécurité plus flexible et plus résiliente. La sécurité de la solution proposée a été évaluée

analytiquement, tandis que l'évaluation des performances a été réalisée de manière expérimentale à travers une implémentation réelle. Les résultats de son évaluation ont clairement montré sa légèreté et son faible coût.

Abstract

The interconnection of private resources on public infrastructure, the user mobility and the emergence of new technologies (vehicular networks, sensor networks, internet of things, etc.) have added new requirements in term of security on the server side as well as the client side. Examples include the processing time, mutual authentication, client participation in the choice of security settings and protection against traffic analysis. Internet of Things (IoT) is in widespread use and its applications cover many aspects of today's life, which results in a huge and continuously increasing number of objects distributed everywhere.

Security is no doubt the element that will improve and strengthen the acceptability of IoT, especially that this large scale deployment of IoT systems will attract the appetite of the attackers. The current cyber-attacks that are operational on traditional networks will be projected towards the Internet of Things. Security is so critical in this context given the underlying stakes; in particular, authentication has a critical importance given the impact of the presence of malicious nodes within the IoT systems and the harm they can cause to the overall system.
The research works in this thesis aim to advance the literature on IoT authentication by proposing three authentication schemes that satisfy the needs of IoT systems in terms of security and performance, while taking into consideration the practical deployment-related concerns.

One-Time Password (OTP) is an authentication scheme that represents a promising solution for IoT and smart cities environments. This research work extends the OTP principle and proposes a new approach to generate OTP based on Elliptic Curve Cryptography (ECC) and Isogeny to guarantee the security of such protocol. The performance results obtained demonstrate the efficiency and effectiveness of our approach in terms of security and performance.
We also rely on blockchains in order to propose two authentication solutions: first, a simple and lightweight blockchain-based authentication scheme for IoT systems based on Ethereum, and second, an adaptive blockchain-based authentication and authorization approach for IoT use cases. We provided a real implementation of our proposed solutions. The extensive evaluation provided, clearly shows the ability of our schemes to meet the different security requirements with a lightweight cost in term of performance.

List of Figures

List of Tables

Part I

General Introduction

Chapter 1

Introduction

1.1 Context and motivation

Internet of Things (IoT) is actually a broad area of interest and research. It will change the way we live and work by making different aspects of life smart. According to IoT Analytics estimates [13], there were roughly 9.5 billion connected IoT devices at the end of 2019. This number is significantly larger than the forecast of 8.3B devices. The three main drivers of such growth:

- An explosion of consumer (particularly Smart Home) devices
- Much stronger than expected cellular IoT/M2M connections
- Particularly strong device connectivity growth in China.

The number of total connected IoT devices is now expected to reach 28B by 2025 [13].
IoT enables the interconnection of smart physical and virtual objects that are managed by various types of hardware, software, and communication technologies. The large-scale deployment of IoT is actually enabling smart cities, smart factories, smart health and many other applications and initiatives all over the world.

IoT introduces new opportunities such as the capability to monitor and manage devices remotely, analyze and take actions based on the information received from various real-time traffic data streams. Consequently, with the advent of smart cities concept [14], IoT products are changing habits and cities by enhancing infrastructures, creating more effective and cost-efficient municipal services, improving transportation services by decreasing road traffic congestion, improving citizens' safety and providing smart health services [15]. However, IoT technologies also open up multiple risks and privacy issues. **Due to hardware limitations of IoT objects, implementing and deploying robust and efficient security and privacy solutions for the IoT environment remain a significant challenge.**

In addition, there are vulnerabilities and bugs in IoT networks, services and protocols. This makes it possible to carry out attacks at all levels. Hence, many stakeholders are increasingly focusing on the security aspects in IoT. On the other hand, standards for IoT networks have been proposed and used. However, they are not mature yet. With thousands or tens of thousands of devices simultaneously communicating with both users and among themselves, the security implications are of high significance. In this context, Smart cities [14] are the ideal target for hackers to create IoT bot networks. An IoT botnet consists of devices compromised and used to perform different tasks without the knowledge of their legitimate users. In 2016, Dyn firm[1] suffered from a denial of service attack caused by tens of thousands of connected objects (they were mostly connected cameras) to saturate its infrastructure[16]. The attack resulted in Dyn's inability to provide the DNS service. Some of connected objects involved in the attack were infected by Mirai malware [17]. This tool exploits authentication vulnerabilities present in some connected objects such as the use of default passwords that are not been changed by users. This demonstrates how IoT networks are increasingly being used as an attack platform by malicious attackers, and proves that authentication of devices is a key success factor for the Internet of Things.
In order to ensure security services (confidentiality, integrity and authentication) in IoT, the security has to be implemented at the different IoT layers: application layer, network layer (network capabilities and transport capabilities) and physical layer (or Perception layer). Each layer has its "own" protocols with their related security challenges. For instance, the following protocols (among many others) operate at the application layer:

- Message Queue Telemetry Transport (MQTT)[18]: developed by IBM and standardized by Organization for the Advancement of Structured Information (OASIS) on 2016.
- Constrained Application Protocol (CoAP) [19] running over UDP
- Extensible Messaging and Presence Protocol (XMPP) [20] running over TCP.

One of the biggest challenges of IoT is to support heterogeneity and to be secure at the same time, with some additional constraints/ requirements related to the limited resources of the IoT devices, the dynamic change of the topology, and the distributed structure of the IoT system(s). Such complex environment makes the IoT systems vulnerable to a large spectrum of threats such as: physical attacks, denial of service attacks, brute force attacks, virus infections, Man-In-The-Middle (MITM) attacks, use of fake certificates, etc. With the increasing and large-scale integration of IoT in many aspects of our lives (cars, appliances, thermostats, airplanes, etc.) and even in some critical infrastructures, IoT threats can become life-threatening.

[1] Dyn is one of the companies providing DNS service

1.2 Problem Statement

It is important to mention that security issues constitute major obstacles to the worldwide adoption and deployment of IoT. In other words, users will not fully adopt IoT as long as IoT devices make security and privacy risks. IoT remains highly vulnerable to attacks for multiple reasons such as:

- Most of the communications are wireless, which makes the system more vulnerable to numerous attacks such as identity spoofing, messages eavesdropping, messages tampering and other security issues.
- Multiple types of devices have limited resources in terms of energy, memory and processing capacity, which prevent them from implementing advanced security solutions [3].
- Device holders think that IoT data is not that important so that attackers would not be motivated to make attacks.
- IoT is not a collection of independent devices but it is a broad and complex ecosystem that includes devices, communications, interfaces and people.
- Manufacturers often ignore security in order to guarantee a low price for IoT devices.

IoT relies on the cooperation of nodes. Multiple scenarios require that only trusted users can use the offered services. Thus, conventional security requirements such as authentication, data integrity, and sometimes confidentiality are critical to IoT objects, networks, and applications. Despite the maturity of security solutions for traditional networked environment, the resource constraints and heterogeneity of IoT objects' resources make existing security solutions not fully adapted (even completely in-adapted in some cases) to the IoT ecosystem. Indeed, due to devices limits in energy and computational resources, IoT devices cannot adopt the most reliable authentication techniques such as Public Key Infrastructures (PKI), since the devices cannot handle certificates. Moreover, the majority of the existing techniques such as PKI are centralized, which cannot support IoT environments scalability. From the other hand, symmetric cryptography-based solutions are well suited for the IoT devices. However, these solutions also cannot handle the scalability of IoT environments [15][3][21].

This led to intensive research efforts to address the security requirements of IoT while considering the resource-constrained aspect of IoT objects. In particular, the authentication (and the subsequent authorization) have been at the center of interest. Such requirements are inherent in the nature of IoT system. Indeed, the idea behind IoT is the ubiquity of a variety of objects, where they are able to interact and cooperate with each other in order to provide a wide range of services. IoT includes a large number of heterogeneous networked objects. Each object should be reachable and produce content that can be retrieved by any authorized

user regardless of his/her location. It is important that only authenticated and authorized users (objects or people) can access IoT objects. Otherwise, they will be prone to numerous security risks such as information and identity theft.

In a related topic, in most IoT systems and use cases, the architecture relies on a key element, the gateway, to ensure the majority of its activity. IoT gateways play a pivotal role in IoT deployments to collect and aggregate sensor data from various devices. An IoT gateway bridges the communication gap between IoT devices, sensors, equipment, systems and the cloud. IoT gateway devices offer local processing and storage solutions, as well as the ability to autonomously control devices based on data input by sensors. For example, Figure 1.1 describes three IoT use cases:

1. In a Wireless Sensor Network (WSN), all the sensors upload some requested data (e.g. temperature, water level, etc.) to a gateway (also called sink).
2. In a Wireless Body Area Network (WBAN), all the monitoring sensors and motion detectors send and upload their data to a gateway (also called, sink or personal server).
3. In a smart home scenario, all the house components interact with a gateway which also manages the users remote access.

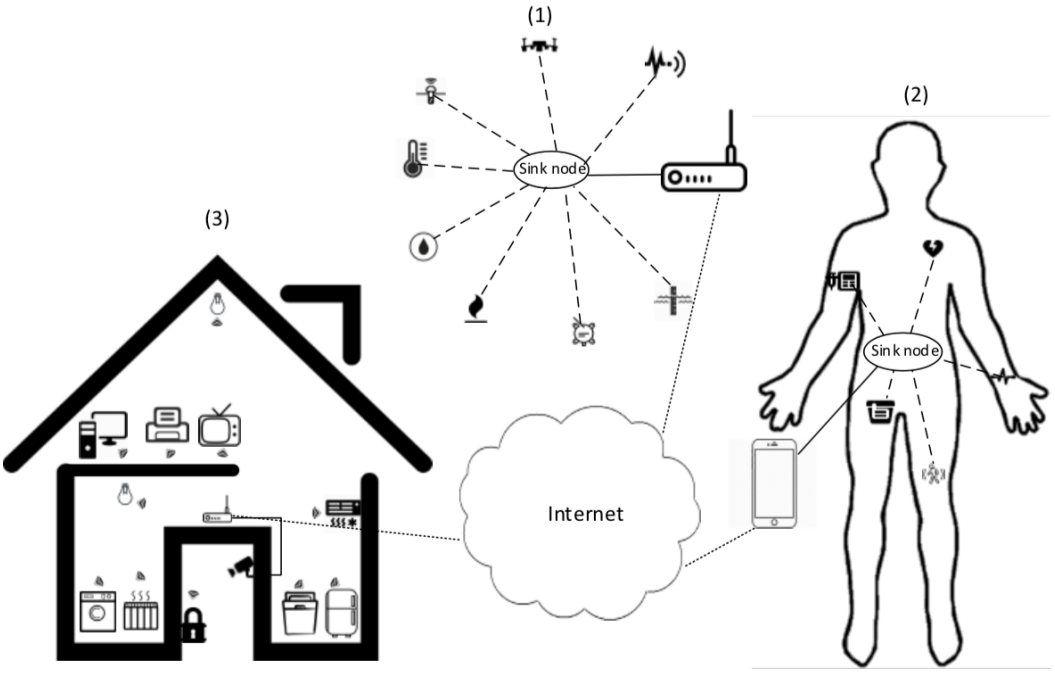


Figure 1.1 – IoT use cases

There are numerous other IoT use cases where a gateway is in the heart of the architecture. Thus, the authentication and authorization functions are performed

at the gateway level [22] [23]. This research interest in IoT authentication has - naturally - led to a large number of IoT authentication schemes which aim to either adapt the existing "traditional" authentication solutions to the IoT context or to propose new schemes and protocols that are by-design oriented for IoT systems.

The first step in our research was to perform an in-depth literature review of the existing IoT authentication schemes (e.g., [24][25] [26][27][3]) Such solutions can be classified according to different criteria, such as [2] [28]:

1. IoT layer: on which authentication is performed; Perception, Network or application layer.

2. Authentication factor: whether the authentication is based on the context (physical or behavior) or the identity.

3. Architecture: whether the architecture of the authentication scheme is centralized or distributed, flat or hierarchical.

4. Hardware support: some authentication solutions rely on hardware and others don't.

5. Authentication procedure: one-way, two-way or three-way (including trusted third party).

The analysis of the existing schemes, led to the identification of a number of open-issues that need to be addressed. Note that some of these issues might be addressed partially in the literature. The open issues can be summarized as follow:

1. IoT systems often have different security requirements (Confidentiality, Integrity and Authentication). Most of the existing authentication schemes do not consider the additional requirements (e.g., confidentiality) and thus a combination of multiple security technologies and solutions is needed which leads to high computational costs. For instance, an IoT system might need one solution for authentication and integrity where at least one key sharing technique is applied, and another solution for confidentiality where another key sharing technique is applied.

2. The lack of support of heterogeneity of authentication approaches: In general, all the nodes related to a gateway use the same authentication method. For example in WSNs a Pre-Shared Key (PSK) is used to authenticate the sensors. In other use cases, if the gateway uses certificate-based authentication, all the nodes must implement this same method. **However, in numerous cases, the network is composed of different types of devices, each with its computation and energy capacities**. For example in a smart home, if some objects, such as heat sensors, can only use PSK-based authentication,

[2] The detailed taxonomy is provided in section 3.3

since symmetric cryptography is less costly than other methods [29], other more powerful objects, such as smart fridges and TVs can easily use public key cryptography based methods, nonetheless, they still use PSK, since the gateway imposes it to satisfy all the existing objects' types.

3. The lack of node mobility support: Each device is often associated to one gateway. However, in some use cases, some nodes can have mobility features (a sensor whose location is changed or any other mobility case). To the best of our knowledge, nodes that have mobility features from a gateway to another are not authenticated at this gateway level but using a central server/service accessible through Internet (e.g., Intelligent Transportation Systems). Nevertheless, realizing authentication at the gateway level can be less time consuming. **Moreover, relying on centralized services in an environment such as IoT where the number of devices is exploding (countable in billions) can represent a real bottleneck**.

4. Initialization phase: in the majority of systems, if a new device is added, a physical intervention on the gateway is needed to set up and configure the credentials of this new device (e.g., adding the PSK of a new sensor on the gateway). **However, knowing that the number of objects is exploding, this feature represent a real brake**.

5. Architecture-specific nature: Most of the schemes in the literature are dependent on a specific architecture of the IoT system, which makes them inadaptable to other IoT architectures.

6. Key storage issues: Many schemes require local key management and need infrastructure for storing keys, which makes them vulnerable to key thefts [30] [1] [31].

7. One factor authentication: Several solutions rely on a one-factor authentication scheme which can be a security risk in such an environment.

1.3 Thesis objectives and Contributions

The objectives of this thesis are to design new secure scheme(s) to ensure the authentication service for the connected objects. The motivation for conducting this research is in fact the lack of **adaptive authentication solutions for IoT** in context of different standards, technologies, infrastructures and platforms for IoT, and the different gaps and open issues identified in the surveyed literature, which were addressed in the different contributions of this thesis.
The first contribution relies on Elliptic-Curve Cryptography (ECC) and One-Time Password (OTP) techniques. One-Time Password is an authentication scheme in which a new password is generated for each authentication session and the reuse of a password is not possible. OTP is one of the most promising solutions for

authentication in IoT and specifically in numerous smart city scenarios such as remote control. However, there are very few works that have adapted OTP schemes to IoT [1] [2].
The second and third contributions are based on blockchains. A blockchain is defined as a distributed database (ledger) that maintains a permanent and tamper-proof record of transactional data. A blockchain is completely decentralized by relying on a peer-to-peer network. More precisely, each node of the network maintains a copy of the ledger to prevent a single point of failure. All copies are updated and validated simultaneously [3]. **We believe, that blockchains represent a very promising technology to meet security requirements in IoT context, which are still far from being solved by the classical centralized architectures which reached their limits in term of scalability especially when thousands or tens of thousands of IoT devices are connected in the same network**.

The contributions can be summarized as follow:

1. Design, implement and evaluate a new lightweight authentication method for Internet of Things: The new approach is a novel Elliptic Curve Cryptography (ECC) based One Time Password (OTP) authentication method and key establishment scheme for IoT. In the proposed scheme, OTP is not only used for authentication, but also to generate a One Time Key (OTK) which will be used in different security services such as Data encryption and integrity. We evaluate the efficiency of our approach with a real implementation and compared its performance with two other approaches namely, Hash Message Authentication Code (HMAC-based) One Time Password (HOTP) [4] and Time-based One Time Password (TOTP) [5]. The performance results obtained demonstrate the efficiency and effectiveness of our approach in terms of security and performance.

2. Design, implement and evaluate a simple and lightweight blockchain-based authentication solution for IoT systems: **We provided a real implementation of our proposed scheme relying on Ethereum blockchain and using different devices in order to confirm its feasibility and evaluate its initial performances**. The results obtained confirm its suitability for such environments.

3. Propose an efficient and lightweight decentralized authentication mechanism for IoT devices to resolve the limitations of using gateways like the non-support of heterogeneity of authentication approaches, the non-mobility of nodes and the need of physical intervention in initialization phase. The proposed mechanism adapts to heterogeneous authentication techniques in order to ensure flexible and more resilient security. The approach is a scalable security solution that allows the mobility of nodes while ensuring their authentication at the gateway level with an easy integration of new devices as well as new ser-

vices. **We provide a real Java-based implementation of our approach. The extensive evaluation provided, clearly shows the ability of our scheme to meet the different requirements, with a very lightweight cost**.

1.4 Manuscript organization

The rest of this manuscript is organized as follows:

- Chapter 2 presents a general overview of the Internet of Things (IoT) domain, including the architecture, standards, technologies, applications and current challenges for IoT-based systems.
- Chapter 3 introduces the IoT security context and challenges with special emphasis on the authentication field and a review of the research literature in this field, and concludes with the requirements of strong IoT authentication schemes
- Chapter 4 presents the first contribution of the thesis: ECC-based One-Time Password (OTP) Authentication for Internet of Things (IoT). It provides the necessary technical cryptography background, then presents how the proposed solution combines ECC with Isogeny to generate One-Time Keys (OTKs) in order to ensure authentication. This chapter also evaluates the proposed solution based on a real implementation.
- Chapter 5 presents the second and third contributions of this thesis which are based on the blockchain distributed ledger. After providing a condensed overview on blockchains, it details two scalable authentication schemes, that have the flexibility to adapt to multitude IoT use cases, the robustness to resist to different attacks, and the lightweightness proven by their experimental evaluation to adapt to the special constrained nature of IoT systems.
- Chapter 6 concludes this manuscript and gives an overview of the future works.

Chapter 2

Introduction to the Internet of Things (IoT)

2.1 Introduction

Internet of Things (IoT) concept is evolving exponentially and covers more areas everyday. For example, numerous activities and habits of our daily lives rely on IoT systems. Indeed, IoT has the potential to introduce and develop a smart world by the development of new applications in different domains like smart homes, smart health, smart cities, industry 4.0, Wireless Sensor Networks (WSN), smart agriculture, etc. [15].
The IoT ecosystem contains platforms, networks, devices[1], applications, which require potential security measures on each layer, as well as analysis and control capabilities on all data and information in order to establish trusted connections.
In order to understand the security issues and challenges of the IoT ecosystem, a first step is obviously to understand the ecosystem itself. This chapter provides a comprehensive overview of the Internet of Things (IoT), which includes:

- The general architecture(s) of such system.
- The research and development efforts and their outcomes in terms of standards, protocols and software.
- The commonly used technologies in the IoT industry.
- A quick analysis and comparison of the IoT platforms.
- The key application domains of IoT.
- The current challenges of IoT-based systems.

[1] In the rest of this thesis the terms thing, object, device and node are used interchangeably to refer to a connected smart thing.

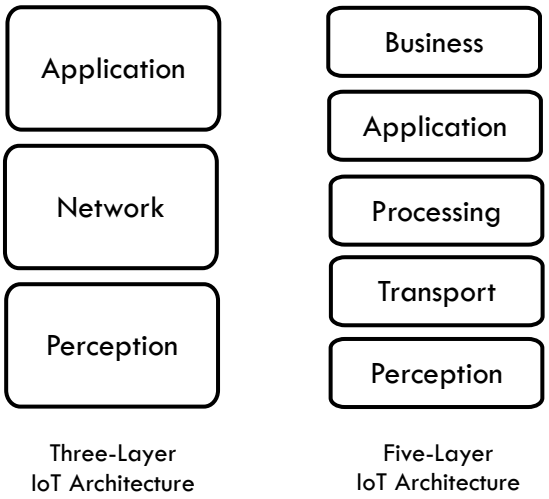


Figure 2.1 – IoT Layered Architecture

2.2 IoT overview

An IoT ecosystem brings together different heterogeneous IoT components in a managed way to build an efficient system [32]. It integrates different types of nodes (e.g., sensors, actuators, gateways, etc.) which are connected using different types of protocols and interfaces.
The sensors and actuators are the leaf nodes of the ecosystem. The sensors (also called perceptors) collect information from the environment (e.g., temperature, pressure, etc.) and send data to the gateway. The actuators receives instructions from the gateway and act accordingly on their surroundings.
The gateway plays a key role in the IoT ecosystem. It is in general responsible of the management of large number of sensors/ actuators, as well as the flow of data between such devices. It has also the responsibility of filtering, aggregating and formatting the data received from the sensors before sending them to the cloud, where a high level processing of the data collected from the different gateways is done, in order to be later provided to the IoT applications which give the complete IoT view to the user [32].

The above-mentioned view of the IoT ecosystem can be projected in a layered architecture, which in its basic form, consists of three layers and can be extended to five layers (Figure 2.1) [33].

A three-layer architecture consists of the following layers:

1. **Perception Layer**: also known as the physical layer. It consists of sensors and actuators that collect information (sensors) from the environment or act on this environment (actuators).

2. **Network/Transport Layer**: responsible for transmitting and processing the information collected at the perception layer.

3. **Application Layer**: responsible for providing users with application-specific services.

The five-layer architecture gives more abstraction to the IoT architecture by introducing two additional layers. It consists of the following layers: Perception, Transport, Processing, Application, Business. The Perception and Application Layers are the same of the three-layer architecture. The Transport layer has the same functionalities of the network layer. The two additional layers are:

1. **Processing Layer**: stores, analyses and processes the information collected from the objects received from the transport layer.

2. **Business Layer**: This layer manages the IoT system and its applications, business model, and services. It utilizes data received from the application layer to build business models, graphs, flowcharts, etc. It is also responsible of ensuring user's privacy[34].

Another Five-layer architecture maintains the first three layers of the above-mentioned architecture, and puts on top of them a middleware layer and an application layer. The middleware layer is responsible to perform different logical and analytical operations on the data and process it to meaningful information.
Other IoT architectures are proposed in the literature. Examples include Service-Oriented (SOA) based architecture and fog-based architecture. In SOA-based architecture, the IoT functionalities are abstracted and exposed through interfaces. Objects and applications use these functionalities through services [35]. The fog-based architecture consists of six layers; the physical (perception) layer being on the bottom and the application layer on the top. Four layers are present in between: monitoring, pre-processing, storage and security [36].

2.3 Standards, Protocols and Software

The deployment of IoT needs communication standards that seamlessly operate among the various objects. Several worldwide organizations are involved in standardizing such communications. These include the International Telecommunication Union (ITU), the Institute of Electrical and Electronics Engineers (IEEE), the Internet Engineering Task Force (IETF), Global Standard1 (GS1), the Organization for the Advancement of Structured Information Standards (OASIS), the Industrial Internet Consortium (IIC), and several others.
We briefly present some of these IoT standards and initiatives in Table 2.1. For example, the Internet of Things Standard Global Initiative (IoT-GSI) supported by ITU made two recommendations: the ITU-T Y.2060 [37], which provides an overview of the IoT concepts and ITU-T Y.2061 [37], which describes the conditions for the machine interface oriented towards applications. Various standards were proposed by IEEE and IETF at different levels for sensor networks based on the Internet Protocol (IP). For example, at the link layer, the IEEE 802.15.4

standard is more suitable than Ethernet in industrial environments. At the network level, the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) standard can adapt the IPv6 protocol for wireless communications [38]. In 2011, the IETF published the IPv6 Routing Protocol (RPL) standard for Low-power Networks. Figure 2.2 compares the 6lowPAN communication stack with other popular communication stacks.

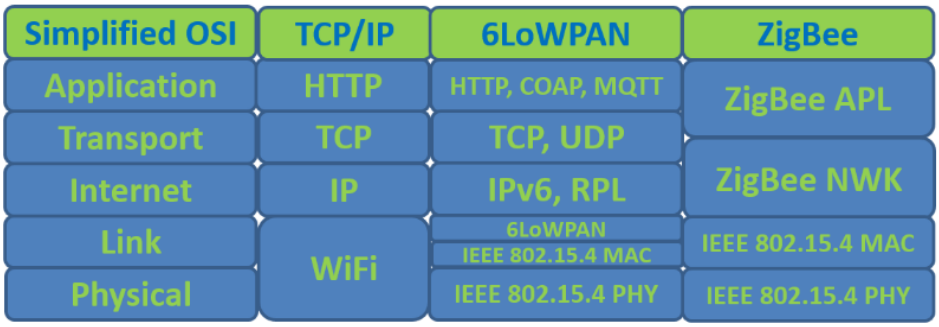


Figure 2.2 – Comparison of 6LowPAN's stack with other stacks [6]

IETF has also launched a Working Group to standardize an application layer-oriented protocol for connected objects. The reference protocol is called the Constrained Application Protocol (CoAP). CoAP [19] provides methods and commands (such as, HTTP Get) to query an object and change its status. CoAP relies on UDP and can optionally use Datagram Transport Layer Security (DTLS), to provide communication security.
Operating systems [39] used in IoT are various; examples are: TinyOS, Contiki OS, MantisOS, Nano-RK, Android, Brillo (Google), Windows 10 IoT Core, LiteOS (Huawei), Mbed OS (ARM). In addition, several platforms [39] have been developed for IoT, such as Arrayent, Californium CoAP Java framework, Erbium, CoAP framework for Contiki, and XMesh networking stack.
At the application layer, a large number of applications have been developed [39] like Iobridge Thingspeak, Nimbits, Evrythng, Open.Sen.se, NanoService, exosite One, HP supposed, Isidorey, SensorCloud, Manybots, and so on.

The Electronic Product Code Global (EPC Global) initiative of the organization Global Standard 1(GS1)[2] defines a unique individual identifier for identifying an electronic product and the overall EPC network architecture that defines the organization of information systems designed to ensure the exchange of information in an EPC network [40] [41]. One of its main components is the Object Naming Service (ONS) which is based on the Domain Name System (DNS). In fact, in 1970 the European Article Numbering (EAN) standard emerged for product identification. However, this EAN barcode is actually used to identify a class of

[2]http://www.gs1.org

	802.11a	**802.11b**	**802.11g**	**802.11n**	**802.11 ac**	**802.11 ad**	**802.15.1**	**802.15.3**	**802.15.4**	**802.15.6**	**NFC**
Network Type	WLAN	WLAN	WLAN	WLAN	WLAN	WLAN	WPAN	WPAN	WPAN	WBAN	Point-to-Point
Date	1999	1999	2003	2009	2014	2012	2002/2005	2003	2007	2011	2011
Network Size	30	30	30	30			7	245	65535	250	-
Bit Rate	54 Mbps	11Mbps	54 Mbps	248 Mbps	3.2 Gbps	$\geq$ 7Gbps	3 Mbps	55 Mbps	250 Kbps	10 Mbps	424 Kbps
Frequency	5 GHz	2.4 GHz	2.4 GHz	2.4/5 GHz	5 GHz	2.4/5/60GHz	2.4 GHz	2.4 GHz	868-915 MHz 2.4 GHz	402-405 MHz	13.56 Mhz
Range	120 m	140 m	140 m	50 m indoor 250 m outdoor	30 m	5 m	100 m	100 m	75 m	2-5 m	0.2 m
Modulation	BPSK, QPSK 16-QAM 64-QAM OFDM	DBPSK DQPSK CCK DSSS	DBPSK DQPSK 16-QAM 64-QAM OFDM	OFDM	OFDM	QAM-256	8DPSK DQPSK PIDQPSK GFSK AFM	QPSK DQPSK 16-QAM 32-QAM 64-QAM	ASK DSSS PSSS		Manschester and Modified Miller
Application	WiFi	WiFi	WiFi	WiFi			Bluetooth		ZigBee		

Table 2.1 – Main communication standards within IoT

products, not individual instances within this class. Furthermore, in IoT, a unique IP address for each connection is required. This is why EPC was proposed by GS1 as a new standard. Meanwhile, OASIS[3] issued various recommendations on network technologies in IoT and messaging technologies such as Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP) and the Data Distribution Service for Real-Time Systems (DDS). In 2014, a new Industrial Internet Consortium (ICC[4]) was launched in order to coordinate and establish the priorities and enabling technologies of the Industrial Internet. There are thousands of founding and contributing members of ICC and they include: Bosh, Intel, IBM, Schneider, Huawei, Cisco, and several others. There are currently 19 Working Groups and teams working on different areas: Business Strategy and Solution Lifecycle, Legal, Liaison, Security, Technology Testbeds, Marketing and Membership, and so on. Figure 2.3 summarizes some IoT protocols and standards.

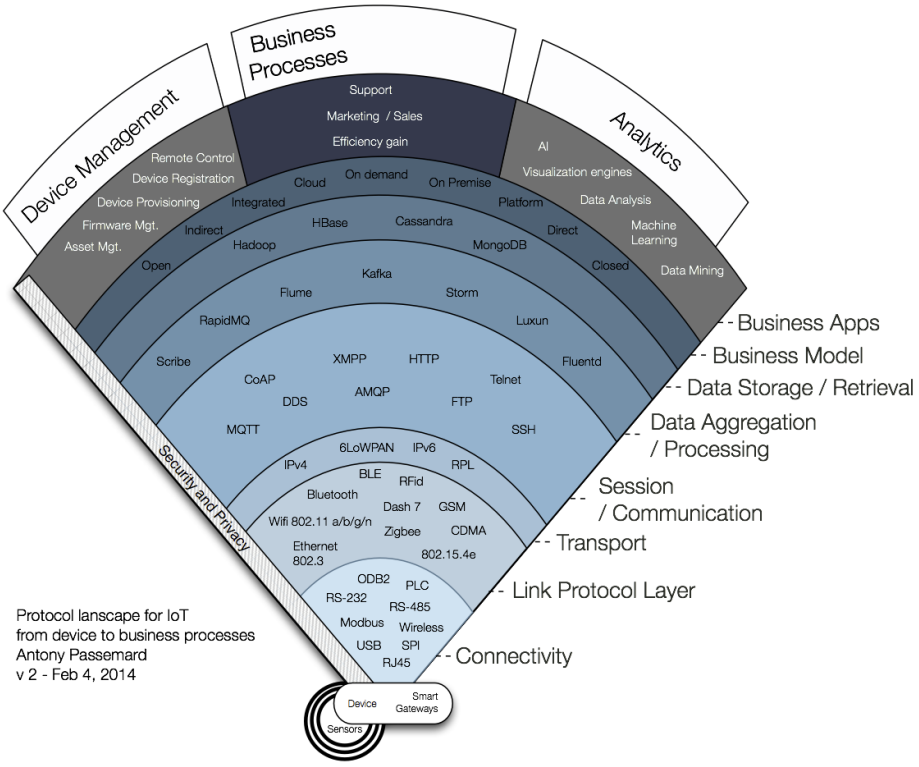


Figure 2.3 – Internet of Things protocol stack [7]

[3]http://www.oasis-open.org
[4]http://www.iiconsortium.org

2.4 IoT Technologies

IoT technologies are expected to be part of large scale networks, with the number of devices in the thousands and areas spanning several kilometers. In this section, we focus primarily on the LoRa Ultra-Narrow Band (UNB) technology developed by Semtech and SigFox .

2.4.1 LoRa

LoRa is a wireless technology designed to provide the low-power within wide-area networks (LPWANs) required for IoT services [42]. The technology offers a mix of long range, low power consumption and secure data transmission. The LoRa standard [43] has been developed for IoT-type devices in regional or global networks. This technology provides seamless interoperability among devices without requiring any complex installations. The services targeted include home energy monitoring, alarm systems, remote health monitoring, transportation, environment protection, and so on. This specification defines the communication protocol and system architecture for the underlying network. It supports frequencies in the 433, 868 or 915 MHz ISM bands, depending on the area where it is deployed. In Europe, it uses either Gaussian Frequency Shift Keying (GFSK) or the proprietary LoRa modulation system, which works with a version of Chirp Spread Spectrum using 125 KHz channel bandwidth [44]. LoRa architecture is describes in Figure 2.4

The hierarchical star-based topology is used by LoRa networks. IoT devices in such networks can be servers, end-points, or gateways. Data rates can range, in Europe, from 0.3 Kbps up to 50 Kbps when channel aggregation is employed. In North America, the minimum data rate is 0.9 Kbps because of Federal Communications Commission (FCC) requirements. The payload for this technology can range from 2 to 255 bytes [45]. This standard is optimized for low cost and battery operated sensors. The devices are asynchronous and communicate only when they have data ready to send whether event-driven or scheduled. Power consumption is proportional to the time devices spent while in the listening mode.

LoRa is gaining significant attention in IoT networks that are being deployed by wireless network operators. It can be deployed with minimum upfront infrastructure investments and operating costs. When increased network capacity is required, further Gateways can be added. It has been estimated that the deployment cost of this technology in unlicensed bands needs much less capital than even a 3G software upgrade [45]. Major Telecom operators (e.g., Swisscom, NKE Electronics, and others) are deploying this technology for nationwide networks because of its benefits over competing technologies. These benefits include bi-directional communications, mobility for asset tracking, security, and accurate localization [8].

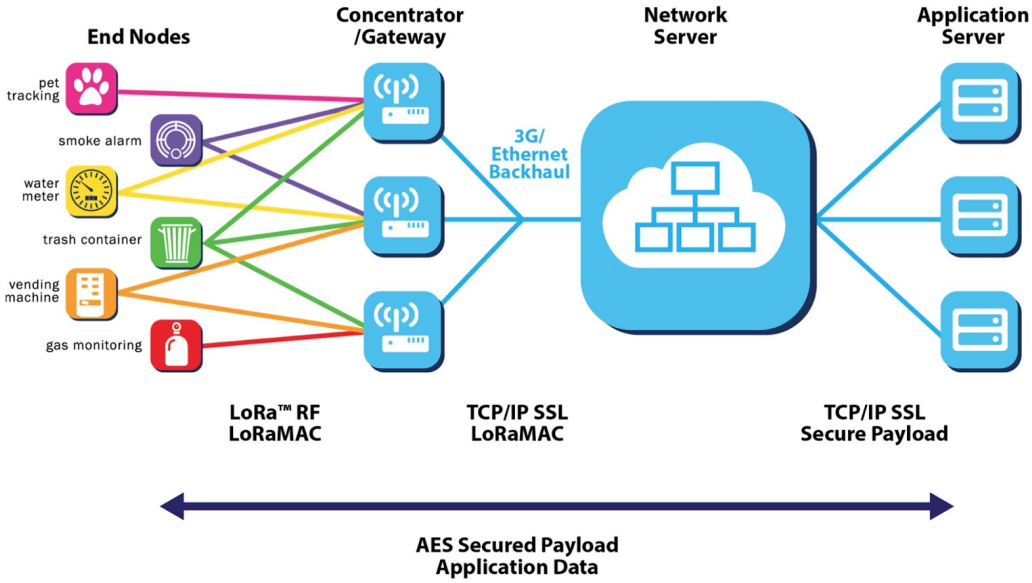


Figure 2.4 – LoRa architecture [8]

2.4.2 SigFox

SigFox[5] created an ultra-narrow band IoT communication system designed to support IoT deployments over long ranges, e.g. in excess of 20 km between a client device and a base station. SigFox uses license-exempt spectrum for its product, namely the 868 MHz band in Europe and 915 MHz band in the US, to transmit data over a narrow spectrum to and from connected objects. The ultra-narrow band operation is achieved using bandwidth channels lower than 1 KHz transmitting data payloads of 12 bytes uplink and 8 bytes downlink with a protocol overhead of 26 bytes [45].

One of the advantages of SigFox devices is their resource efficiency. The power demand is negligible because devices are only "on" when they are transmitting; this means that the power demand is a fraction of that for a device operating on cellular networks. SigFox technology allows deploying very efficient, low throughput communications by limiting the number of antennas (base stations). For the same level of coverage, SigFox requires around 1,000 times less antennas and base stations, compared with some cellular networks[46]. This technology offers access to a service management interface, which can enable the control of main communication parameters such as battery and temperature settings, signal quality, volume of exchanged data, and others. Networks based on SigFox technology have already connected thousands of devices in several international cities. They are currently operational in 14 countries, covering an area of more than 1.2 million km^2

[5] http://www.sigfox.com

Standard	Frequency Band	Range
SigFox	868 MHz/902 MHz ISM	30-50km (rural) 3-10km (urban) 1000km LoS
LoRaWAN	868 MHz/902 MHz ISM	2-5k (urban) 15k (rural)
LTE-M	Cellular	2.5- 5km
IEEE P802.11ah (low power WiFi)	License-exempt bands below 1 GHz, excluding the TV White Spaces	Up to 1Km (outdoor)
Dash7 Alliance Protocol 1.0	433, 868, 915 MHz	0.5 km
Ingenu RPMA	2.4 GHz ISM	500 km LoS
nWave	Sub-GHz ISM	10km (urban) 20-30km (rural)

Table 2.2 – Comparison of low power WAN technologies [12]

and reach 223 million people[6].

Table 2.2 compares different low power WAN technologies used in IoT use case scenarios.

2.5 Platforms and Deployment

The significant growth in IoT deployment has led to the emergence of IoT platforms which support:

- Easy integration of new devices and services.
- Communication between devices (objects and servers).
- Management of different devices and communication protocols.
- Transmission of data flows and the creation of new applications.
- Interoperability among components, objects, gateway, cloud data, and software applications.
- Scalability of the IoT infrastructure.

According to the level of services provided, IoT platforms can be divided into:

[6] http://www.iotglobalnetwork.com

1. Infrastructure-as-a-service platforms: they provide hosting space and processing power for applications and services, e.g. IBM Bluemix[7].

2. M2M connectivity platforms: they focus only on the connectivity of IoT objects through telecommunication networks and protocols, e.g. Comarch[8] and AirVantage[9].

3. Hardware-specific software platforms: numerous companies sell their proprietary technology which includes the hardware and the software back-end, e.g Google Nest[10].

4. Enterprise software extensions: some software and operating system companies such as Windows and Apple are increasingly allowing the integration of IoT devices such as smartphones, connected watches and home devices.

According to [47] [48], the main features an IoT platform must achieve are:

- Device integration
- Networking
- Device management
- Security
- Protocols for data collection
- Analytics
- Support for visualization

Based on these features, the authors in [9] have proposed a stack for the IoT platform architecture as shown in Figure 2.5.

As we mentioned earlier, the number of IoT platforms is growing at a fast pace. According to [49], this market will reach $1 billion in 2019. Table 2.3 provide a comparison of different platforms, describing their offered services, advantages and limitations.

2.6 IoT Applications

The IoT facilitates the development of a large range industry-oriented and user-specific IoT applications in different fields, such as domain industry applications, smart agriculture, smart logistics, intelligent transportation, smart grids, smart environmental protection, smart safety, smart healthcare, smart home, etc. [61]

[7] https://www.ibm.com/cloud-computing/bluemix/fr
[8] http://www.comarch.com/telecommunications/solutions/m2m-platform/
[9] https://airvantage.net
[10] https://nest.com/

Platform	Services	Advantages	Disadvantages
AWS IoT [50]	Visualization, monitoring, and analyzing data received from wired or wireless sensors	Data transactions security	Private platform
Oracle IoT cloud [51]	Real-time data capture, M2M platform	Support millions of device endpoints, heterogeneous connectivity	No support for open source based devices
Microsoft Research Lab of Things [52]	Smart home services	HomeOS source code (platform, drivers, apps) is available for academic research	Available only for Microsoft based products
Open remote [53]	Buildings, home automation, healthcare and smart cities services	Variety of supported protocols Cloud services	High cost
KAA [54]	Multi-domain open source platform : agriculture, healthcare, industrial IoT, applications for consumer electronics and smart home	Open IoT cloud platform Big data support	Limited hardware modules supported
ThingsBoard [55]	Open-source IoT platform: data collection, processing, visualization, and device management Support IoT protocols : MQTT, CoAP and HTTP	Data confidentiality and device authentication	Relatively new platform and not tested in large scale use
Plotly [56]	Data visualization, Interactive charts and dashboards	Excellent visualization tools	Offers only visualization services
IBM IoT [57]	Cognitive IoT	Single Sign-On (SSO) authentication, IDentity as a Service	Network latency
Kii [58]	Mobile back-end as a service (MBaaS) : user management, data management and push notification functionality, numerous services in multiple areas	API specification is open to the public Load balancing and security	Communication latency between the mobile app and devices
Echelon [59]	Industrial platform for different use cases: area and street lighting, building automation, transportation systems	autonomous control and security Integration of multi-vendor devices in extensible architecture	High cost proprietary technology
Axeda [60]	Cloud-based software for managing IoT	Highly secure communications, M2M learning	Integration of third party systems

Table 2.3 – Comparison of the main IoT platforms

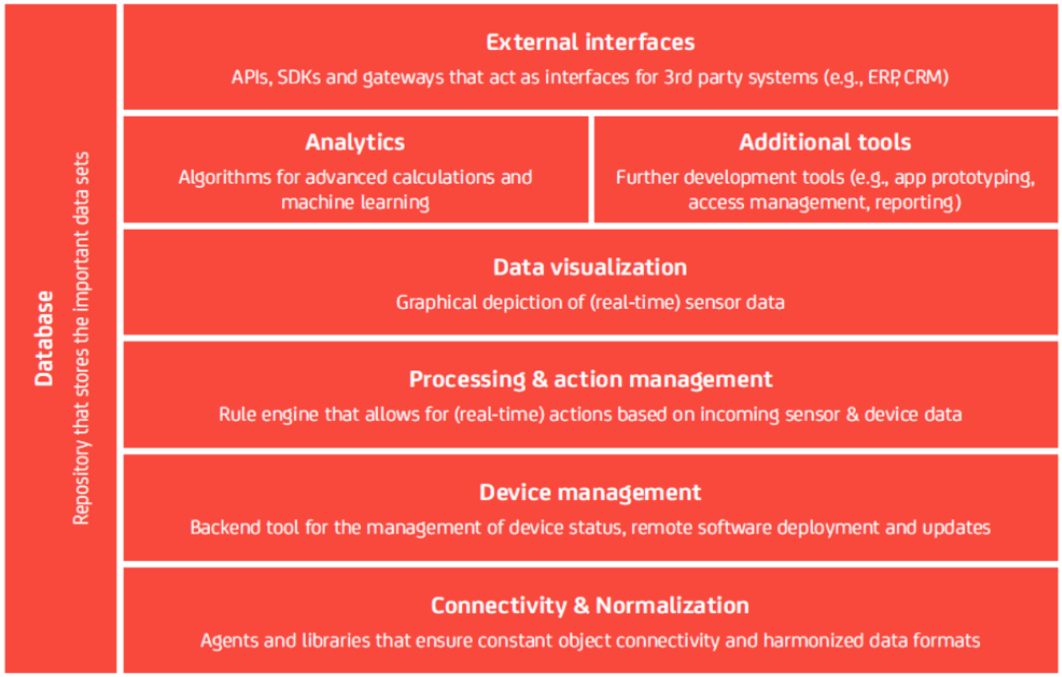


Figure 2.5 – Main components of an IoT Application Enablement Platform [9]

[62]. In other words, IoT applications will have the role to add smartness to most aspects of today's life.
In this section, we present a number of IoT applications with a special emphasis on smart city domain, given the increasing interest of government and industry to invest in such field and most importantly the direct and growing impact such applications will have on both people and the planet.
Indeed, the IoT concept leverages several ubiquitous services to enable Smart City deployments all over the world. IoT introduces new opportunities such as the capability to monitor and manage devices remotely, analyze and take actions based on the information received from various real-time traffic data streams. As a result, IoT products are changing cities by enhancing infrastructures, creating more effective and cost-efficient municipal services, improving transportation services by decreasing road traffic congestion, and improving citizens' safety. To achieve the full potential of IoT, smart city architects and providers recognize that cities must not offer a separate smart city feature, but rather deliver scalable and secure IoT solutions that include efficient IoT systems.

Health of Buildings

To properly maintain the historical buildings of a city we need to: (1) continuously monitor the actual conditions of each building and (2) to identify the most affected areas due to various external agents [63]. The city contains multiple structures,

which have different sizes and different ages. It is different from one city to another, but, generally, most of the structures are very old (such as buildings, dams, or bridges [64]). To assess the conditions of a building, passive IoT sensors can be embedded within a concrete structure, and periodically send a radio signal of suitable amplitude and phase characteristic to inform about the structure's state [65].

Environmental Monitoring

IoT processes, analyzes, and disseminates information collected from multiple environments [64]. The various parameters measured by sensors [66] are:

- Water level for lakes, streams, sewages.
- Gas concentration in the air for cities, laboratories, and deposits.
- Soil humidity and other characteristics.
- Inclination for static structures (e.g., bridges, dams).
- Position changes (e.g., for landslides).
- Lighting conditions either as part of combined sensing or standalone (e.g., to detect intrusions in dark places).
- Infrared radiation for heat (fire) or animal detection.

Waste Management

Waste management becomes an increasing problem in urban living. It is related to many aspects including socioeconomic and environmental ones. One important feature in waste management is environmental sustainability [67]. A major benefit of global IoT infrastructures is that they provide us with the ability to collect data and, further help in improving effective management for various issues. Nowadays, the garbage-truck needs to pick-up all garbage cans even when they are empty [68]. By using IoT devices inside the garbage can, these devices will be connected to the computing server using one of LPWAN technologies. The computing server can collect the information and optimize the way to garbage-collection is performed by the garbage trucks.

Smart Parking

In this use case, there is a wireless sensor (or connected object) at each parking spot. If a vehicle parks, or if a parked vehicle leaves a parking spot, the sensor at the parking spot sends a notification to a management server. By collecting information regarding the parking bay occupancy, the server can provide parking vacancy information to drivers through visualization platforms such as smart-phones, vehicles' Human Machine Interfaces (HMIs) or advertisement boards. These information will also enable the city council to apply fines in case of parking infringements [65].

Radio Frequency IDentification (RFID) technology is automated and can be very useful to vehicle identification systems. Vehicles are identified and parking-lot fees are collected automatically via this system [69].
As for the hardware requirements, by utilizing RFID readers, barriers, parking-lot check-in and check-out controls can be achieved. In this way, in contrast to personnel-controlled traditional parking-lot operations, an unmanned, automated vehicle control and identification system can be developed as described in [69]. The development of Vehicle Ad Hoc Networks (VANETs) [70] along with the advances and wide deployments of wireless communication technologies, many major car manufactories and telecommunication industries are increasingly fitting their cars with On Board Unit (OBU) communication device. This allows different cars to communicate with each other as well as with the roadside infrastructure. Thus, applications that provide information on parking space occupancy or guide drivers to empty parking spaces, are made possible through vehicular communications [71].

Smart Health

A Wireless Body Area Network (WBAN) which is based on a low-cost wireless sensor network technology could greatly benefit patient monitoring systems in hospitals, residential and work environments [72]. The miniature sensors can be embedded inside the body or mounted on the surface of the body. The sensors communicate with medical devices using different technologies of WPAN (ZigBee, 6LowPAN, CoAP, etc.). The sensors are also capable of measuring various physiological parameters information (e.g., blood flow, respiratory rate, blood pressure, blood PH, body temperature, and so on), which are collected and analyzed by remote servers (see Figure 2.6). The wearability requirement poses physical limitations on the design of these sensors. The sensors must be light, small, and should not hinder patient' movements and mobility. Moreover, because the sensors need to operate on small batteries included in the wearable package, they need to be highly energy-efficient [10].

Navigation System for Urban Bus Riders

Urban Bus Navigator (UBN) is based on an IoT architecture which uses a set of distributed software and hardware components that are tightly integrated with the bus system. The UBN system deployed in Madrid, Spain is composed of three key components [73]:

1. The network-enabled urban bus system with WiFi equipped buses

2. The UBN navigation application for bus riders

3. The bus crowd information server which collects real-time occupancy information from buses operating on different routes in Madrid

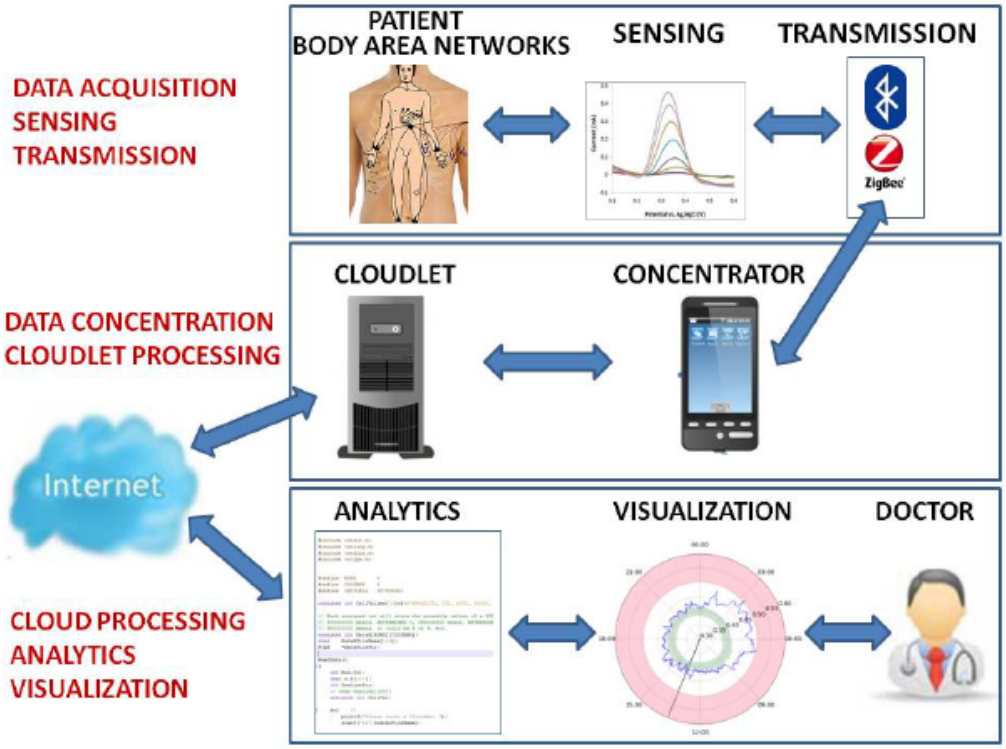


Figure 2.6 – Remote patient monitoring system based on IoT-Cloud architecture [10]

Smart Grid

The smart grid uses new technologies such as intelligent and autonomous controllers, advanced software for data management, and two-way communication between power utilities and consumers, to create an automated and distributed advanced energy delivery network (see Figure 2.7) [11]. Deployed as an infrastructure for sensing and transmitting information for the smart grid, the IoT technology, when applied to the power network, will play a significant role in cost-effective power generation, distribution, transmission and consumption [74].

Autonomous driving

In a smart city, autonomous driving technologies will be synonymous to saving time for the user. This technology would help speed up the flow of traffic in a city and save almost 60% [75] of parking space by parking the cars closer to each other. Through a combination of radar, cameras and ultrasonic sensors located around the car, an autonomous car can detect anomalies all around and trigger an alert that automatically activates the emergency brakes to prevent accidents or collisions. The

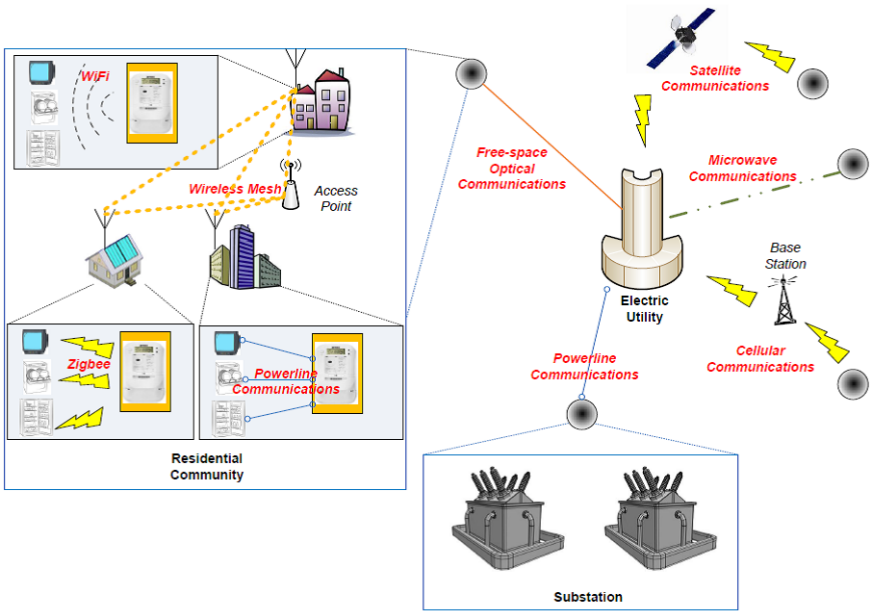


Figure 2.7 – Smart Grid architecture [11]

Intelligent Transport System (ITS) could enable us to calculate the best route in real-time by connecting different transport modes to save time and reduce carbon emissions.

2.7 Challenges and Open Issues

IoT systems rely on multiple technologies. Thus, the challenges and issues of IoT could be divided into two categories:

1. Issues related to the technologies on which IoT is based on.
2. New issues that emerge with IoT deployments.

This section presents the general IoT-related challenges and issues. For consistency purposes, the examples are mainly related to the smart city IoT applications (section 2.6). The main challenges for IoT systems are (Figure 2.8):

1. IoT Resource-Constrained Nature
2. Networking and Transport
3. Heterogeneity
4. Scalability
5. Big Data Management
6. Security

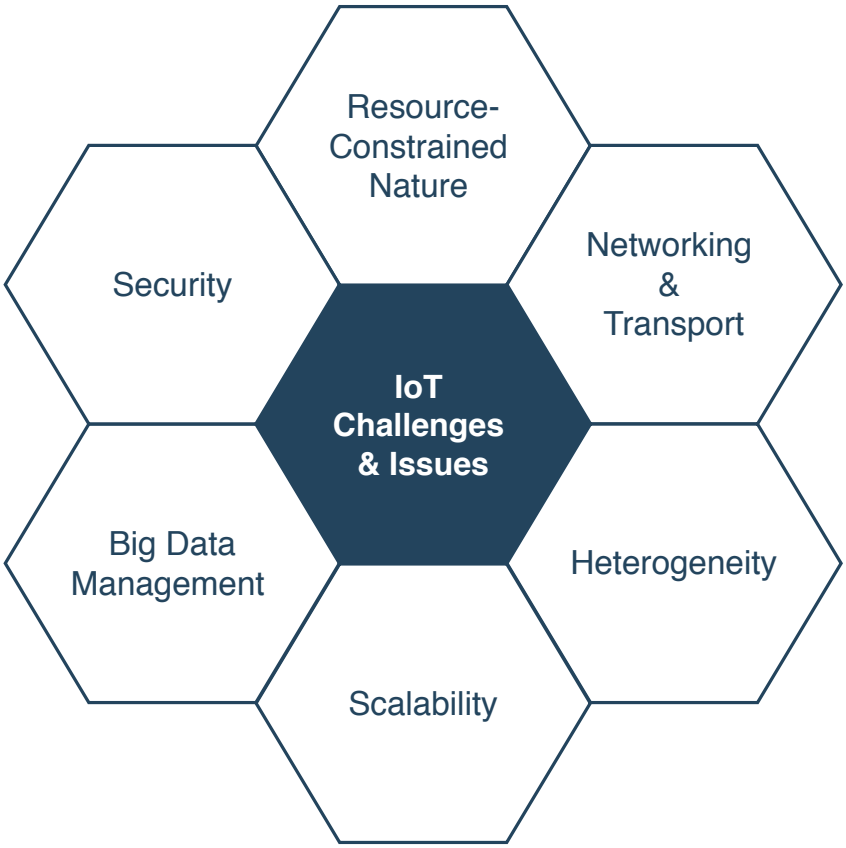


Figure 2.8 – IoT Challenges and Issues

IoT Resource-constrained Nature

One of the key characteristics of IoT nodes is their limitations in terms of processing, memory and energy. This intuitively implies that any "viable" IoT-based solution, should consider these constraints by-design. This also makes the performance and energy consumption analysis of such solutions critical elements for their acceptability at a first stage and their deployment in a later stage.

Networking and Transport

IoT includes a huge number of objects that should be reachable. Besides, each object will produce content that can be retrieved by any authorized user regardless of his/her location. To achieve this goal, effective addressing policies should be implemented. Currently, IPv4 is the most predominant protocol. However, it is well-known that the number of available IPv4 addresses is decreasing rapidly and IPv4 will soon become inadequate in providing new addresses, hence, the need to use other addressing policies. IPv6 addressing represents the best alternative to IPv4. Many works that aim to integrate IPv6 with IoT have been undertaken recently. For example, 6LowPAN [76] describes how to implement IPv6 protocol in a WSN

context. However, since RFID tags use identifiers rather than MAC addresses (as standardized by EPC global [77]), it is necessary to propose new solutions in order to enable the addressing of RFID tags in IPv6-based networks. Recently, multiple studies that intend to integrate RFID tags into IPv6 networks have been investigated and multiple approaches aimed at integrating RFID identifiers and IPv6 addresses have been proposed [78][79][80]. However, results in this area are not completely mature and in particular there are no standards that currently describe how this integration should be done. It is also important to note that RFID mobility is not supported and still represents an open research issue.
In traditional networks, IP addresses are resolved through the Domain Name System (DNS). In IoT, communications occur between objects. Thus, the concept of Object Name Service (ONS) must be introduced and supported [81] [82] [83]. The difficulty of ONS arises especially in the case where the object is an RFID tag. In this case, the tag identifier (or IP address) is mapped onto an Internet Uniform Reference Locator (URL), which points to the relevant information of the object. In other cases, the ONS must have the capacity to associate the object's description with a given RFID tag identifier (or IP address). However, the design and standardization of such a system are still being investigated by researchers and designers of such systems [83].
The main goals of the transport layer reside in guaranteeing end-to-end reliability and in performing congestion control. In traditional networks, the Transmission Control Protocol (TCP) supports these goals. However, it is known that TCP is not adapted to IoT environments [83] [84] for many reasons:

1. Connection setup: in TCP, each session begins with a connection phase procedure called the three-way handshake. Within the IoT ecosystem, a small amount of data will be exchanged. Therefore, the setup phase would last for a large part of the session time. This may lead to additional consumption of resources and energy.

2. Congestion control: TCP ensures end-to-end congestion control. In the IoT context, it can generate performance problems as most of the communications are wireless. Indeed, such an environment is not well optimized for TCP [85]. Besides, the exchanged data amount within a single session, is in general, very small. Finally, TCP congestion control is not very adapted to the IoT environment because the whole TCP session includes just the transmission of the first segment and the reception of subsequent acknowledgements [83] [38].

3. Data buffering: TCP stores data in a memory buffer at both source and destination: (1) at the source for re-transmission needs and (2) at the destination for ordered delivery purposes. The management and allocation of such buffers may be too costly for objects.

As a result, TCP cannot be used efficiently for the end-to-end transmission control in IoT and new transport layer protocol solutions are required [38]. The transport

layer plays an essential role in IoT. Indeed, attacks towards this layer and its underlying routing protocol will seriously affect the network's operation. Therefore, the design of secure and effective routing protocols is an important research area in the IoT context. Due to typical characteristics of IoT objects, existing solutions that have been previously applied to adhoc and sensor networks do not completely address the needs of IoT. For example, Denial of Service (DoS) attacks could be more easily achieved on multiple IoT systems. The consequences of such attacks would be disastrous to the systems and their end-users. The best way to detect and stop DoS and DDoS attacks is by using Intrusion Detection Systems (IDSs). However, the implementation of such systems in an IoT infrastructure appears to be a very challenging task because of the specific characteristics of the objects and their capabilities.
Another important issue is traffic characterization. Indeed, in IoT, highly heterogeneous objects lead to different scenarios. The characteristics of the related traffic flows generated by these scenarios have not really been studied extensively [83]. The traffic's characterization represents a very important step, because it helps network providers to plan the expansion of their infrastructures when it is needed, and to develop appropriate solutions for Quality of Service (QoS) support when needed.

Heterogeneity

Often, in IoT scenarios, data is collected from large number of objects which are widely distributed. However, the data collected in different ways using different protocols typically have different formats.
Thus, it is not possible to effectively analyze, process, store such data without some standard format. This lack of standard also makes the integration of data obtained from heterogeneous sources difficult. Thus, it is necessary to develop

1. Standards regarding unified data encoding
2. Information exchange protocols that will enable efficient and seamless data collection among heterogeneous IoT objects

Scalability

The huge number of Internet objects that need to be deployed in the different applications (section 2.6) pose a scalability challenge for any IoT-based solution. The scalability can be directly linked to a critical issue/ threat which is the denial of service. Indeed, scalability issues arise often with centralized solutions which exhibit single point (or few points) of failure.

Big Data Management

As the number of devices grows exponentially, IoT-based solutions become a source of huge amounts of data often referred to as big data [86] [87]. Indeed, according to the literature [88] [86] [89], big data is characterized by specific characteristics:

- Volume: the huge number of devices continuously generate large amounts of data.
- Dynamicity : for many applications data is created and used in real or near real-time. For example, traffic data must be used in real time to inform users and to guide them [90]. Another example is social media, where, sometimes messages, tweets, status updates, and so on which are only a few seconds old may no longer be of interest to users.
- Variety: there are multiple types of devices, parts of different applications that are communicating through various protocols that generate a lot of heterogeneous data.

An efficient use, mashing and correlation of these different types of data can enhance multiple applications and tasks such as:

- Facilitating decision making in order to enhance the quality of service offered to end-users.
- Visualization and simulation of events and use cases.
- Modeling new use case scenarios.
- Accidents and disaster management.

For example, if an accident occurs, it will be reported to the management infrastructure (center) through traffic information. Then, the management center sends the nearest police cars (having lower task priorities according to their transmitted information) and requests for ambulances. Besides, the center sends accident warnings to car drivers in the accident area through car's Human Machine Interfaces (HMI) and road advertisement boards, and recommends journey modifications to cars going through the accident area. It can also modify traffic flows and lights to facilitate ambulances' tasks. The same scenario is possible in case of fire where the management center will be informed through dedicated devices.
However, none of the aforementioned benefits can be achieved without an efficient way to manage and leverage such large quantities of data generated and large-scale infrastructures. Currently, the most popular technology is cloud computing [88] [91] [92] [93]. Indeed, cloud computing solutions help in the storage, visualization, and processing of the data collected in order to make timely inferences and decisions [92].

Security

The rapid integration of IoT in various aspects of our daily lives and their use in critical infrastructures makes the security of IoT crucial and even critical given that its insecurity can be life-threatening. The IoT systems are becoming a primary choice for attackers whether as a target or even as an attacking tool. The latter

is no more a hypothetical threat model since IoT systems are becoming the ideal target for hackers to create IoT bot networks. An IoT botnet consists of devices compromised and used to perform different tasks without the knowledge of their legitimate users.
In 2016, Dyn firm[11] suffered from a denial of service attack caused by tens of thousands of connected objects (they were mostly connected cameras) to saturate its infrastructure[12]. The attack resulted in Dyn's inability to provide the DNS service. Some of connected objects involved in the attack were caused by Mirai malware [17]. This tool exploits vulnerabilities present in some connected objects such as the use of default passwords that have not been changed by users. Hence, IoT networks are increasingly being used as an attack platform by malicious attackers.
The security challenges, issues and requirements will be extensively detailed in chapter 3.

2.8 Conclusion

This chapter provided a comprehensive overview of the Internet of Things (IoT) including the proposed architectures in the literature, the advances of research and development efforts, the main IoT technologies and platforms used and deployed in the market and finally the different IoT use case applications.
The chapter concluded with a presentation of the main challenges facing the IoT systems related to their resource-constrained nature, the networking and transport, the heterogeneity of devices, the scalability of the IoT solutions, the management of the huge data generated and stored in the IoT systems and finally and most importantly the security of such systems. The latter challenge is the field of interest of this thesis, in particular the authentication of the "things" in IoT which will be further discussed in the next chapter.

[11] Dyn is one of the companies providing DNS service
[12] https://krebsonsecurity.com/2016/10/hacked-cameras-dvrs-powered-todays-massive-internet-outage/

Chapter 3

IoT Authentication Schemes

3.1 Introduction

The security of IoT is a major challenge for the sustainability and competitiveness of companies and administrations. The US Federal Trade Commission (FTC) pointed out in a report [94] that the planned deployment of IoT technology will open up various security and privacy issues for IoT users, which need to be well addressed or resolved. For many of these critical IoT applications, the use of incorrect or maliciously corrupted data can have serious consequences. Conventional security objectives such as authentication, confidentiality, and data integrity are critical to IoT objects, networks, and applications.
Moreover, the IoT security context is more complex due to the inherent nature of IoT systems. Indeed, objects are most of the time unattended, which makes physical attacks on them relatively easy. In addition, Most of the communications are wireless, which makes the system more exposed to attacks; exchanged messages may be subject to eavesdropping, malicious routing, message tampering and other security issues which can affect the security of the entire IoT system. Also, Multiple types of objects such as RFID tags have limited resources in terms of energy and computation power, which prevents them from implementing well-established traditional security solutions.
The IoT security "scene" becomes more complex, when considering the current security context of IoT devices. There is a clearly identifiable gap in the security standards for such objects. Indeed, according to the European Union Agency for Cybersecurity (ENISA), "standards are not treated holistically so it is possible to deliver a device to the market that can authenticate its user, that can encrypt data it transmits, that can decrypt data it receives, that can deliver or verify the proof of integrity, but which will still be insecure. Similarly, the organisation developing the IoT product or service may have the development processes defined in management guidelines such as those of ISO-27000 but still delivers an insecure product" [95]. Moreover, Various types of operating systems are used by the connected objects, which are not always well known or validated in term of security. The code of

an operating system is usually in the order of tens of thousands or millions of code lines. Hence, the likelihood of having vulnerabilities is high. In addition, the software integrity update of connected objects is not guaranteed.

If IoT objects have enough memory and processing power, existing security protocols and algorithms might have been applicable, but because of the resource constraints of IoT objects, these existing security solutions are too costly for the objects in IoT.
Security issues remain major obstacles to the worldwide adoption and deployment of IoT. In other words, users will not fully adopt IoT if there is no guarantee about the related security concerns.

This chapter examines the general security context of IoT systems with particular emphasis on authentication-related issues and solutions. It starts with a presentation of the IoT security objectives and issues. Then, it gives a particular attention to the research efforts done to address the authentication problem, where a classification of the existing solutions is provided as well as an analysis of the most-known solutions in the literature. The chapter concludes with a list of requirements for a strong authentication scheme.

3.2 IoT Security Objectives and Issues

IoT systems like any other information system have their own security objectives and related security issues. However, the ubiquitousness of such systems and their widespread range of applications make these issues more alarming and hence, the security objectives become more critical in the IoT context.

3.2.1 IoT Security Objectives

Depending on the intended application, an IoT system can have one or more of the following security objectives. Also, the type of the application dictates how critical a security objective can be.

Confidentiality

The property that sensitive information is not disclosed to unauthorized individuals, entities, or processes [96].
The confidentiality might be of high concern for IoT systems especially when connected objects are used for transmitting confidential data. This obviously depends on the nature of the IoT system; the confidentiality level for securing smart water sprinklers will definitely not be the same as that required to secure patient records stored in or exchanged between devices in healthcare IoT systems.

Integrity

The data stored in and transmitted by IoT devices might be sensitive; whether it was related to the users (e.g., bank details, social security number), the manufacturers or the network service providers (e.g., billing/ payment records, usage statistics, business secrets, cryptographic keys). Such data should be protected against intentional or unintentional alteration or deletion, which can have irrecoverable damages, where comes the importance of protecting the integrity of data/information.
Data and system integrity deals with consistency and reliability issues associated with the different components of IoT systems and their data and software resources [96]. In the IoT context, integrity can be defined as "maintaining the accuracy, consistency and trustworthiness of data in transit or stored on any IoT device" [97]. Protecting the integrity results in the assurance that information can only be altered (or deleted) by authorized entities.
Integrity of IoT systems not only considers protecting the IoT devices from tampering but also detecting any attempt to tamper with these devices whether logically or physically, especially that large number of IoT devices (e.g., sensors) are deployed in open environments without sufficient physical protection, allowing attackers to have direct contact with them.

Availability

Availability can be defined as "timely, reliable access to data and information services for authorized users"[96]. Availability in an IoT context requires implementing measures to guarantee the persistence of the connectivity of devices and services despite of any link failure, carrying out a regular preventive maintenance and update of all hardware and software, maintaining an immediate and adaptive recovery in worst case scenarios, storing back-up copies of data, etc. [97].
Availability is a default security requirement for all the IoT applications and at the different architectural layers (Perception, Transport and Application). However, the availability percentage might vary depending on the IoT applications and its criticality.

Authentication

Authentication can be defined as the verification of the identity of a user, process, or device [96]. The verification is based on a proof, called authentication factor, that can be based on [98, 99]:

1. Knowledge: Something the user (or any entity to authenticate) knows (e.g., passwords, pre-shared keys)
2. Ownership: Something the user has, i.e. the items that he/she possesses (e.g., smart cards)
3. Biometric: Something the user is; i.e. user's physiological and biometric traits (e.g., face, fingerprint, and voice)

4. Behavior: Something the user do; a type of authentication which proves identities by observing actions (e.g., gestures or touches).

5. Location: Somewhere the user is; i.e. the user's location information (e.g., GPS, IP address)

In an IoT context, all IoT devices must be able to prove their identity in order to be granted certain privileges of operation (e.g., send/receive information), or be allowed to access resources in an information system. The proof of identity (credentials) can be one or more of the previously mentioned authentication factors, and is often based on identical credentials stored in a safe location.
Authentication (and thus identity proof) can be done for the device as a whole, but can also be done for device parts, in order to ensure that no third party components with potential security risks are connected to the system [97].
Authentication concerns and existing solutions will be further detailed in section 3.4.

Authorization

Authorization can be defined as the right or a permission that is granted to a system entity to access a system resource. In the IoT context, an authorization scheme provides a mechanism to bind an IoT specific device to a number of permissions; e.g., accessing a resource such as sensor data or a file. This limits the privileges of devices, users, applications and components so that they can only access resources necessary for their normal operations [97].
Authorization is often linked to access control; the two concepts are closely related and even used interchangeably sometimes.

Privacy

A general definition of privacy can be obtained from [96] as "the right of a party to maintain control over and confidentiality of information about itself". With regards to IoT, privacy concerns are on the rise with the technology becoming increasingly intrusive in our personal lives. Many companies/businesses collected (and are still collecting) huge amounts of data about IoT users. The sources of the collection can be any node of the system (e.g., web cookies, video cameras, RFID bracelets, etc). The data collected and stored contain for sure a considerable amount of personal and sensitive information.
Privacy protection can be obtained either by trusting third parties not to abuse the data generated by IoT systems or by controlling the collection and use of such data. More specifically, in order to ensure privacy in the IoT domain, a user should have the capability (among other) to [100]:

- Find out what information is collected and shared, when, with whom and for how long.

- Control how the data/information is collected and shared with third parties.
- Determine (and control) how identifiable he/she is while using the IoT system.

These privacy aspects have thus strong implications of trust, transparency and control for IoT systems, and therefore introduce a number of privacy-related requirements, such as (un)linkability, (un)traceability and anonymity.

Unlinkability

From an attacker's perspective, "unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not" [101]. This requirement takes additional importance in the healthcare domain or in vehicular-related applications.

(Un)Traceability

A brief definition for the untraceability is to ensure that "no one knows content inside that data and no one knows where it is sent to" [102]. While untraceability can be a principal privacy-related requirement, traceability can be an application-related requirement such as in supply chain, food management, etc. [103, 104, 105, 106].

Anonymity

Anonymity of a subject means that "the subject is not identifiable within a set of subjects, the anonymity set"[101]. Anonymity can be a critical requirement in some healthcare IoT applications to conceal the identity of the patient and thus to ensure the privacy of his/her health information.

Non-Repudiation

Non-Repudiation is defined as "the assurance the sender of data is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the data" [96].
A simpler definition can be to prevent any user or device from denying an action, by providing evidence of such actions. For many IoT applications, non-repudiation is not considered as a primary security requirement. However, it becomes of critical nature for business applications that include financial transactions.

3.2.2 IoT Security Issues and Challenges

The security requirements presented in the previous section can be ensured using cryptographic algorithms. Confidentiality in particular (among other requirements) can be achieved throughout data encryption. Data encryption algorithms are divided into two categories: symmetric algorithms, and asymmetric or public-key algorithms. The latter are known to consume more resources which make them

difficult to implement on objects with limited power and energy resources.
In contrast, symmetric algorithms are suitable for such devices and are widely used in this context [107]. However, they suffer from several drawbacks: first, the symmetric key exchange protocols of such cryptosystems are too complex which limits infrastructure scalability [27], and second, they suffer from the confidentiality problem for shared keys. Indeed, the higher the number of objects is, the bigger is the security risk. If one key is compromised, all system communications are compromised also. As a solution, the system can be divided into multiple groups and a different symmetric key is used within each group. However, the risk remains, since if one key is compromised, the communications with the group are also compromised. To address this problem, researchers have considered public-key encryption algorithms. In this solution, each object owns a pair of public and private keys. Each object keeps its private key, while sharing its public key with corresponding entities with which it needs to communicate securely. Actually, the main proposals [108] [109] [110] of public key encryption algorithms suitable for IoT [107] include Rabin's Scheme [111], NtruEncrypt[112] and Elliptic Curve Cryptography (ECC) [113]. ECC offers good scalability, without complex key management protocol. However, the application of these algorithms to the IoT environment is still being investigated. In addition, they are not applicable to all types of objects especially RFID tags, where the problem of limited resources remains a challenging issue. Furthermore, public key encryption solutions suffer from trust issues. Indeed, a node that owns public keys cannot prove that the objects are really what they pretend to be.

Key management is another important issue in IoT. It plays a vital role in the implementation of various security solutions. Key management includes multiple steps that include key generation, distribution, storage, update and destruction. An important component of the key management cycle is key distribution which includes secure transmission and distribution to legitimate users of (1) public keys and shared secrets in the case of asymmetric cryptography, and (2) secret keys in the case of symmetric cryptography.
Numerous works [114] [115] [116] have proposed key management schemes adapted to technologies making up the IoT ecosystem, and more specifically for Wireless Sensor Networks (WSNs) in recent years. They use symmetric key management, public keys, abbreviated (a shortened certificate where some fields are removed) [117] or implicit certificates. However, these solutions were designed for WSNs primarily, and are not suited for all object types. Consequently, the design of lightweight key management schemes adapted to the IoT environment and its application scenarios remains a key issue that needs to be solved in the future.

IoT trust management mechanisms are also a challenging key element to be developed and implemented. Indeed, in numerous scenarios, the network relies on the cooperation of all nodes.
The vulnerability of a single node can have serious consequences on the entire

network. Indeed, if an attacker succeeds to compromise or add one or multiple objects in the network, the attacker can provide fake or erroneous information, which can subsequently affect the cooperation of nodes, data treatment and the result provided to the final user. Thus, the credibility of each single node is key to ensuring accurate and reliable network service delivery. Current trust management schemes like those proposed in [114] [118] only provide verification of data consistency and validity, but cannot guarantee object authentication. Furthermore, these previously proposed schemes are not completely adaptable to the IoT context. Consequently, more research is needed to develop lightweight trust management techniques and protocols that are specifically well suited for IoT scenarios in the future.

Finally, security issues and threats can arise from any of the security objectives previously presented in section 3.2.1, which can be further associated to the IoT architectural view. Table 3.1 summarizes the most common IoT issues that arise from the the three-layered IoT architecture as presented in section 2.2.

	Issues	***Confidentiality***	***Integrity***	***Authentication***	***Authorization***	***Unlinkability***	***Anonymity***	***Traceability***	***Non-Repudiation***
Application Layer	Applications' issues	x	x	x	x	x	x	x	x
	Storage issues	x	x	x	x		x		
	Key management issues	x	x	x			x	x	x
	Trust management issues		x	x	x				x
	Middleware issues	x	x	x	x	x	x	x	x
	Data integrity issues		x	x					
	Data confidentiality issues	x	x						x
	Data authentication issues		x	x					x
Transportation Layer	WIFI issues	x	x	x	x	x	x	x	x
	ZigBee issues	x	x	x	x	x	x	x	x
	3G issues	x	x	x	x	x	x	x	x
	GPRS issues	x	x	x	x	x	x	x	x
	Network access issues		x	x	x	x	x	x	
	Routing protocols issues		x	x			x		
	Encoding issues		x						
	Heterogeneity issues		x	x			x	x	x
Perception Layer	RFID issues	x	x	x	x	x	x	x	x
	WSN issues	x	x	x	x	x	x	x	x
	RSN issues	x	x	x	x	x	x	x	x
	GPS issues	x	x	x	x	x	x	x	x
	Platform issues	x	x	x	x	x	x	x	x

Table 3.1 – Overview of main IoT issues

3.3 Classification of IoT Authentication Solutions

A quick examination of Table 3.1 clearly shows that authentication is a security requirement at all the IoT layers.
Authentication plays a central role in the overall security of IoT systems. A secure IoT device authentication scheme ensures that these devices can be trusted to be what they claim to be. Such scheme provides each IoT device a unique identity (can be dynamic in time) that can be verified when this device attempts to connect to the IoT network. This gives the possibility, among other things, to track each device throughout its lifecycle (whenever applicable), to exchange securely with it, to prevent it from running malicious code, and even though it happened that an IoT device exhibited a suspicious unexpected behavior, its privileges can simply be revoked.
In this section, we present a general taxonomy of IoT authentication schemes based on the surveyed literature [119] [120] [121] as illustrated in Figure 3.1.

- **IoT layer**: It depends on the layer where the authentication method is implemented. In this classification, we consider the most common IoT architecture which consists in three layers (1) Perception layer, (2) Network layer or (3) Application layer. However, this classification can be adapted to any layered architecture for IoT (Section 2.2)

- **Application domain**: Different authentication schemes are used for the different application domain or IoT environment[119]: Machine to Machine (M2M), Internet of Vehicles (IoV), Internet of Energy (IoE), Internet of Sensors (IoS) and Internet of Medical Things (IoMT)

- **Hardware-based**: Hardware-based authentication uses the physical characteristics of the hardware to process the authentication. Based on this criterion, one can distinguish between (1) Implicit hardware-based solutions which use the "existing" hardware during the authentication (e.g., Physical Unclonable Function (PUF)[122] [123] or True Random Number Generator (TRNG)[124] [125]) and, (1) Explicit hardware-based solutions which require the use of additional component dedicated for the operations (cryptographic or other) performed during authentication (e.g., Trusted Platform Module (TPM)[126] [127], Trusted Execution Environment (TEE)[128] [129] [130]).

- **Authentication factor**: Depending on the number of factors considered for authenticating a device (section 3.2.1), a solution can be classified as a single-factor authentication (SFA), a two-factor (2FA) or a multi-factor authentication (MFA). For example, if only the user password is used, the authentication scheme is called single-factor scheme. A two-factor authentication scheme might use user password and smart card to authenticate users. A Multi-factor authentication might use additional factors such as location information, biometric, etc. [131]

- **User Access**: Different methods are used for user access; Some use the public key cryptography and can be either based on the use of digital certificates (Certificate-based) or only relying on the public/private key pair (Certificate-Less), other solutions rely on the use of Pre-Shared keys. Hybrid solutions mixing both uses also exist.

- **Cryptographic algorithms**: The cryptographic algorithms used during the authentication phase can also be used as a classification criterion. Some authentication mechanisms rely solely on symmetric algorithms given their low-overhead compared to asymmetric algorithms. Within this category of solutions, one can distinguish between authentication schemes using traditional symmetric algorithms and those using lightweight symmetric algorithms which were introduced for constrained devices such as IoT objects. Another category of solutions relies only on asymmetric cryptography during the authentication phase, and can be further divided into those using traditional algorithms (e.g., RSA) and those relying on Elliptic-Curve Cryptography (ECC). A third category of solutions rely on the use of hash functions given their lightweight nature. Finally, hybrid solutions mixing two or all the above mentioned methods also exist.

- **Authentication procedure**: The authentication procedure can be one-way, two-way or tripartite three-way authentication. In one-way authentication, only one of the two entities wishing to communicate will authenticate itself to the other entity. In a two-way authentication, the two entities wishing to communicate will verify each other. This type of authentication is also called mutual authentication. In three-way authentication, the two communicating entities are authenticated using a third entity called central authentication. The central entity authenticates the two entities using mutual authentication.

- **Authentication architecture**: Two Authentication architectures are used to process the authentication procedure, centralized and distributed (decentralized) architecture. In a distributed architecture, the two communicating entities are authenticated using a straight authentication. In a centralized architecture, the communicating entities are authenticated using a third trusted centralized entity, which shares and manages the credentials between the entities for the authentication procedure. In both architectures, the structure can be hierarchical or flat. Hierarchical used different levels and flat has no level structure. [121]

3.4 Analysis of IoT Authentication Schemes

Given the high importance of authentication in the IoT general security context, it was extensively addressed in the recent years [132, 119, 133, 134, 28] . This section provides a survey of the most known IoT authentication schemes in the literature.

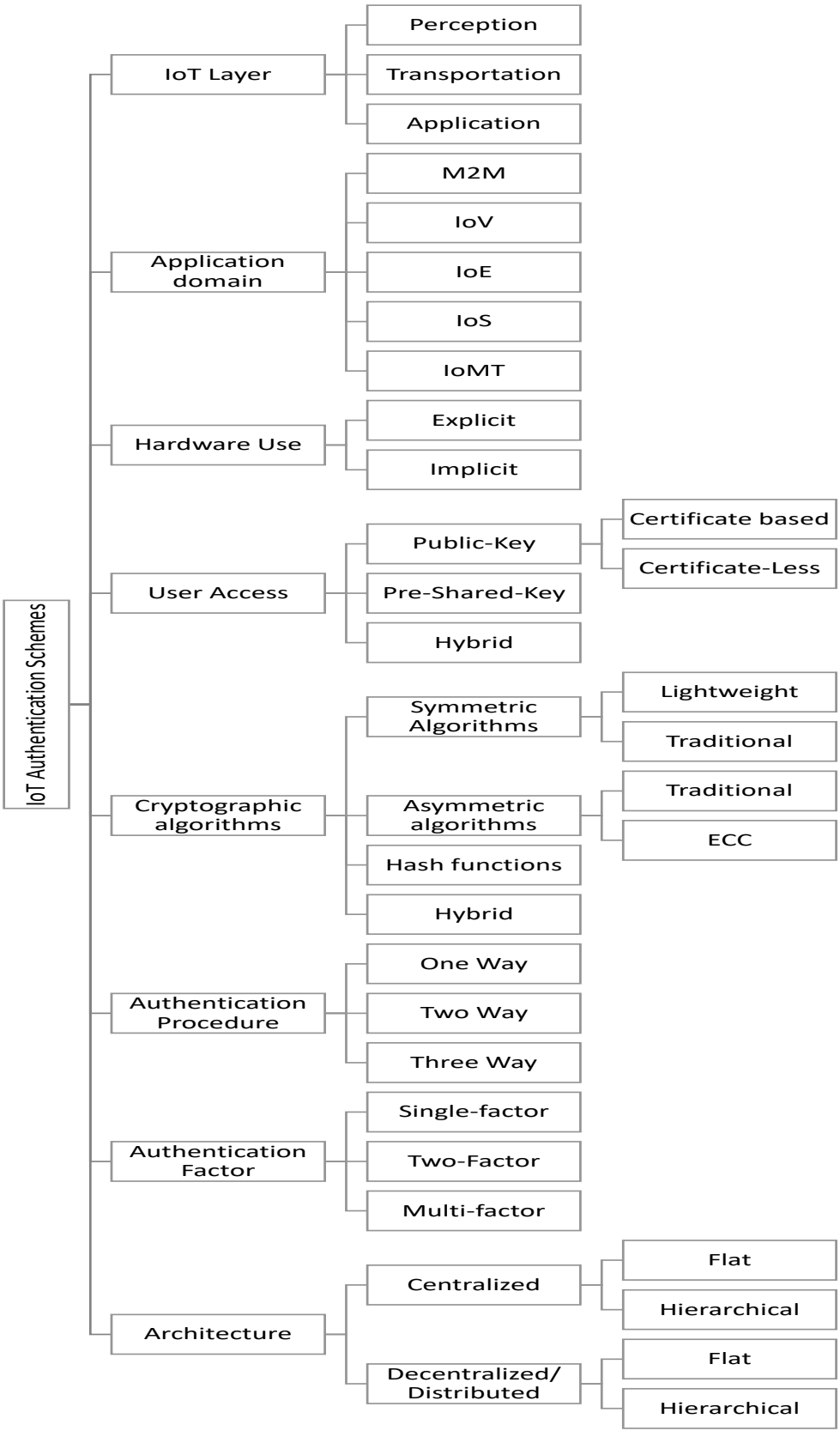


Figure 3.1 – Taxonomy of IoT Authentication Schemes

For organizational purposes, they are classified into two main categories: centralized and decentralized authentication schemes.

Centralized authentication schemes

As they name indicates, centralized authentication schemes allow the devices information to be stored on one host, minimizing the risk of security loopholes.

The authors in [135] proposed an implementation of two-way authentication security scheme for the Internet of Things (IoT) based on existing Internet standards, Datagram Transport Layer Security (DTLS), with IEEE 802.15.4 used for the physical and Media Access Control layer, and UDP for the Transport layer. In this approach, authors considered three security goals, authenticity, integrity and confidentiality. The authentication is performed during a fully authenticated DTLS handshake and based on an exchange of X.509 certificates containing RSA keys.
One of the main limitations of such solutions is the use of "heavy" X.509 certificates and their overhead in the context of IoT.

In [136], authors proposed a lightweight mutual authentication scheme which validates the identities of the participating devices before engaging them in communication for the resource observation. The proposed scheme had less connection overhead and provides a robust defense solution to combat various types of attacks. The authors chose the Constrained Application Protocol (CoAP)[19] as the underlying application layer protocol for enabling communication among various physical objects, which is based on the conditional resource observation. Authors also proposed to add security features for the authentication in the CoAP. The proposed authentication scheme is based on DTLS, which was not designed to be implemented in constrained environments. Moreover, DTLS use UDP as a transport protocol which is considered as unreliable, and using certificate is not apt for constrained devices. Finally, although the proposed scheme is an effective solution against eavesdropping, key fabrication, resource exhaustion and denial of service attacks. However, it is not efficient against Sybil attack.

A secure and efficient authentication and authorization architecture for IoT-based health-care systems, is proposed in [137]. The authors proposed an architecture relying on the certificate-based DTLS handshake protocol. This architecture exploits distributed smart e-health gateways to perform authentication and authorization of remote end-users securely and efficiently on behalf of medical sensors. A gateway supports different wireless protocols and inter-device communication and has a local database, where it can temporarily store medical sensors' information and provide local processing of medical sensors'. The smart e-health gateway and the remote end-user must have sufficient resources in order to perform various heavy-weight security protocols as well as certificate validation efficiently. The proposed

architecture has one major drawback, which is inherent in the centralized nature of the architecture: the denial of service attacks. The server can be compromised easily in a DoS attack, which allows an attacker to access and retrieve all available stored data in the constrained medical domains. Another drawback is the issue of privacy in IoT-based healthcare applications [138].
In [139], the authors proposed a secure Elliptic Curve Cryptography (ECC) based mutual authentication protocol for secure communication of embedded devices and cloud servers, using Hyper Text Transfer Protocol (HTTP) cookies. ECC-based algorithms give better security solutions in comparison to other Public Key Cryptography(PKC) algorithms due to small key sizes and efficient computations. Although the authors claimed a low computation cost, but the embedded device needs to be configured with TCP/IP protocol stack in order to act as a HTTP client, with HTTP being a heavy protocol to be used for the constrained devices. Moreover, the proposed scheme is susceptible to offline password guessing and insider attacks and it does not achieve device anonymity, session key agreement, and mutual authentication [140].

The authors of [141] proposed a protocol , called TinyTo, to ensure the end-to-end security with two-way authentication based on Public Key infrastructure (PKI). In the authentication phase, a number of handshake messages are exchanged between the end device and the server, for generating a session key to be used for upcoming communication. The main drawback of the proposed solution is the high memory consumption for certificate authority (and other PKI-related) operations. Moreover, the authors did not address the security of their scheme against replay and denial of service attacks [28].
in [142], the authors proposed set of lightweight authentication and authorization mechanisms in order to support smart objects during their life cycle. During the authentication phase, the device authenticates using Extensible Authentication Protocol (EAP) with a Remote Authentication Dial-In User Service (RADIUS) server, and an authorization server which takes authorisation decisions, to generate authorisation token for the device.The proposed authentication approach is based on EAP over LAN (EAPOL) for security bootstrapping in order to establish keys for DTLS. This constitutes an overload for the constrained device with the necessity to implement and execute the EAPOL protocol in addition to DTLS, an approach which seems to be not well adapted to constrained devices [143].
The authors of [144] proposed a lightweight authentication scheme based on nonces and Keyed-Hash message authentication (HMAC) function to check the integrity of authentication exchanges in healthcare IoT systems. This scheme propose a mutual-authentication between sensors and base station (BS).
In [145], the authors proposed a lightweight authentication mechanism, based only on hash and XOR operations, for M2M communications in Industrial IoT environment. The proposed mechanism is characterized by low computational cost, communication and storage overhead, while achieving mutual authentication, session key agreement and device's identity confidentiality.

The One Time Password (OTP) mechanisms took a lot of attention in the applied cryptography literature [146, 147]. Multiple works highlighted the different strengths and weaknesses of the different schemes used for OTP.
The authors of [148] showed how RSA based cryptosystems were broken. They also proved that RSA-based OTP systems and SecureID token algorithms are vulnerable to numerous attacks. Florencio *et al.* [149] proposed an approach which allows OTP access to any web account, without any change to the server or the client, and without storing user credentials in the cloud. However, their solution relies on a set of pre-shared keys and its implementation is very complex, which makes it suitable only for limited contexts. Lee *et al.* [150] proposed an insider attack-resistant OTP scheme based on bi-linear pairing operation. However, this approach is also computationally complex and it is not suited for the IoT environment.
In [151] the authors proposed an OTP scheme that calculates an OTP by relying on the previous OTP. However, this scheme suffers from initialization problems.
Although OTP has been extensively studied over the years, there are only few studies that have investigated the integration and usage of OTP in IoT due to objects' constraints. Nonetheless, currently, the objects are becoming increasingly powerful in terms of their resources thereby removing some of the limitations associated with OTP use.
In [152], the authors proposed Sturdy OTP (S-OTP), a connection-less authentication mechanism for mobile devices which does not rely on SMS. S-OTP's operation incurs high overheads because it requires a large number of communication exchanges. Furthermore, it is adapted only for smart phones.
Shivraj *et al.* [1] proposed an efficient lightweight OTP scheme based on Identity Based Elliptic Curve Cryptography (IBE-ECC). However, this approach relies on a simple Diffie-Hellman key exchange and pre-shared keys. Moreover, during the key generation, the problem of the point at infinity can occur because there is no method that can prevent it.
In [2] [153], the authors proposed a robust and lightweight mutual authentication protocol, designed especially for IoT limited capacity devices. Their personalization mechanism limits damages if one device is corrupted thereby protecting the rest of the device's cluster. However, their approach is based on an asynchronous OTP type, which requires challenge/response management operations by the server.

Decentralized authentication schemes

There have been numerous works that tackled authentication and authorization in IoT in a decentralized way. In particular, in the last years, many works aimed the proposal of IoT security systems that rely on blockchains. Indeed, blockchain usage provides a decentralized architecture, anonymity of users if wished, tamperproof record of data transactions and highly trusted data verification [154].

Dorri *et al.* [154][155] propose a blockchain based architecture for IoT. Their approach relies on three interconnected blockchains: a local blockchain (private) for each use case, a shared blockchain (private) and an overlay blockchain (public).

Even if the solution resolved the problem of identification, it still has multiple shortcomings like (1) each operation requires at least 8 network communications which can flood quickly the whole communication medium in case of high activity of nodes; and (2) the local blockchains are not distributed but centralized which is contrary to its principle because it can limit its power and availability.

Hardjono *et al.* [156] propose *ChainAnchor*, a privacy-preserving method for commissioning an IoT device into a cloud ecosystem. ChainAnchor supports device-owners being remunerated for selling their device sensor-data to service providers, and allow device-owners and service providers to share sensor-data in a privacy-preserving manner. However, Its goal is the full anonymity of the participating devices and is not adapted to numerous IoT use cases where the identification is needed.
In [157], the authors propose a robust, lightweight and energy-efficient security protocol for the WSN systems that rely on blockchains. However, this work was proposed for OCARI [2], a very specific WSN architecture.
The authors of [158] propose a decentralized authentication and access control mechanism for lightweight IoT devices, which can be applicable to a large number of scenarios. The proposed mechanism is based on the technology of the fog computing and the concept of the public Blockchain. The main limitation of the proposed scheme is the huge amount of energy consumption required by the Proof of Work (PoW) consensus mechanism to verify each block.
The authors of [159] proposed a distributed authentication system named DecAuth (Decentralized Authentication) using the Blockchain technology The implementation of the proposed authentication is done on Ethereum platform. The authors also provided an analytical security evaluation (no formal security analysis has been done).
In [160], the authors proposed a decentralized Public Key Infrastructure (PKI) for IoT, called IoT-PKI, which utilizes distributed nodes in a blockchain network instead of Certification Authorities, in order to addresses scalability issues of traditional PKIs. IoT-PKI protects against key leakages at device manufacturers since it allows the owners of IoT devices to manage the certificates of their IoT devices.
A blockchain-based Trust and Reputation System (TRS) for IoT access control is proposed in [161]. The proposed system progressively evaluates and calculates the trust and reputation score of each participating node to achieve a self-adaptive and trustworthy access control system. Trust and reputation are explicitly incorporated in the attribute-based access control policy, so that different nodes can be assigned to different access right levels, resulting in dynamic access control policies. The authentication of the different IoT nodes is achieved in the proposed system through the use of public key cryptography (i.e. digital signature). The solution was implemented in a private Ethereum blockchain, and benchmarked using various performance metrics to highlight its applicability for IoT contexts.

A decentralized blockchain-based security framework for administering security in WSN-IoT communications is presented in [162]. The framework achieved both device and network level of security through device verification and message authentication. It uses tree-based hash along with conventional cryptography to retain communication security. It also use hash pruning to reduce the computation complexity during the message authentication process, in order to adapt to the resource constraint of the IoT devices/nodes.

The authors in [163] presented the main issues related to the use of blockchain with IoT devices (computation and bandwidth overhead). They proposed a framework of modified blockchain models suitable for IoT devices that rely on their distributed nature and other additional privacy and security properties of the network. The framework, mainly addressing healthcare applications, uses a hybrid approach that combines public key cryptography, blockchain and many other lightweight cryptographic primitives. It aims to develop a patient-centric access control for electronic medical records, capable of providing security and privacy.

A fully decentralized anonymous authentication protocol is presented in [164], which aims at encouraging the implementation of privacy-preserving IoT target-driven applications. The system is set up by an adhoc group of decentralized founding nodes. It does not rely on any central organization, not even for the set-up phase, which means that the parameters required by the system are not generated centrally but cooperatively amongst all the nodes in the system.

The authors of [165] proposed BlendCAC, a blockchain-enabled decentralized capability-based Access Control for IoT security. The BlendCAC aims at an effective access control processes to devices, services and information in large scale IoT systems. The authentication part of the framework relies on public key cryptography. During a registration phase, all entities must create at least one main account defined by a pair of keys to join the blockchain network. Each account is indexed by its address that is derived from his/her own public key. Account address (which is unique) serves as a virtual identity for the identity authentication and management.

The authors of [166] proposed a blockchain-based data security architecture for IoT networks. Their work differs from other blockchain-based solutions in that it proposes a hierarchical structure of blockchain (blockchain of blockchains) to overcome the resource problems in IoT. Nevertheless, the computational complexity is still high regarding the consensus algorithms and other blockchain-related operations.
Another blockchain-based solution is presented in [167], to ensure validity and integrity of cryptographic authentication data and associate peer trust level, from the beginning to the end of the sensor network lifetime. The model uses the blockchain to store decentralized authentication and node trust information. This model is

evolutive, adaptative and ensure reliability over time.
The authors of [168] proposed a blockchain-based solution for authentication and secure communication to IOT devices. The authentication module is based on the Ethereum smart contracts. The proposed solution is promising regarding its resistance against different types of attack but it is still not evaluated in terms of scalability especially with regard to the computational overhead related to blockchain operations.
The authors of [3] proposed bubbles of Trust, an efficient decentralized authentication mechanism. The scheme creates secure virtual zones (bubbles) where things can identify and trust each other. The communication (transactions) between different objects is controlled and validated by the public blockchain implemented using Ethereum.

3.5 IoT Authentication Challenges and Open Issues

The literature review led to the identification of a number of open-issues that need to be addressed to fill the gaps in the existing authentication schemes; some of these issues might have been addressed in the literature but in an incomplete way:

1. Most of the existing authentication schemes do not consider the additional requirements of the IoT systems (e.g., confidentiality); their role "stops" once the authentication is achieved and thus require additional solutions and technologies to be deployed, which introduces in some cases high computational costs. A clear example is the exchange of keys for confidentiality and integrity reasons; the existing authentication schemes which includes some key exchange don't help in exchanging the additional keys (confidentiality or integrity) and thus the key exchange for such keys needs to start from zero.

2. In general, the gateway and its related nodes must use the same authentication method (e.g., user access, specific algorithms, etc.). This lack of support for heterogeneous systems is a main limitation since, in numerous cases, the network is composed of different types of devices, each with its computation and energy capacities. For example in a smart home, if some objects, such as heat sensors, can only use PSK-based authentication, since symmetric cryptography is less costly than other methods [29], other more powerful objects, can easily use public key cryptography based methods. Nonetheless, they still have to use PSK based on the homogeneity constraint imposed by the gateway.

3. Each device is often associated to one gateway. However, in some use cases, some nodes can have mobility features (a sensor whose location is changed or any other mobility case). To the best of our knowledge, nodes that have mobility features from a gateway to another are not authenticated at this gateway level but using a central server/service accessible through Internet (e.g., Intelligent Transportation Systems). This lack of efficient mobility support in authentication schemes costs in term of performance, since (1) realizing

authentication at the gateway level can be less time consuming, and (2) relying on centralized services in a large and continuously expanding environment such as IoT constitutes a real bottleneck.

4. Many of the authentication schemes includes a compulsory initialization phase for newly added devices where a physical intervention on the gateway is needed to set up and configure the credentials of these devices (e.g., installing PSKs, public keys) which is not a scalable choice.

5. The architecture-dependent nature of authentication schemes which makes them in-adaptable to other IoT architectures.

6. The centralized nature of some authentication systems which creates single or few points of failure and makes the scheme not scalable.

7. The use of traditional asymmetric algorithms or X.509 certificates which requires high computational cost for resource-constrained IoT objects.

3.6 Conclusion

This chapter drew the general IoT security context and shed the light in particular on the IoT authentication field. It analyzed the general security objectives, issues and challenges for the Internet of things, and focused on those related to the authentication objective through a comprehensive survey of the most important research works in the literature.
The literature review led to the identification of a number of open-issues that need to be addressed to fill the existing gaps in the literature.
The next chapter presents the first contribution of the thesis which is a lightweight IoT Authentication Scheme based on Elliptic Curve Cryptography (ECC).

Part II

Contributions

Chapter 4

A lightweight ECC-based Authentication Scheme for IoT

4.1 Introduction

The previous chapter shed the light on the multiple risks and privacy issues raised by the increasing deployment of IoT technologies. Due to hardware limitations of IoT objects, implementing and deploying robust and efficient security and privacy solutions for the IoT environment remain a significant challenge.

One-Time Password (OTP) is an authentication scheme that represents a promising solution for IoT and smart cities environments.
In the first contribution of the thesis, we extend the OTP principle and propose a novel approach of OTP generation that relies on Elliptic Curve Cryptography (ECC) and Isogeny in order to ensure IoT security.
We evaluate the efficacy of our approach with a real implementation and compare its performance with two other approaches namely, Hash Message Authentication Code (HMAC-based) One Time Password (HOTP) and Time-based One Time Password (TOTP). The performance results obtained demonstrate the efficiency and effectiveness of our approach in terms of security and performance.
This chapter provides the background information needed for understanding the contribution by introducing the cryptographic elements used in the proposed scheme (Elliptic Curves, Isogeny, OTP and its variants). Then, it provides an overview of the proposed authentication protocol. In addition, it describes in details the performance evaluation framework and presents and analyzes the obtained results with a comparison to related works.

4.2 Cryptography Elements

This section presents the cryptography background needed for the rest of this thesis. In particular, it focuses on the Elliptic Curve Cryptography (ECC), a technique

based on elliptic curve theory that aims to create faster, smaller, and more efficient cryptographic keys. The second part of this section introduces the concept of One-Time Password (OTP), which is as its name indicate a password to be used once (e.g., one login session or one transaction) on a digital system. The advantage of OTP is that it avoids a number of shortcomings that are associated with traditional (static) password-based authentication. Two OTP-generation algorithms are presented: HMAC-based OTP (HOTP) and Time-based OTP (TOTP).
The understanding of ECC and OTP is necessary to understand our first contribution.

4.2.1 Elliptic Curve Cryptography (ECC)

Cryptographic System (also known as cryptosystem) can be classified into symmetric (or secret-key) and asymmetric (or public-key) cryptosystem.
In symmetric cryptosystem, the same key is used for encryption and decryption of the messages. This key needs to be shared and kept secret between the communicating parties. This constitutes the main issue for symmetric cryptosystem: the key distribution over an insecure channel. In addition, the number of keys to be shared increases quadratically with the number of users in the system.
In asymmetric cryptosystem, each communicating party owns a pair of keys; one private only known by the owner and a public key that is made public. This solves the problem of key distribution and management for symmetric cryptosystem.
One of the main issues for asymmetric cryptography is the large key sizes needed to achieve a certain security level[1] compared to symmetric algorithm. For instance, the famous Rivest–Shamir–Adleman (RSA) asymmetric algorithm needs to work with keys of 3072 bit length to offer the same security level of the Advanced Encryption Standard (AES) symmetric algorithm operating with 128-bit keys.
Elliptic Curve Cryptography (ECC) is a member of the family of established public-key cryptographic algorithms. ECC provides the same level of security as RSA with considerably smaller key sizes (approximately 160–256 bits vs. 1024–3072 bits)[170]. This means that, for the same level of security, significantly smaller parameters can be used in ECC than RSA [171]. For example, to achieve 112 bits of security level, RSA algorithm needs a key size of 2048 bits, while ECC needs a key size of 224 bits [172] as shown in Table 4.1. The importance of ECC, particularly in IoT context, is that the smaller key size (with high security level) is synonymous to lower computing power and battery resource usage, which is inline with the challenges for IoT-based solutions.

[1] Definition: "a cryptographic system offers security level λ if a successful general attack can be expected to require effort approximately 2^{λ}"[169]

Security Bit Level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	34
256	15360	512

Table 4.1 – NIST Recommended Security Bit Level

4.2.1.1 Definitions and Basics

Definition 1. *In cryptography, an Elliptic Curve (EC) is defined over a finite field* $\mathbb{Z}_p$*, as a set of points* (x, y) *satisfying the equation*

$$y^2 = x^3 + ax + bmodp \tag{4.1}$$

along with an imaginary point at infinity, denoted O[2]*, where* a, b *denoted as the coefficients of the curve* $\in \mathbb{Z}_p$*, and* $4a^3 + 27b^2 \neq 0$ *mod* p

$$\mathrm{E} = \{(x, y) : y^2 = x^3 + ax + b\} \cup \{O\}$$

The following operations are defined for the elements (points) of the elliptic curve:

1. **Addition**: Given P, R two points on the EC. To find a point $Q/\ P + R = Q$, draw a line joining P & R on the EC, this line cuts the curve in a point Q' (also denoted as $-Q$). Q is the symmetric of Q' with respect to the x-axis (Figure 4.1[3]).

2. **Point doubling**: Given P, point doubling means finding the point $Q = 2P = P + P$. In this case, the same "geometrical" operation is done but the secant becomes a tangent (Figure 4.1)

3. **Multiplication:** Scalar multiplication is the operation of successively adding a point along an elliptic curve to itself repeatedly.
   Given a scalar n and a Point P, $nP = \underbrace{P + P + \cdots + P}_{ntimes}$

The Point at "infinity" O is the Neutral point for the addition; $P + O = P$ for all points P on the EC.

Elliptic curve algorithms work in a cyclic subgroup of an elliptic curve over a finite field. Cyclic subgroups are defined using a generator point G and an order n. A generator point of the Elliptic Curve E , is defined in equation 4.2).

$$\begin{cases} \mathrm{E} =< \mathcal{P} > \\ \forall Q \in \mathrm{E}\ \exists n \in \mathbb{N}\ such\ as\ Q = n\mathcal{P} \end{cases} \tag{4.2}$$

[2]sometimes denoted ∞
[3]https://www.smalsresearch.be

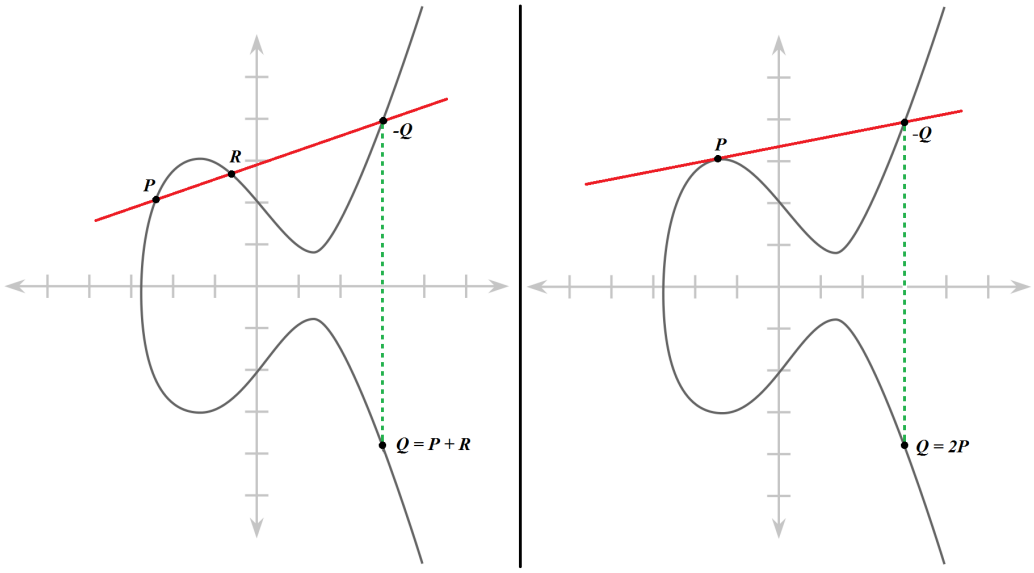


Figure 4.1 – ECC Point Addition and Doubling

The order is the smallest positive number n such that $nG = O$ (O being the point at infinity).
An elliptic curve group is thus completely specified by the following parameters (called Domain parameters) [173]:

- The prime number p that indicates the order of the field $\mathbb{F}_p$.
- The value a used in the curve equation.
- The value b used in the curve equation.
- The generator G of the subgroup.
- The order n of the subgroup generated by G.

An additional domain parameter h can also be used to define an elliptic curve which represents the number of points on the elliptic curve divided by n. Hence, the EC is generally represented by the following tuple (p, a, b, G, n, h)
The elliptic curves commonly used in cryptographic applications are those recommended by the National Institute of Standard and Technology (NIST)[174].

The security of the Elliptic curve cryptography is based on their Discrete Logarithmic Problem (DLP) called ECDLP.

Definition 2. *Given an Elliptic Curve E, a point P and another point T. The Discrete Log problem is finding the integer d where $1 \leq d \leq \#E$ such that* $\underbrace{P + P + \cdots + P}_{dtimes} = dP = T$

4.2.1.2 Cryptographic Operations and Protocols using Elliptic Curves

The generation of key pair on Elliptic curve is simple: The private key is a random number ($d \leq n$). The public key is a point in the elliptic curve, obtained by multiplying the private key d with the generator point G in the curve.
"Traditional" cryptographic algorithms were also "adapted" to be used for Elliptic Curves. A typical example is the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol. The ECDH protocol works as follows (Figure 4.2):

1. Sender and receiver agree on parameters (p, a, b, G, n, h).
2. Sender and receiver generate their private key d_S and d_R, which are random integers less than n.
3. The public keys for sender and receiver are respectively $e_S = d_S G$ and $e_R = d_R G$.
4. Sender and receiver exchange their private keys.
5. Sender computes the point K on the EC with coordinates $(x_K, y_K) = d_S e_R$.
6. Receiver computes the point L on the EC with coordinates $(x_L, y_L) = d_R e_S$.
7. Given $d_S e_R = d_S d_R G = d_R d_S G = d_R e_S$, Hence, $K = L$ and $x_K = x_L$.
8. The shared secret is x_K.

Additional EC-based cryptographic schemes include Elliptic Curve Digital Signature (ECDSA) Algorithm [175], Elliptic Curve Integrated Encryption Scheme (ECIES) [176], Edwards-curve Digital Signature Algorithm (EdDSA) [177], etc.

4.2.1.3 Isogeny on Elliptic Curves

To define an Isogeny, one has first to define morphism between Elliptic Curves and some properties:

Definition 3. *Let* E_1 *and* E_2 *be two Elliptic Curves defined on the same field. A morphism [178] or a rational application of* E_1 *toward* E_2 *is an application* t*:*

$$\begin{cases} t : \mathrm{E}_1 & \longrightarrow \mathrm{E}_2 \\ \mathcal{P} & \longrightarrow t(\mathcal{P}) = (r(\mathcal{P}); s(\mathcal{P})) \end{cases} \tag{4.3}$$

where r *and* s *are rational functions*[4].

[4] A rational function is any function which can be written as the ratio of two polynomial functions.

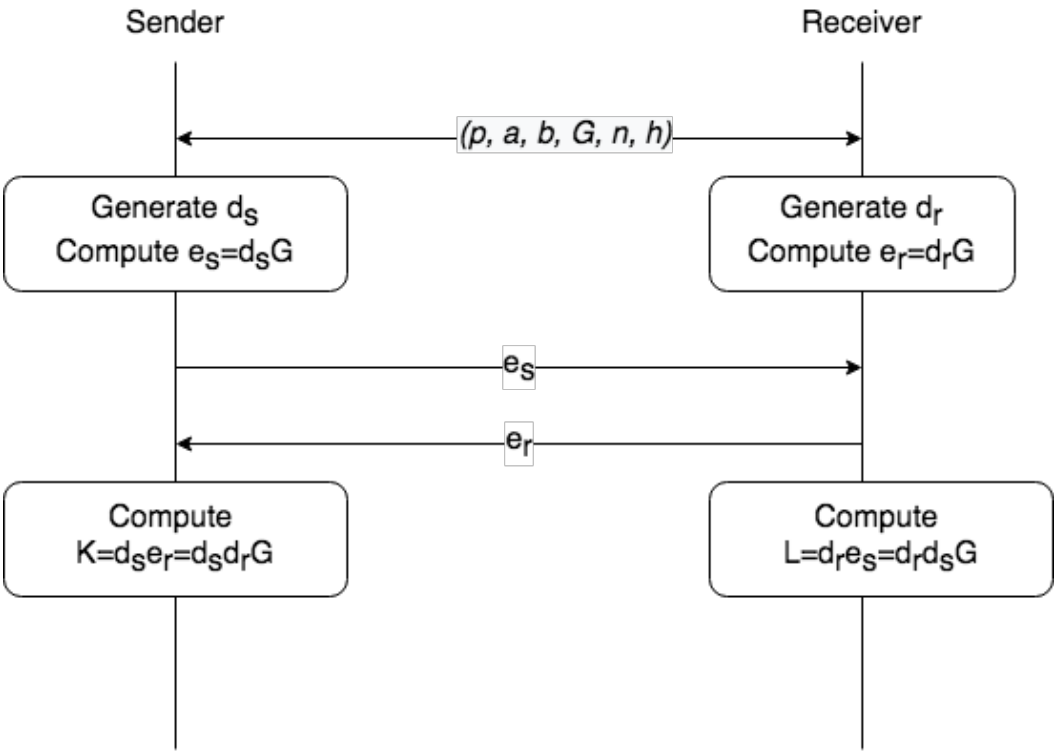


Figure 4.2 – ECDH Key Exchange Protocol

Definition 4. *An Isogeny [178] of* E_1 *toward* E_2 *is a non-constant morphism that maps the point at infinity* O *of* E_1 *to the point at infinity* O *of* E_2.
Property 1*: every Isogeny is a morphism of algebraic groups. which means that for every point* P *and* Q *of* E_1*, we have:*

$$\varphi(P+Q) = \varphi(P) + \varphi(Q) \text{ and } \varphi(-P) = -\varphi(P) \tag{4.4}$$

An Isogeny can always be reduced to the form described by Equation 4.5.

$$\varphi(x, y) = (r(x), ys(x)) \tag{4.5}$$

There are numerous types of Isogeny such as Isogeny of Translation and Isogeny of Frobenius and others. In this work, we consider the Isogeny of Translation, which can be defined by the Equation 4.6.

$$\begin{cases} [n] : \mathrm{E} & \longrightarrow \mathrm{E} \\ K & \longrightarrow nK = (r_n(x), ys_n(x)) \end{cases} \tag{4.6}$$

By considering $F(x)$ as the equation of the Elliptic Curve (Equation 4.1), we obtain

r_n and s_n as described by Equation 4.7.

$$\begin{cases} r_2(x) & = \frac{F'(x)^2}{4F(x)} - 2x \\ s_2(x) & = -\frac{F'(x)}{2F(x)}(r_2(x) - x) - 1 \\ & = -\frac{F'(x)}{2F(x)}(\frac{F'(x)^2}{4F(x)} - 3x) - 1 \\ r_{n+1}(x) & = F(x)[\frac{s_n(x)-1}{r_x(x)-x}]^2 - r_n(x) - x \\ s_{n+1}(x) & = -[\frac{s_n(x)-1}{r_x(x)-x}](r_{n+1}(x) - x) - 1 \end{cases} \tag{4.7}$$

4.2.2 One Time Password (OTP)

A One-Time Password (OTP), as its name indicates, is a password that is used only once and then "thrown away"; in that sense, it is a password valid for only session. OTP is mostly used for multi-factor authentication. Its main advantage is that it avoids the shortcomings commonly associated with traditional (static) password-based authentication. This section presents two OTP-generation algorithms: HMAC-based OTP (HOTP) and Time-based OTP (TOTP). Both algorithms require the use of tokens (hardware- or software-based) capable to generate the OTP, which can be verified by a validation service based on a shared secret.

4.2.2.1 HMAC-Based One-Time Password (HOTP)

HMAC-Based One-Time Password [4], is an algorithm to generate one-time password values, based on Hashed Message Authentication Code (HMAC). HOTP is one of the standards developed by the Initiative for Open AuTHentication (OATH), an international collaboration group which aims to promote strong authentication in open-source. The HOTP algorithm is based on:

- an increasing counter value
- a static symmetric key known only to the token and the validation service

As the output of the HMAC-SHA-1[179] calculation is 160 bits, this value must be truncated to something that can be easily entered by a user.

```
HOTP(K,C) = Truncate(HMAC-SHA-1(K,C))
```

Where:

- `Truncate` represents the function that converts an HMAC-SHA-1 value into an HOTP value.
- The Key (`K`), the Counter (`C`), and Data values are hashed high-order byte first.

4.2.2.2 Time-Based One-Time Password (TOTP)

Time-Based One-Time Password (TOTP) [5] appeared in 2011. It is also part of the standards developed by OATH. It is based on HOTP, with the only difference that the factor of change is time and not a counter. It is based on the "POSIX" time. Indeed, all processors have an internal clock implementing directly or indirectly "POSIX time". This avoids the synchronization problem because all the clocks (that of the OTP generator and the server) are synchronized with perhaps a tiny offset.
The initial sharing of the "secret" between the two entities remains the same. However, the generation of the OTP will be done with the couple "secret" and time (or more precisely a "timestamp") over a defined period (generally 30 to 60 seconds). This means that TOTP uses time incrementally and that each OTP is valid for the duration of the time interval.
Basically, we define TOTP as `TOTP = HOTP(K, T)`, where `T` is an integer and represents the number of time steps between the initial counter time T_0 and the current Unix time. More specifically, `T=(Current Unix time -` T_0`)/X`, where the default floor function is used in the computation. For example, with $T_0 = 0$ and Time Step $X = 30$, $T = 1$ if the current Unix time is 59 seconds, and `T = 2` if the current Unix time is 60 seconds.

4.3 Scheme Overview

This section presents the ECC-based authentication scheme, which consists of two main phases: (1) the initialization phase or pre-authentication phase in which the parameters of the authentication protocol are agreed upon between the object (denoted as Obj) and the management server (detoted as S), and (2) the operation phase in which the protocol is "in action". The algorithm of the proposed authentication scheme is described in Algorithm 1 and illustrated in Figure 4.3.

4.3.1 Initialization Phase

When a new object Obj is added to the system, Obj and S agree on a Pre-Shared Key (PSK), and on $\mathcal{P}$; a generator point of the Elliptic Curve E, as defined in Equation 4.2. We assume that objects are protected against physical attacks, where an attacker can retrieve some/all of the object's secrets such as private or shared keys. Numerous methods exist to protect them from such attacks by making these information readable only by the device itself [180][181]. Although these techniques are more expensive compared with traditional methods but they guarantee the protection of keys. Consequently, our approach is resilient against PSK theft that can be achieved through: (1) network communication cryptanalysis (since a new key is used for each communication) and (2) physical attacks by using dedicated methods as described above.

Algorithm 1: System functioning (Object side)

parameter: $\mathcal{P}$: Coordinates // $\mathcal{P}(x_{\mathcal{P}}, y_{\mathcal{P}})$
PSK: String // the Preshard key
ServerAddress: Address

begin
// Main program
Coordinates $K \leftarrow$ ECDH (ServerAddress) ;
Real $r_2 \leftarrow$ Isogeny (K, 2).getR();
Real $s_2 \leftarrow$ Isogeny (K, 2).getS();
Real $Z_2 \leftarrow$ XOR (r_2, s_2);
Real $OTP_1 \leftarrow$ FirstBytes ((HmacSHA (Z_2), PSK), NbBytes);
// PSK is used for HmacSHA() function
// NbBytes represents the number of needed bytes
Send (OTP_1, ServerAddress);
if ReceiveServerResponse $()$ == *AuthSuccess* **then**
for $n \leftarrow 3$ **to** NbOfNeededKeys **do**
Real $r_n \leftarrow$ Isogeny (K, n).getR();
Real $s_n \leftarrow$ Isogeny (K, n).getS();
Real $Z_n \leftarrow$ XOR (r_n, s_n);
Real $OTK \leftarrow$ FirstBytes ((HmacSHA (Z_n), PSK), NbBytes);
CryptographicOperation (OTK);
;
Function : ECDH (Address ServerAddress)
begin
Int $n_1 \leftarrow$ GenerateInt () ;
Real $Q_1 \leftarrow n_1.\mathcal{P}$;
Send (Q_1, ServerAddress);
Real $Q_2 \leftarrow$ ReceiveServerResponse ();
return Coordinates K
Function : Isogeny (Coordinates K, Integer n)
begin
Real $r_n \leftarrow$ ComputeR(n); // Equation 4.7
Real $s_n \leftarrow$ ComputeS(n); // Equation 4.7
return r_n, s_n

4.3.2 Authentication Protocol

For each communication session between an object *Obj* and the server S, first, a new OTP is generated for *Obj* authentication. Then, a new key (OTK) is needed for each message (for authentication or confidentiality). To do so, for each communication session, an Elliptic Curve Diffie-Hellman (ECDH) exchange is established as follows:

1. *Obj* and S generate two natural numbers denoted n_1 and n_2 respectively

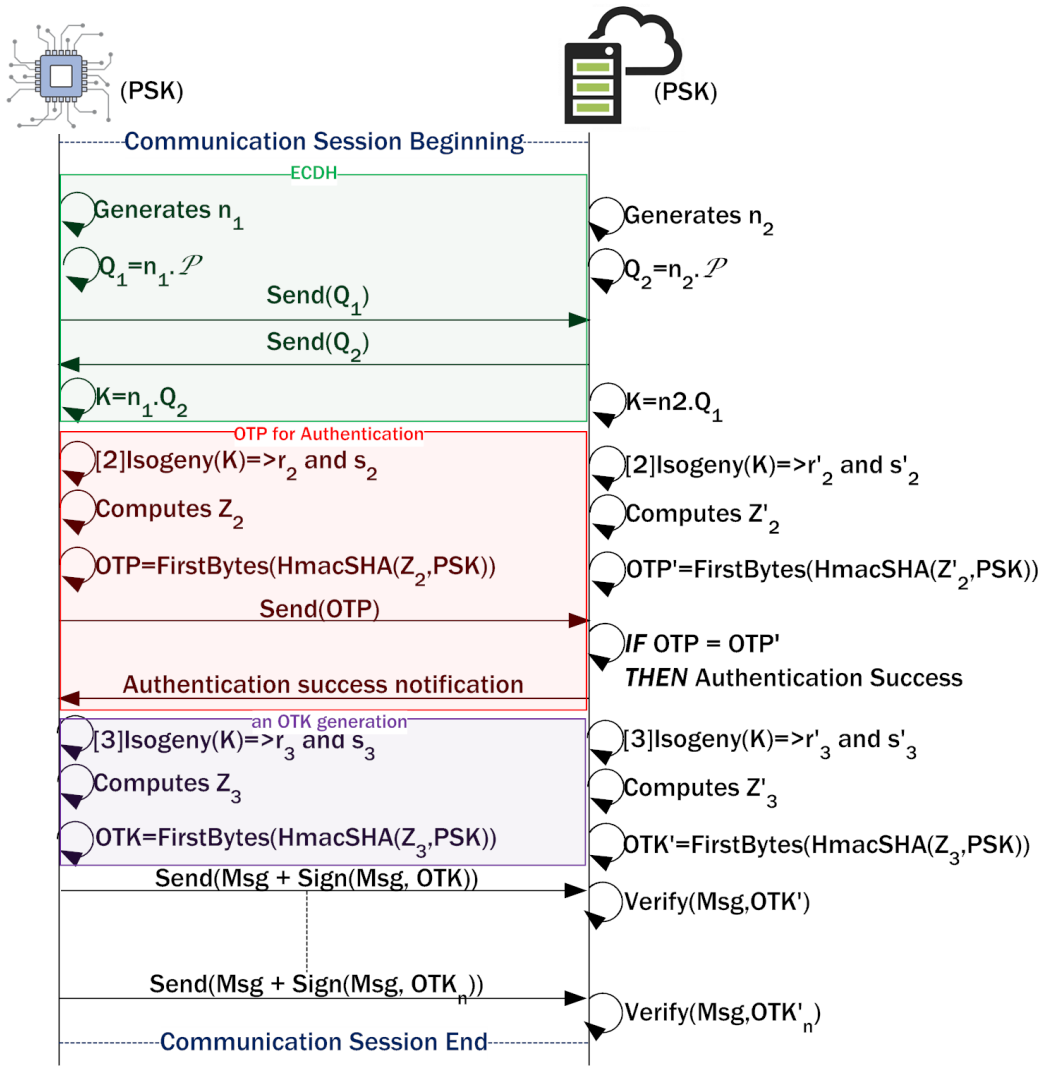


Figure 4.3 – Authentication Protocol

2. For $i \in \{1, 2\}$, Obj and S compute and exchange the value $Q_i = n_i\mathcal{P}$.

3. Obj and S agree on the secret $K = n_1 n_2 \mathcal{P}$. We recall that K is a point of the curve E, which coordinates are denoted as x_k and y_k; $K(x_k, y_k)$

For each communication session only one ECDH is needed.

The next step is to derive the OTP from K. This is done by transforming K, while ensuring that K will not have the point at infinity of E as image after this transformation. This is done using Isogeny and in particular, the Isogeny of Translation for its simplicity. Other Isogeny applications will be considered in future works.

The isogeny of translation will provide the values of r_n and s_n as indicated in

equation 4.7. The next step is to compute an intermediate value that we denote Z_2 as described by Equation 4.8, for $n = 2$

$$Z_n = r_n \oplus s_n \tag{4.8}$$

Then, an HMAC-SHA[5] algorithm is applied to Z using the PSK as described by Equation 4.9, for $n = 2$

$$OTP = FirstBytes(HMAC - SHA(Z_n), PSK) \tag{4.9}$$

The function HMAC-SHA() can be replaced by any known hash function. Finally, the first bytes of the hash function's result are retrieved according to the OTP or key size needs. For instance, if 8 digits are needed for the OTP then the first 4 bytes are used.

The first OTP (OTP) is dedicated to the authentication. Indeed, *Obj* sends OTP to S. In turn, S repeats the same steps and computes the OTP' using K for $n = 2$. It compares the computed OTP' and the received one. If they match, then, it sends an authentication success result to *Obj*.

Once *Obj* is authenticated, it generates a new key for each communication (for signature or encryption) by applying the Isogeny on incremented values of n i.e. $n = 3, 4, ...$ as described by Equation 4.7. After getting the values of r_n and s_n, Z_n is computed (as for OTP) according to Equation 4.8 and then a new OTP named OTK is computed according to Equation 4.9. According to the needs of the cryptographic operation's key size (signature or encryption according to the used algorithm), the first bytes of the generated OTK are used to compose the new key.

4.4 Performance Evaluation

The performance evaluation of the proposed scheme is done as follow:

1. Evaluation of the performance for different scheme parameters namely, the Elliptic Curve type and the HMAC algorithm. This is important to tune the parameters of the proposed protocol.

2. Evaluation of the performance on different types of devices using the parameters showing the best performance results.

3. Comparison of the performance with other existing approaches.

4.4.1 Evaluation Framework

During the experiments, we used different types of devices: a Raspberry Pi and a Laptop as potential objects (Obj) and another machine as a server (S). Table 4.2 presents the specifications of the different devices.

[5] e.g HMAC-SHA256, HMAC-SHA512, ...

Node type	CPU Architecture	CPU Operation Mode	CPU Max. Speed	RAM	Operating System
Raspberry Pi	armv6l	32-bits	700 MHz	450 MB	Raspbian 4.9.41
HP laptop	x86_4	64-bits	2600 MHz	8 GB	Ubuntu 16.04
Dell Precision (S)	x86_4	32-bits	3100 MHz	8 GB	Ubuntu 14.04

Table 4.2 – Features of nodes used in our experiments

We implemented our OTP-based protocol in Java using *BouncyCastle* [6] library for Elliptic Curve functions. We are aware about the fact that the results obtained depend partially on the used language (Java) and will be different when using other languages (C for example). However, the goal of our study is to compare the performance of our protocol for different parameters with other existing works implemented in the same way to achieve fair comparison.
We evaluate only the OTP generation time, and not the signature or encryption time. More precisely, each evaluation covers the following scenarios:

1. *Obj* generates OTP and sends it to S.
2. S verifies the received OTP. If the verification is successful, it sends an authentication success result to *Obj*
3. *Obj* generates a new OTK relying on the same $K(x_k, y_k)$ used to generate the first OTP.
4. *Obj* sends the new OTK to S.
5. S checks if the received OTK matches with the calculated OTK'.
6. The last three steps (3, 4 and 5) are repeated continuously until the required number of OTKs (from 10^3 to 10^6, according to the experiment) is reached.

4.4.2 Tuning the protocol parameters

To find which parameters are best suited for our protocol, we conducted multiple experimental tests where we varied the HMAC algorithm and the Elliptic Curve used. We used the following Elliptic Curves: *Secp192r1, Secp256r1, secp521r1* [182] [183], *BrainpoolP192r1, brainpoolP256r1* and *brainpoolP512r1* [184]. Our experiments used *Secp* and *Brainpool* curves because they are the most popular in security standards [7]. For each Elliptic Curve used, we evaluate the connection time for the

[6] https://www.bouncycastle.org
[7] *ISO* standards use both of them, and *European Telecommunications Standards Institute (ETSI)* use *Secp* curves.

following three HMAC algorithms: HMAC-SHA1[8], HMAC-SHA256 and HMAC-SHA512.

We present the different results obtained from the evaluation tests that cover the average generation time of an OTP. For each chosen Elliptic Curve and hash algorithm, we performed five experimental tests where we vary the number of connections from 10^2 to 10^6. Each result is the average of at least 15 experimental tests. In other words, if we consider the case of 10^2 connections, in order to obtain the point present on the plot (Figure 4.4 and Figure 4.5):

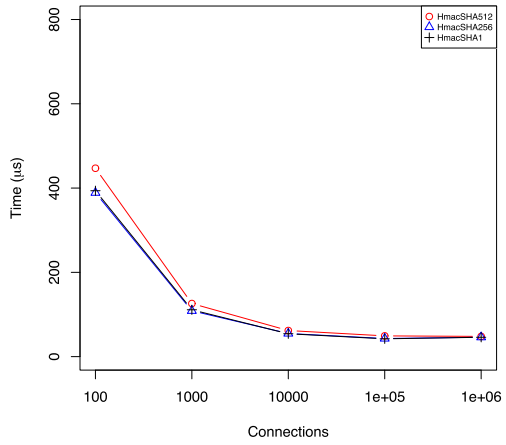
1. Conduct an experiment where 10^2 connections are made.
2. Compute the average generation time.
3. Repeat the same test 14 times where the average connection time is computed for each test.
4. Compute an average value over the 15 results obtained. The latter represents the final result for this case. The same procedure is repeated for the other experimental tests (from 10^3 to 10^6).

Figure 4.4 represents the results obtained for the *Secp* evaluated curves[9]. The results describe the OTP average generation time without considering the time needed for the ECDH exchange. For all the experimental tests, the average connection time decreases when the number of connections increases and begins to stabilize from 10^5 connections. This is due to caching and computation optimization techniques used by Java (as most of the programming languages). We make the following observations based on the results obtained:

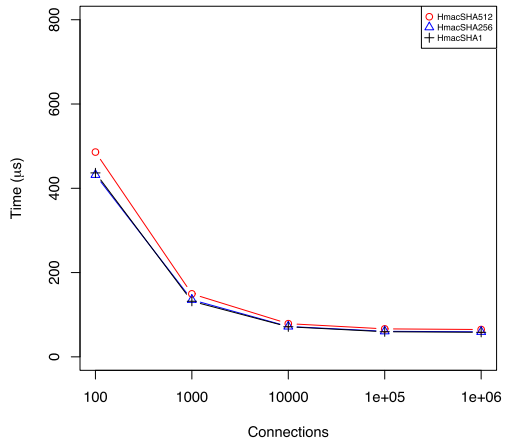
- For all the tests conducted, HMAC-SHA512 always gives the worst performance.
- For the test with *Secp521r1* curve (Figure 4.4.c), HMAC-SHA256 yields the best performance results.
- The results obtained with HMAC-SHA1 and HMAC-SHA256 are very close with the difference being about is less than 10 microseconds (μs).
- For *Secp192r1* (Figure 4.4.a), HMAC-SHA256 yields better performance results for the experimental tests conducted on $[10^2..10^3]$ connections, but HMAC-SHA1 yields better results for the rest.
- For *Secp256r1* (Figure 4.4.b), HMAC-SHA256 yields better performances for the tests conducted on 10^2 connections, but HMAC-SHA1 realizes better results for the rest.

[8]It is not advised to use SHA1 because of its vulnerability to collisions [185]. We have used SHA1 for performance comparison only.

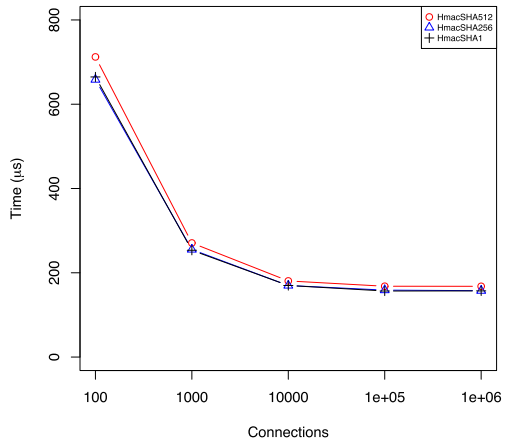
[9]Herein, the experiments were realized using the Laptop as *Obj*.



(a) Secp192r1

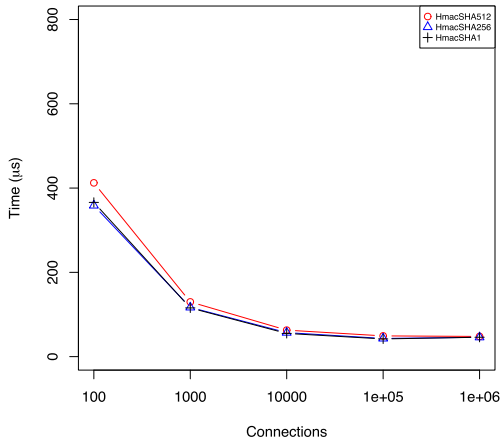


(b) Secp256r1

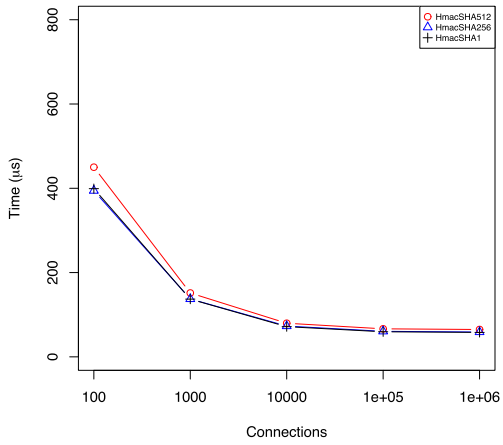


(c) SecP521r1

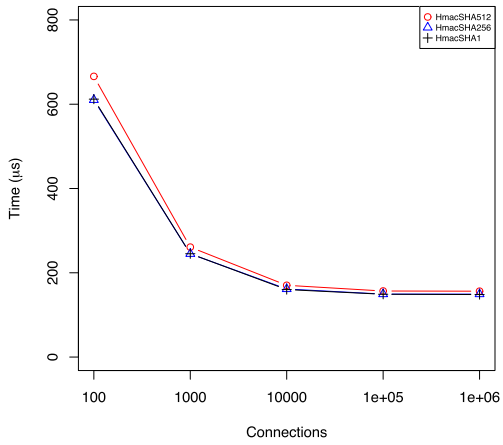
Figure 4.4 – Average OTP generation time - Secp Curves



(a) Brainpoolp192r1



(b) Brainpoolp256r1



(c) Brainpoolp512r1

Figure 4.5 – Average OTP generation time - Brainpool Curves

To compare the different curves, we considered the tests involving 10^6 connections and we used HMAC-SHA1 as the hash algorithm. We note that *Secp192r1* gives the best performances with an average generation time of 45.95 μs. *Secp256r1* yields 58.26 μs whereas *Secp521r1* yields the worst result with 157 μs.

Figure 4.5 shows the results obtained with *Brainpool* curves. As for *Secp* curves, HMAC-SHA512 always yields the worst performance for all tests. Furthermore, HMAC-SHA1 and HMAC-SHA256 yield almost the same generation times.

As for the last case, to compare results obtained with the different curves we considered the case of 10^6 connections using the HMAC-SHA1 algorithm. We note that *Brainpoolp192r1* gives the best performances with 46 μs, *Brainpoolp256r1* yields 58 μs and *Brainpoolp512r1* gives the worst case with 148.33μs.

The time taken for the ECDH exchange also depends on the curve used. Table 4.3 presents the average times of ECDH exchanges obtained where we note the following ranking:

1. *Secp192r1* with an average time of 576020 μs;
2. *Secp256r1* with 0.577100 sec
3. *Brainpool192r1* with 0.580653 sec
4. *Brainpool256* with 0.586569 sec
5. *Secp521r1* having 0.606800 sec, and
6. *Brainpool512* with 0.630071 sec.

Curve	**Secp**			**Brainpool**		
	192r1	**256r1**	**521r1**	**p192**	**p256**	**p512**
Time (sec)	0.576020	0.577100	0.606800	0.580653	0.586569	0.630071

Table 4.3 – Average time of ECDH exchange

To compare the overall performance of our proposed protocol regarding both curves *Secp* and *Brainpool*, we note that both EC families achieve almost similar results. The time needed for the ECDH exchange is more important than the connection time, which makes the curve choice the first factor to consider. Thus, the *Secp192r1* curve is best suited for our proposed protocol.
For the rest of this chapter, all the results presented have been obtained through experiments using Secp192r1 and HMAC-SHA256 as parameters because of their better performances in comparison to other parameters.

Node type	OTP generation time (μs)		OTP CPU power (mW)		OTP NIC power (mW)	
	Avg.	**SD**	**Avg.**	**SD**	**Avg.**	**SD**
Raspberry Pi	1130	20.53	52.57	4.06	9.17	0.35
Laptop	108.75	4.98	10.86	2.41	3.70	0.09

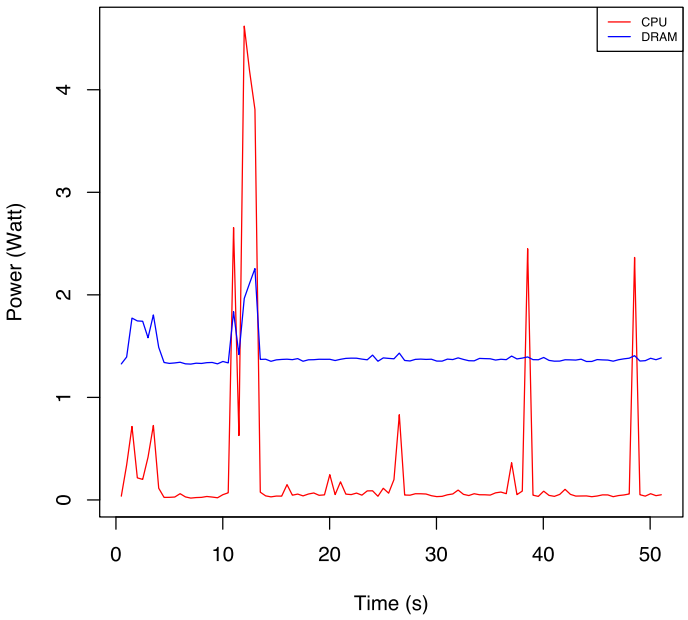
Table 4.4 – Results Averages and Standard Deviations

4.4.3 Performance evaluation

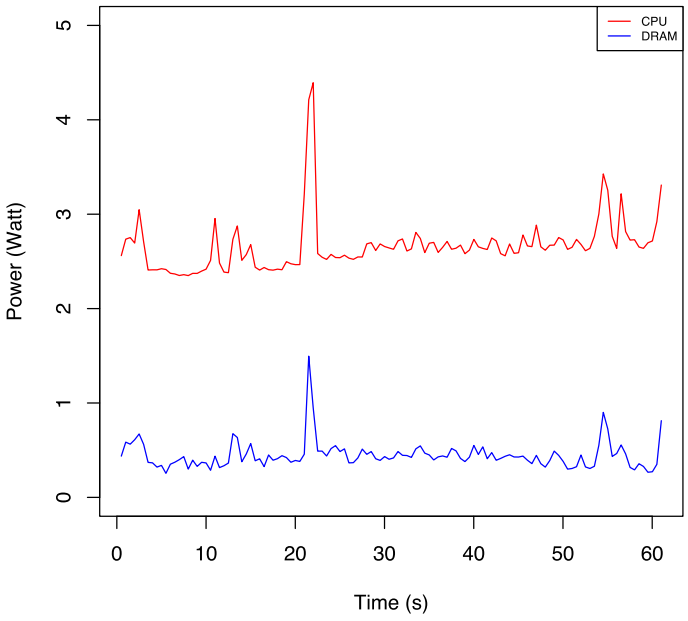
Table 4.4 shows the average and standard deviation (SD) of the obtained results over 10^3 realized experimentations using *Secp192r1* and HMAC-SHA256 as parameters.

The average time needed to compute an OTP is 108.75 μs for the Laptop. The Raspberry Pi needs more time with 1130 μs. However, in both cases the standard deviations are low which demonstrates the stability of the computations.
For energy consumption, the CPU of the Raspberry Pi consumes 52.57 mW to compute an OTP whereas the Laptop consumes only 10.86 mW. Besides, the Network Interface Controller (NIC) of the Raspberry Pi requires 9.17 mW, while the Laptop needs 3.70 mW.
Our proposal needs 2 messages for each communication part (total of 4 messages) to perform a full authentication. Then, it does not need any message to generate OTKs, which reduces the processing time and energy consumption. This feature, along with its ease of deployment, make our approach well suited for various use case scenarios in the IoT context.
Figure 4.6 shows the impact of OTP computation and the transmission of messages on the CPU and the Dynamic Random Access Memory (DRAM) energy consumption.
The figures show three phases of the system's operation: (1) an idle phase, (2) the execution of a loop that generates and sends 50 messages with a break of 500 milliseconds between each message, and (3) the return to the idle phase. The measurements were obtained using RAPL[10] measurement tool[11]. Figure 4.6.a shows the Laptop's results where the loop is executed in the interval $[20, 46]$ seconds. Figure 4.6.b describes the Server's results where the application server was executed at the 15^{th} second. Then, the requests began at the 20^{th} second until the 46^{th} second. In both cases, we note that the impact of the loop is almost negligible. The other existing peaks are related to the operating system activity.

[10] https://github.com/kentcz/rapl-tools
[11] RAPL measures the total CPU and DRAM activity and cannot isolate the measurements by a selected process



(a) Object



(b) Server

Figure 4.6 – Impact of OTP generation and message transmission processing

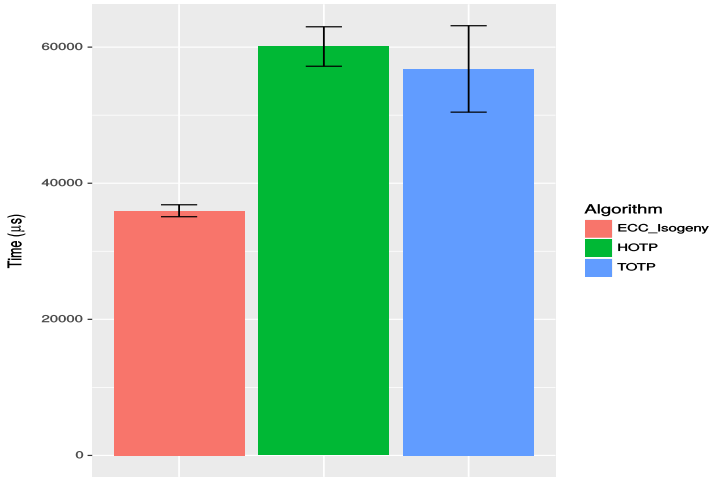
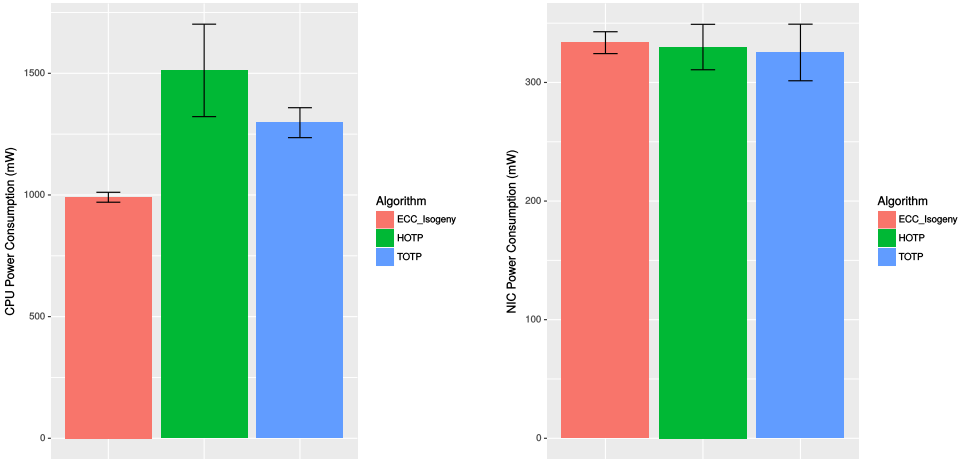


Figure 4.7 – Generation time of 100 OTPs - Average and SD



(a) CPU Power Consumption (b) NIC Power Consumption

Figure 4.8 – Generation of 100 OTPs - Power Consumption

4.4.4 Comparison with related works

In this section we compare the performance of our approach with the following algorithms:

- HMAC-based One Time Password (HOTP)[4]
- Time-based One Time password (TOTP) [5]

We choose to compare our approach with these two algorithms because they are the most well-known OTP standards. The comparison covers :

- Time required to generate OTPs
- CPU power needed to generate OTPs
- NIC power needed to generate OTPs

For our experiments, we used a Java implementation of each algorithm that is available in each RFC[12].

If we consider a comparison based on the results obtained for only one OTP generation, the differences will be very hard to evaluate because of the small values and their similarity. Thus, we compare the results obtained for the generation of 100 OTPs.

Figure 4.7 shows the average time needed to generate 100 OTPs. HOTP needs 60089.85 μs with a standard deviation of 2899 μs while TOTP needs 56795.22 μs (SD = 6349.28 μs).
In our approach, Figure 4.7 shows that the time needed for the OTP generation is only 35956.70 μs (SD = 877.70 μs) which represents almost half of the time taken by the other algorithms. These results demonstrate that Isogeny is not costly in computation time. However, an ECDH key sharing phase is needed before these OTPs generation which needs time (576020 μs) compared with the OTP generation time, which results in our approach having the worst time considering the whole process of the generation of 100 OTPs. However, this time waste (due to the network communication) is also present for the challenge-response in asynchronous OTP-type approaches.
Figure 4.8.a shows the average CPU power consumption needed to generate 100 OTPs. The worst result is obtained by HOTP which needs 1511.94 mW (190 mW SD). The second position is for TOTP which consumes 1297.08 mW (61.52 mW SD). Our approach consumes only 990.61 mW (10.82 mW SD) thereby achieving the best result. This value also considers the ECDH exchange.
Figure 4.8.b shows the NIC power consumption needed to generate and send 100 OTPs. The set of obtained results are very close to each other. Our approach consumes more power because of the ECDH exchange. It consumes 333.54 mW

[12]Java implementation of HOTP is described in [4] on page 27. TOTP Java implementation is described [5] on page 8.

(9.22 mW SD), while HOTP needs 329.87 mW (19.16 mW SD) and TOTP achieves the best result by consuming 325.28 mW (23.86 mW SD).

To summarize the comparison with HOTP and TOTP, our approach:

- Consumes less power and energy for computation.
- Has a similar NIC consumption as the standards we studied.
- Has the worst result in terms of execution time.

4.5 Conclusion

In this chapter, we have proposed an original approach for OTP generation that relies on *Elliptic Curve Cryptography* and Isogeny. We have extended the concept of OTP and used it to generate a new key to be used for each exchange between the IoT device and the server. Furthermore, our approach has the advantage of not relying on a counter or a timestamp as in synchronous OTP-type approaches. In addition, it does not need challenge/response management as in asynchronous OTP-type approaches.
To evaluate our protocol, we deployed a real implementation using Java. The extensive evaluation that we conducted has demonstrated its efficiency.
The next chapter presents the second and third contribution of the thesis: Blockchain-based Authentication protocols.

Chapter 5

Blockchain-based Lightweight IoT Authentication Schemes

5.1 Introduction

The IoT security issues are among the main obstacles for its adoption. In particular authentication and authorization methods hold a golden place in priority rank. The existing approaches suffer from numerous limitations mainly related to their inappropriateness (partially or fully) with the special nature of IoT systems which requires flexible, scalable, lightweight and robust solutions.
Such requirements cannot be satisfied by centralized architectures which although have proved their robustness (in traditional networked systems) but cannot meet the high scalability needs of IoT systems especially when thousands or tens of thousands of IoT devices are connected in the same network.
The second and third contributions of this thesis, presented in this chapter, attempt to remedy this architectural issue by relying on Blockchains which are considered as a very promising technology for the development of decentralized and resilient security solutions in IoT context.
This chapter consists of three main parts:

- The first part provides an introduction to Blockchain which builds the minimum background information needed to understand the two contributions.
- The second part proposes a simple and lightweight blockchain-based authentication solution for IoT systems with two possible authentication scenarios and an evaluation based on real implementation.
- The last part proposes an adaptive approach that ensures the authentication and the authorization for IoT devices, that addresses in particular some main deployment issues not sufficiently tackled by current approaches. It also presents the results of the performance and security evaluation of the proposed scheme.

5.2 Blockchain - an overview

Blockchain, as a decentralized and distributed public ledger technology in peer-to-peer networks, has received considerable attention recently. Blockchain was introduced in October 2008 [186] as part of a proposal for bitcoin, a virtual currency system that eschewed a central authority for issuing currency, transferring ownership, and confirming transactions. Additionally, blockchain technology is becoming one of the most promising technologies for the next generation of internet interaction systems, such as smart contracts, public services , internet of things (IoT) , reputation systems and security services [187].
The ledger is composed of a set of blocks. Each block contains two parts:

- The first part represents the body of the block. It contains the transactions, also called facts, which the database must record. These facts can be monetary transactions, medical data, industrial information, logs systems, etc.
- The second part is the header of the block. It contains information about the block such as timestamp, transaction hash, etc, as well as the hash of the previous block.

As a result, all the existing blocks form a chain of linked and ordered blocks. The longer the chain, the harder it is to falsify it. Indeed, if a malicious user wants to modify or exchange a transaction on a block, (1) it must modify all the following blocks, since they are bound by their hashes. (2) Next, it must change the version of the blockchain that each participating object stores. Figure 5.1 illustrates a simplified example of a blockchain.

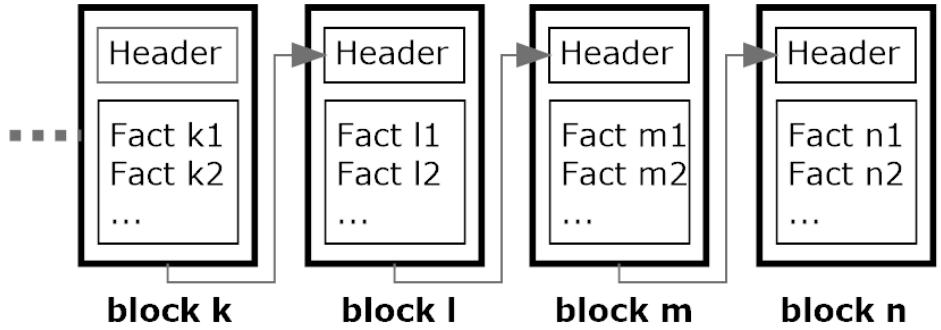


Figure 5.1 – Simplified example of a blockchain

5.2.1 Architecture

There are several types of nodes in a Blockchain [188]: Full Nodes, Lightweight Nodes and Mining Nodes

1. **Full nodes**: full nodes are responsible for holding and distributing copies of the entire blockchain ledger. In that sense, full nodes download every block

and transaction and check them against consensus rules (section 5.2.4); a full node is able to validate transactions all the way back to the genesis block [1].

2. **Lightweight nodes**: Lightweight nodes (some times referred to as light nodes) do not download the complete Blockchain; they just download the block headers, only to validate the authenticity of the transactions.

3. **Mining nodes**: Also known as "Miners", they can be simply defined as the nodes that produce the blocks for the blockchain. They confirm the blocks to add to the blockchain in a process called “mining”.

5.2.2 Key characteristics

Blockchain has the following key characteristics [187] :

- *Decentralisation*: In conventional centralised transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank)inevitably resulting in the cost and the performance bottlenecks at the central servers. Differently, a transaction in the blockchain network can be conducted between any two peers (Peer-to-Peer) without the authentication by the central agency. In this manner, blockchain can significantly reduce the server costs (including the development cost and the operation cost) and mitigate the performance bottlenecks at the central server

- *Persistence*: Since each of the transactions spreading across the network needs to be confirmed and recorded in blocks distributed in the whole network, it is nearly impossible to tamper. Additionally, each broadcasted block would be validated by other nodes and transactions would be checked. So any falsification could be detected easily.

- *Anonymity*: Each user can interact with the blockchain network with a generated address. Further, a user could generate many addresses to avoid identity exposure. There is no longer any central party keeping users' private information. This mechanism preserves a certain amount of privacy on the transactions included in the blockchain. Note that blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint.

- *Auditability*: Since each of the transactions on the blockchain is validated and recorded with a timestamp, users can easily verify and trace the previous records through accessing any node in the distributed network. In Bitcoin blockchain, each transaction could be traced to previous transactions iteratively. It improves the traceability and the transparency of the data stored in the blockchain.

[1] A genesis block is the first block of a Blockchain.

5.2.3 How Blockchain works?

To add a new block to the blockchain, one have to follow the following steps:

1. A transaction is grouped with other transactions in a block
2. The miners verify that the block transactions comply with the defined rules
3. The miners execute a consensus mechanism to validate the added block
4. A reward is given to miners who validate the block
5. The verified transactions are stored in the blockchain.

The steps to run the network are as follow[189]:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works, according to the consensus algorithm of the blockchain, to validate and add a new block into the blockchain.
4. Nodes accept the block only if all transactions in it are valid and not already spent.
5. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash

5.2.4 Consensus Algorithms

Blockchain platforms use a range of consensus protocols to prove the honest validation of a block. There are three main properties that determine the applicability and efficiency of a consensus protocol[190]:

1. **Safety**: A consensus protocol is considered safe if all nodes produce the same valid output; the validity of the outputs is determined according to the rules set for the model. This is also referred to as consistency of the shared state.
2. **Liveness**: The liveness of a consensus protocol is ensured if all non-faulty nodes participating in consensus eventually produce a value.
3. **Fault Tolerance**: Fault tolerance of a consensus protocol is provided if it can recover from failure of any node participating in the consensus.

There are many mechanisms for validation; the most used are the Proof-of-Work (PoW) and Proof of Stake (PoS).
Proof of Work is a consensus protocol used in the Bitcoin network [186]. In POW, each node of the network is calculating a hash value of the block header (which is constantly changing upon addition of new transactions). The consensus requires that the calculated value must be equal to or smaller than a certain given

value. All miners participating in the consensus have to calculate the hash value continuously by using different nonces until the target is reached. When one node obtains the relevant value, all other nodes must mutually confirm the correctness of the value. This allows to validate transactions in the new block in case of frauds. Then, the collection of transactions used for the calculations is approved to be the authenticated result, which is denoted by a new block in the blockchain. Given the time and thus the power consumption needed for the calculation process, an incentive mechanism (e.g., granting a small portion of Bitcoin to the miner) is also proposed [186] [187] [190]

Proof of Stake is a method by which a blockchain of a cryptocurrency aims to reach a distributed consensus. Proof of Stake is a consensus method by which a network node is randomly selected to add a new block to the blockchain. In order to be able to claim to add a new block, the node must provide proof of possession of the cryptocurrency associated with the blockchain. The random selection of the node is then weighted by the amount of cryptocurrency owned by each node of the network. The main advantage of the Proof-of-stake is that it avoids the phenomenal energy expenditure of Proof-of-Work. On the other hand, many believe that the blockchain thus forged does not have the same robustness and the same level of immutability of Bitcoin Proof-of-Work. POS is an energy-saving alternative to POW, some research efforts focus on hybrid PoW/PoS mechanisms.

Other consensus algorithms have different advantages and disadvantages : Delegated proof of stake (DPOS) was developed by Daniel Larimer, in 2014, [191], similarly to POS, miners get their priority to generate the blocks according to their stake.
There are other kinds of consensus mechanism algorithms using the variants of Byzantine Fault Tolerance (BFT): Practical Byzantine Fault Tolerance (PBFT) [192], is the first practical solution to the achieving consensus in the face of Byzantine failures, PBFT is a consensus algorithms used in HyperLedger Fabric. Cross-Fault Tolerance (XFT) [193] is a protocol that simplifies the attack model and make Byzantine Fault tolerance feasible and efficient for practical scenarios. Ripple and Stellar are two blockchain based platforms and payment protocols that use variations of the Byzantine Fault Tolerance consensus models by making them open-ended with respect to node participation. Ripple and Stellar use respectively Ripple Consensus Protocol Algorithm [194] and Stellar Consensus Protocol [195].

5.2.5 Blockchain Platforms

The list of blockchain platforms in in continuous growth; the most important are: Bitcoin, Ethereum and Hyperledger Fabric.
Bitcoin [189] [186] is a money transfer and verification system. It uses a totally decentralized Peer-to-Peer (P2P) network and does not depend, in fact, on any central authority.The network timestamps transactions by hashing them into

an ongoing chain of hash-based proof-of-work. The Bitcoin protocol originally aims to provide an experimental alternative electronic payment system based on cryptographic proof instead of trust. Its unit of account has eight decimal places and it is written bitcoin. It works with software that allows users to create payment email addresses that can send or receive bitcoins. Its code is completely open source.

Ethereum is considered as the most promising blockchain besides Bitcoin. It is considered by its creators as the "first true global computer"[196]. It allows to provide a cryptocurrency called Ether (ETH). This blockchain is used to perform financial transactions as well as processing other types of data. In addition to block validation, Ethereum miners deal with programs called Smart Contracts. A smart contract is a computer program consists of a set of rules run on the blockchain [197]. This is why Ethereum represents a platform for decentralized applications. Smart contracts are executed by participating objects using an operating system called Ethereum Virtual Machine (EVM).

Hyperledger Fabric [198] is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability. It is designed to support pluggable implementations of different components and accommodate the complexity and intricacies that exist across the economic ecosystem.
Hyperledger Fabric delivers a uniquely elastic and extensible architecture, distinguishing it from alternative blockchain solutions. Planning for the future of enterprise blockchain requires building on top of a fully vetted, open-source architecture. The validating peers run a byzantine fault tolerance (BFT) consensus protocol for executing a replicated state machine.

5.3 A Blockchain-based Authentication Scheme for IoT systems

The authentication between nodes is essential to the functioning of the majority of IoT use cases. Currently, using a PKI represents the most reliable authentication method. However, this approach is centralized and cannot handle the future evolution of IoT scenarios especially considering the huge number of devices. In this context, we propose a decentralized and lightweight blockchain-based authenticated approach, that does not require a special hardware which makes it suitable for numerous IoT use case scenarios. This section describes the different details related to the design and the functioning of the proposed approach.

5.3.1 Authentication Scenarios

Our approach relies mainly on the usage of a blockchain. Thus, when a new device is added to the network, the user that adds it must register it (initialization phase). This is provided by sending a transaction to the blockchain, which contains: (1) the object's identifier (Object ID) and (2) the object's public key. When the transaction is treated and stored in the blockchain, the device obtains the transaction identifier (Transaction ID).

Let O_1 and O_2 be two objects. In order to establish a communication session between these two objects, the authentication is a mandatory requirement and there are two defined scenarios: (1) simple authentication or (2) mutual authentication. We assume that each node provides basic protocol primitives to send and receive messages as well as other functions such as signature provision, signature verification and so on. Algorithm 2 depicts such an Application Programming Interface (API).

Algorithm 2: Basic operations of a node

Function `Send` (Message Msg, Node $Receiver$) : Void `// Sends a message`
Function `Receive` (Message Msg, Node $Sender$) : Message `// Receives a message`
Function `Sign` (Message Msg, PrivateKey $privateKey$) : Message `// Signs a message`
Function `VerifySignature` (Message Msg, PublicKey $publicKey$) : Boolean `// Verifies the signature of a message`
Function `SessionReq` (String ID, String $Transaction\_ID$, String $CipherSuite$) : Message `// Creates a Session Request for a given destination node`
Function `SessionReqMutualAuth` (String ID, String $Transaction\_ID$, String $CipherSuite$) : Message `// Creates a Session Request with mutual authentication for a given destination node`
Function `SessionRep` (Message $AuthResponse$) : Message `// Creates a Session Response for a given destination node`
Function `GenrateRandom` ( ) : String `// Signs a message`
Function `GetTransaction` (String Transaction_ID) : Transaction `// Signs a message`
Function `ExtractParamFromTransaction` (Transaction transaction) : PublicKey `// Signs a message`
Function `Error` (String $errorMessage$) : Void `// returns and error message`
Function `ConnectionAbortion` ( ) : Void `// Connection session abortion`
Function `ConnectionEstablishment` (Node N_1, Node N_2) : CommSession `// Connection session establishment`

5.3.1.1 Simple Authentication Scenario

When O_1 wants to establish a communication session with O_2, The latter needs to authenticate O_1. More precisely, O_1 sends a Session Establishment Request ($SEReq$) that contains its *Object ID*, the cipher suite used and the *Transaction ID* of the blockchain's transaction that contains its authentication parameters (following initialization phase). When O_2 receives the request, relying on the *Transaction ID*,

it downloads, from the blockchain, the parameters related to that requester object. Then, O_2 generates a random challenge R and sends it to O_1. When the challenge is received by O_1, the latter provides a signature (using its private key), then, sends the challenge signed to O_2. Finally, O_2 verifies the signature using the public key it downloaded from the blockchain and if the signature is verified, than it means that O_1 is authenticated.
Afterwards, a Session Establishment Response (*SERep*) is sent to the device to inform it whether it is successfully authenticated or not. Finally, if there is a successful authentication, then, the session establishment can be set up.
Figure 5.2 exhibits the authentication steps such as detailed by Algorithm 3.

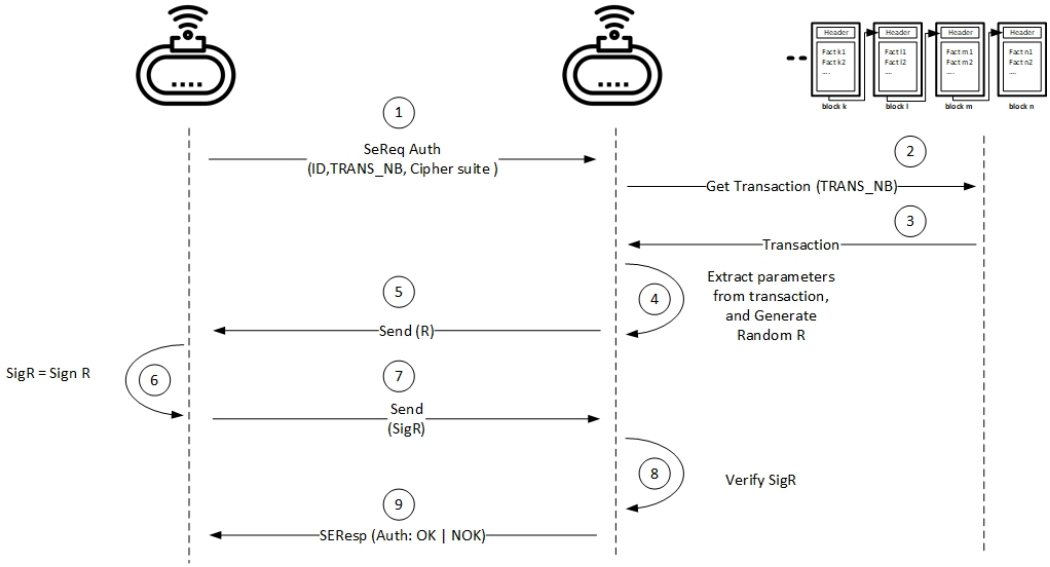


Figure 5.2 – Simple authentication use case scenario

5.3.1.2 Mutual Authentication Scenario

In some cases mutual authentication is needed. That is, O_2 needs to authenticate O_1, but also, O_1 needs to ensure that it communicates with the real O_2. This use case scenario is detailed by Figure 5.3 and Algorithm 4. More precisely:

1. O_1 sends a Session Establishment Request (*SEReq*) with mutual authentication. The *SEReq* contains O_1's *Object ID*, the cipher suite used as well as the Transaction Identifier (*Transaction ID_ 1*) of the blockchain's transaction that contains O_1's authentication parameters.

2. When O_2 receives the request, relying on $Transaction\ ID_1$, it downloads from the blockchain, the parameters related to O_1. Then, it generates a random challenge R_1 and sends it to O_1 along with a transaction identifier ($Transaction\ ID_2$) obtained from the initialization phase of O_2.

Algorithm 3: Simple authentication algorithm

```
O1, O2 : Node
Transaction_ID: String
// Transaction ID of the blockchain transaction containing device's
   parameters
CipherSuite: String
// The cipher suite used by the device
SeReq, SeRep: Message
Transaction: Transaction
R: String
begin
    SeReq = SessionReq (O1.ID, O1.Transaction_ID, O1.CipherSuite);
    O1.Send (SeReq, O2);
    SeReq' = O2.Receive (SeReq, O1);
    Transaction= O2.GetTransaction (SeReq.Transaction_ID);
    Params = O2.ExtractParamFromTransaction (Transaction);
    R = O2.GenrateRandom;
    O2.Send (R, O1);
    R' = O1.Receive (R, O2);
    Signature = O1.Sign (R', O1.PrivateKey);
    O1.Send (Signature, O2);
    Signature' = O2.Receive (Signature, O1);
    if VerifySignature (Signature', Params.PublicKey) then
        SeRep = SessionRep ("OK");
        O2.Send (SeRep, O1);
        SeRep' = O1.Receive (SeRep, O2);
        Session = ConnectionEstablishment (O1, O2);
    else
        Error ("Challenge signature not verified");
        ConnectionAbortion ( );
```

3. When the challenge is received by O_1, it signs R_1, and downloads from the blockchain, the parameters related to O_2 based on $Transaction\ ID_2$. Then, it generates a random challenge R_2, and sends the signature of R_1 as well as the challenge generated (R_2) to O_2 .

4. O_2 verifies the received signature of R_1 using the public key related to O_1 that was extracted from the blockchain. If the signature verification fails then, the authentication fails and no session is established. If the signature is verified, then, O_2 signs the received challenge R_2 and sends the authentication success notification as well as the challenge R_2 signed.

5. O_1 verifies the received signature of the challenge R_2. If the signature verification fails then, the authentication fails and no session is established. If the signature is verified, then, O_1 sends an authentication success notification to O_2, the communication session is established.

Algorithm 4: Mutual authentication algorithm

```
O1, O2 : Node
begin
    SeReq = SessionReqMutualAuth (O1.ID, O1.Transaction_ID,
     O1.CipherSuite);
    O1.Send (SeReq, O2);
    SeReq' = O2.Receive (SeReq, O1);
    Transaction1= O2.GetTransaction (SeReq'.Transaction_ID);
    Params1 = O2.ExtractParamFromTransaction (Transaction1);
    R1 = O2.GenrateRandom ( );
    O2.Send ((R1, O2.Transaction_ID, O2.CipherSuite), O1);
    Resp1 = O1.Receive ((R1, O2.Transaction_ID, O2.CipherSuite), O2);
    Transaction2= O1.GetTransaction (Resp1.Transaction_ID);
    Params2 = O1.ExtractParamFromTransaction (Transaction2);
    Signature1 = O1.Sign (Resp1.R1, O1.PrivateKey);
    R2 = O1.GenrateRandom ( );
    O1.Send ((Signature1, R2), O2);
    Resp2 = O2.Receive ((Signature1, R2), O1);
    if O2.VerifySignature (Resp2.Signature1, Params1.PublicKey) then
        Signature2 = O2.Sign (Resp2.R2, O2.PrivateKey);
        SeRep1 = SessionRep ("OK");
        O2.Send ((SeRep1, Signature2), O1);
        Resp3 = O1.Receive ((SeRep1, Signature2), O2);
        if O1.VerifySignature (Resp3.Signature2, Params2.PublicKey) then
            SeRep2 = SessionRep ("OK");
            O1.Send (SeRep2, O2);
            SeRep'2 = O2.Receive (SeRep2, O1);
            Session = ConnectionEstablishment (O1, O2);
        else
            Error ("Challenge signature by O2 not verified");
            ConnectionAbortion ( );
    else
        Error ("Challenge signature by O1 not verified");
        ConnectionAbortion ( );
```

5.3.2 Evaluation and Discussion

In this section, we describe the evaluation of our approach regarding its execution time. We evaluated the needed time for the whole authentication mechanism, that is to say, we measured the authentication process from the moment where a node sends an *SEReq* until the moment when it receives an *SERep*. The evaluation was performed through a real implementation.

5.3.2.1 Context and evaluation framework

We developed and deployed our approach on two devices: two *Raspberry Pi 3* that implements a *Python3* version of our approach. Table 5.1 describes the features of

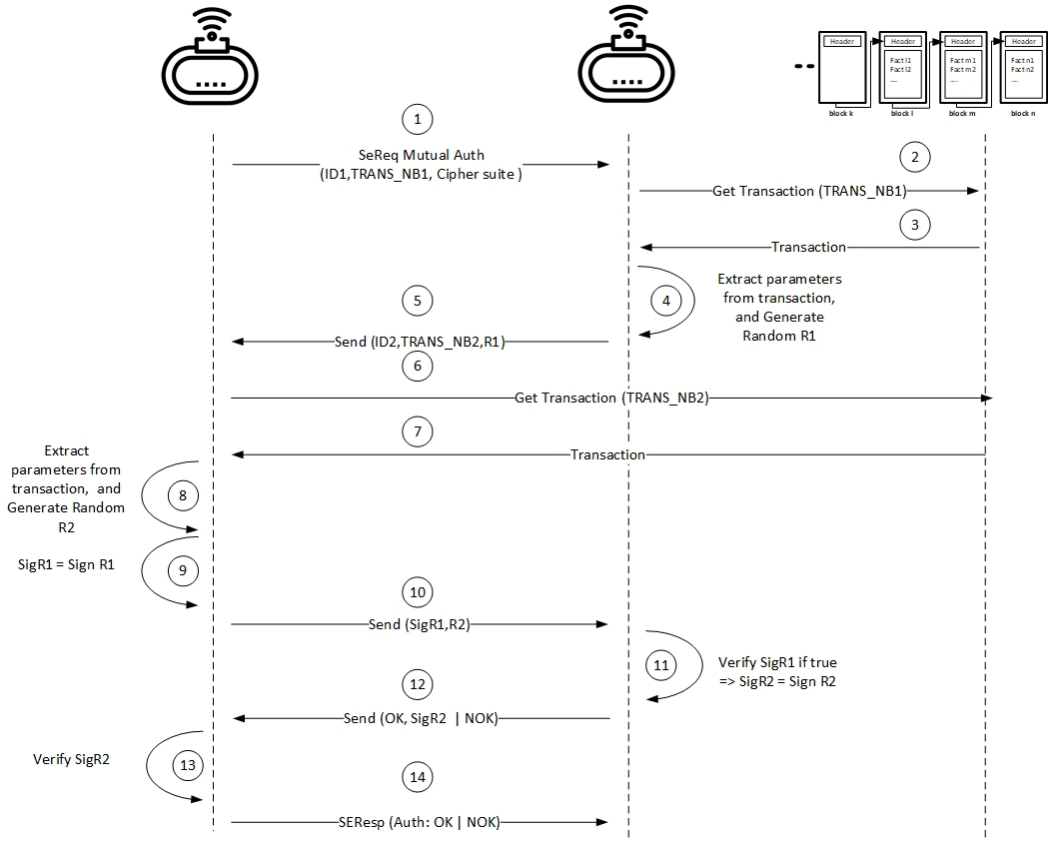


Figure 5.3 – Mutual authentication use case scenario

the devices used. The nodes communicate through a *Bluetooth* network. We used *Ethereum* blockchain, where the tests have been provided on the *Ethereum Testnet Ropsten* and relying on the *Infura* Application Programming Interface (API) [2]. Figure 5.4 illustrates our implementation framework.

Node type	**CPU**	**Memory (RAM)**	**Network**
Raspberry PI 3 Model B	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz	1GB LPDDR2	Bluetooth 4.2

Table 5.1 – Experimentation node Specifications

For signature scheme, we used the *Elliptic Curve Digital Signature Algorithm (ECDSA)*, given its multiple advantages over traditional signature algorithms (e.g., RSA) especially concerning key sizes and signature times, which makes it more adapted to IoT contexts [199] [200][3]. Our ECDSA implementation relies on the *SecP256k1* elliptic curve.

[2]https://infura.io

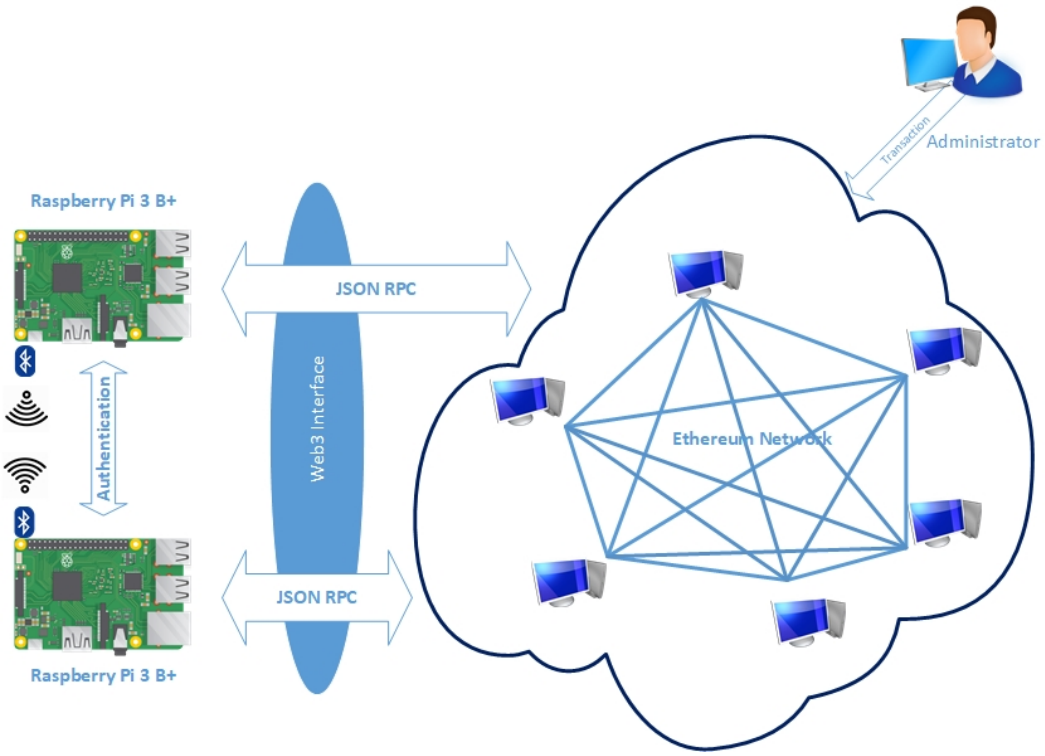


Figure 5.4 – Evaluation framework

5.3.2.2 Numerical results

Table 5.2 and Table 5.3 describe the results obtained following our experimentation set. The results presented represent the average and standard deviation of the 100 experimentations realized in order to measure the time needed to perform an authentication for both scenarios.

	O_1	O_2		
	Signature Time (s)	**Verification Time (s)**	**Time to get transaction (s)**	**Total Authentication time (s)**
Average	0.046	0.025	0.151	0.227
SD	0.002	0.010	0.029	0.037

Table 5.2 – Simple authentication scenario - Experimentation results

Regarding the simple authentication scenario, our approach has an average time of 0.227 s. The standard deviation obtained is low with 0.037 s which demonstrates the approach stability. For the mutual authentication scenario, our approach realized an average time of 0.450 s. The standard deviation obtained is always low with 0.058 s. We are aware that the results obtained by our approach can be out-

		Average	SD
O_1	**Signature time (s)**	0.043	0.008
	Verification time (s)	0.037	0.006
	Transaction time (s)	0.155	0.035
O_2	**Signature time (s)**	0.028	0.015
	Verification time (s)	0.029	0.01
	Transaction time (s)	0.157	0.049
	Total Authentication time (s)	0.45	0.058

Table 5.3 – Mutual authentication scenario - Experimentation results

performed by some of the existing centralized authentication methods. However, the obtained values are fully adapted to such environment. Moreover, our approach is completely decentralized. Furthermore, it ensures a fully mobility of the nodes. In other words, if a node moves to any location, any other node can obtain its authentication parameters without any initialization phase since all the authentication parameters are accessible through the public blockchain used.

5.4 An adaptive authentication and authorization scheme for IoT's gateways

The third contribution of the thesis is an adaptive authentication and authorization scheme for IoT's gateways. It was designed to satisfy a number of requirements/needs presented in section 1.2, which can be summarized as follow:

1. Adaptability to heterogeneous authentication techniques in order to ensure flexible and more resilient security.

2. High scalability to allow the mobility of nodes while ensuring their authentication at the gateway level

3. Easy integration of new devices as well as new services.

In that sense, in the proposed blockchain-based authentication and authorization approach:

1. Any device can be added without any physical intervention

2. Can move freely and still be authenticated and authorized at the gateway level

3. It meets the IoT scalability requirements

The proposed scheme can be applied to a huge number of IoT use cases and does not require special hardware. In this section, we describe the different details related to the design and the functioning of the proposed approach.

5.4.1 Initialization phase

When a new device is added to the network, the user that adds it must register it. This is provided by sending a transaction to the blockchain, which contains the following information about the added object: (1) Object ID; (2) Authentication method (e.g., PSK, One Time Password (OTP), Certificate, etc.); (3) Authentication parameters (e.g, the PSK, the object's public key, the object's certificate, etc.); (4) Authorization list (e.g., only upload to server IP on $Port$, upload/download to/from All, upload data of maximum 10 bytes per session, etc.). Except the object ID, all the parameters are encrypted as shown in the example of the block content below.

```
========================================
ObjectID: E43AC16A93
=======BEGIN ENCRYPTION==========
Authentication Method: PSK
Authentication Params: DE6S4ZB$CN1U0
Authorization List: {
Only Download From IP:Port
Only Upload To     IP:Port
}
=======END ENCRYPTION===============
========================================
```

Within the scope of our work, we focus mainly on the protocol exchanges between the objects and the gateways. The details regarding: (1) how the information are encrypted in the Blockchain and which encryption type is used, (2) the blockchain type and choice, (3) how the gateways downloads safely the blocks and decipher them, are outside the scope and will be subject of a future work.

5.4.2 Authentication Approach

First, when a device wants to establish a communication session with the gateway, it sends it a Session Establishment Request (*SEReq*) that contains the object ID and its authentication parameters (e.g., PSK). When the gateway receives the request, and based on the Object ID, it downloads from the blockchain, the block containing the parameters related to that requester object. Then, the gateway decrypts the block and according to the retrieved parameters, it triggers the authentication operation. Afterwards, a Session Establishment Response (*SERep*) is sent to the device to inform it whether it is successfully authenticated or not. Finally, if there is a successful authentication, then, the session establishment can be set up.

Once the device successfully authenticated and the session established, the gateway controls each exchange and communication of the object relying on the list of authorization downloaded within the block. Indeed, since the gateway represents

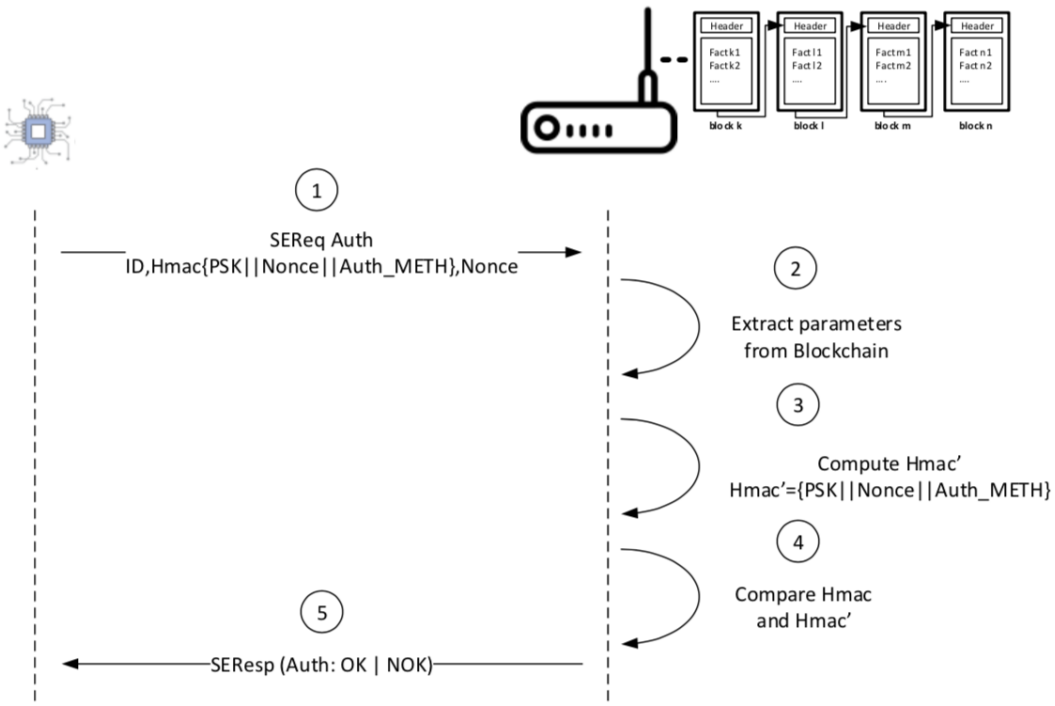


Figure 5.5 – Adaptive authentication approach: PSK use case

the sole link between the object and all its surroundings, it can control each action outgoing/coming from/to the object.

In our approach, any existing authentication method can be used as it is designed. Thus, the *SEReq* parameters are secured as it is the case in the method's classical use. Nonetheless, If the authentication method requires the usage of secret information like PSK, as highlighted in Figure 5.5, we propose a customized *SEReq* format to protect against man in the middle combined to cryptanalysis attacks as below:

```
================================================================
SEReq = [ID, HMAC{PSK || Nonce || Authentication Method}, Nonce]
================================================================
```

Once the gateway receives this type of request, it retrieves the needed authentication parameters from the blockchain according to the received object ID. Then, it computes `HMAC(PSK || Nonce || Authentication Method)` using the received Nonce and the deciphered data. If the obtained HMAC[3] matches the received one, it successfully authenticates the requester device.

[3] Any other hash algorithm can be used, and this according to the hosting system's requirements, needs and capacity

5.4.3 Context and use case scenarios

As described earlier, the main advantage of our proposed approach relies in its suitability to the majority of IoT scenarios, all within ensuring an easy integration of new devices and services. In this section, we evaluate our approach regarding its execution time as well as its financial cost.
For this evaluation, we choose two different use cases of the authentication between a gateway and an object:

1. PSK-based authentication

2. Mutual Certificate-based authentication

Knowing that the usage of the blockchain induces an additional cost, for both scenarios, we measure the time needed for the authentication: (1) in a classical scenario, without relying on our approach, and (2) using our adaptive approach.
Also, knowing that the operation of searching a block in the blockchain is feasible in a sequential method (block by block). To verify if the position of the block that owns the parameters needed for the authentication may induce to a additional cost, we use a blockchain having 1000 blocks and for both scenarios, we measure the time needed for the authentication in the following cases: (1) the parameters needed for the authentication are in the first block of the blockchain; (2) the parameters needed for the authentication are in the block 500 of the blockchain; and (3) the parameters needed for the authentication are in the last block of the blockchain (1000).

5.4.4 Evaluation framework

In order to evaluate the time consumption of our approach, we used two Virtual Machines (VMs) as end nodes. The first VM was designed as the gateway that hosts the blockchain and the second as a smart object. Table 5.4 describes the VM features. The authentication scheme was implemented in `Java`.

CPU	CPU max speed	RAM	Operating System
x86_4, 32 bits	2 GHz	256 MB	Debian 7.8

Table 5.4 – Experimentation VM feature

We are aware about the difference of performance between VMs and some of the common smart objects. We are also aware about the fact that the obtained results depends on the used language (Java) and are different when using other languages (e.g., `C/C++`). Nonetheless, the goal of our evaluation is the evaluation of our approach's additional cost in comparison to classical methods. Consequently, the comparison is fair since all the protocol's operations are realized on the same basis, language and material.

5.4.5 Performance and Security Evaluation

5.4.5.1 Qualitative Evaluation

Our approach is an adaptive scheme that allows the usage of other authentication methods. Thus, some of the security features and robustness such as integrity, confidentiality, non-repudiation, robustness against replay and spoofing attack, protection against Sybil attacks, etc. depend only on the chosen scheme. Therefore, in this section we focus on the evaluation of the security and performance features, that must be satisfied by an adaptive authentication approach:

Scalability: our scheme relies on a blockchain, which, in turn, relies on a P2P network, which is of the best solutions to meet large scalability requirement [3][201].

Mobility of nodes: in our approach, the parameters related to the authentication and the authorization are stored in a blockchain. Since the blockchain is a decentralized system, all the gateways which are the peers that host the blockchain, host an updated version of the latter, which contains all the parameters needed to the authentication. Thus the nodes' mobility does not represent any obstacle, since any gateway can read the blockchain.

Heterogeneity of supported approaches: our scheme represents a way to extract and send the parameters needed and used by other schemes. Any method parameters can be stored, thus, our approach supports the use of a multitude of schemes.

Initialization phase: in order to add a new device, the user needs only to add the parameters needed for the authentication and authorization in the blockchain, which does not require any physical intervention on the gateways. Hence, we propose a very lightweight initialization approach, that brings numerous savings in comparison to other methods.

Robustness against cryptanalysis attacks: our scheme uses the other methods as they are designed. Thus, the robustness to cryptanalysis attacks are exactly the same as in classical cases. Nevertheless, we proposed a customized SAReq (described in Section 5.4.2) where only a hash (HMAC) on the parameters is sent. Therefore, an attacker that intercepts this data, cannot reconstruct the original data, since hash algorithms are injective functions.

5.4.5.2 Quantitative Evaluation

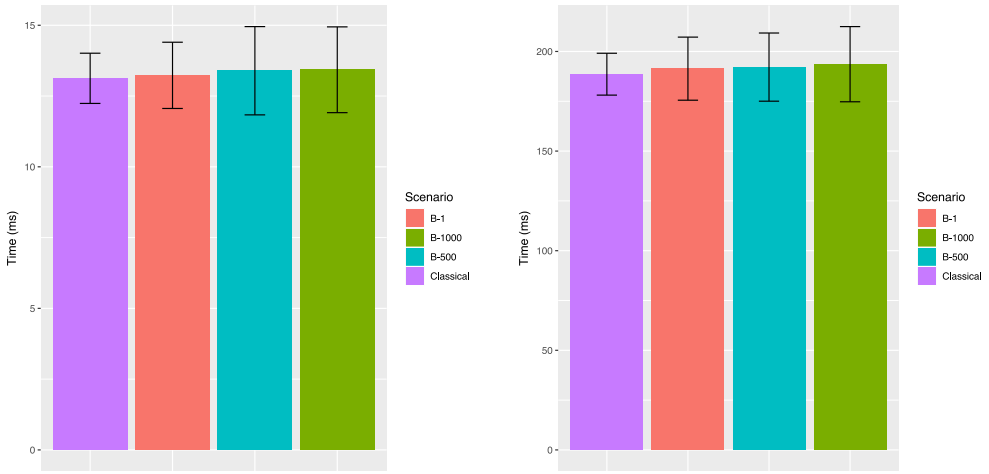
In this section, we present the numerical results obtained upon the time consumption evaluation of our approach. As described in Section 5.4.3, for each use case (authentication with PSK and mutual authentication with certificate) we measured 4 values of the authentication time:

1. Without our approach, which we present as *Classical case* in the results

2. The parameters needed for the authentication are in the first block of the blockchain, presented as *B-1*

3. The parameters needed for the authentication are in the block 500 of the blockchain, presented as *B-500*

4. The parameters needed for the authentication are in the last block of the blockchain, presented as *B-1000*



(a) PSK use case (b) Mutual Auth. with certificates

Figure 5.6 – Authentication time

The Figure 5.6a exhibits the results obtained from the experimentation of the first use case (authentication using PSK). Each result describes the average obtained through the execution of the same scenario **200 times**.
Without surprise, the *Classical* method realizes the best authentication time with 13.13 milliseconds (ms) and a standard deviation of 0.88 ms. Then, closer is the needed block in the blockchain, better is the time needed for the authentication, getting the following times: 13.23 ms, 13.39 ms and 13.43 ms, respectively for the scenarios *B-1*, *B-500* and *B-1000*. Also, the standard deviations obtained are very narrow with respectively 1.17 ms, 1.56 ms and 1.52 ms. One can note that the additional cost caused by our approach is negligible.

The Figure 5.6b describes the results of the second use case: mutual authentication using certificates. As in the last experimentation, each result represents the average obtained through the execution of the same scenario 200 times. We obtained the same ranking of the last experimentation for the needed time to realize the mutual authentication with 188.60 ms, 191.37 ms, 192.15 ms and 193.62 ms, respectively for the *Classical*, *B-1*, *B-500* and *B-1000* scenarios. Similarly to the last use case, the standard deviations obtained are very narrow with respectively 10.51 ms, 15.84 ms, 17.11 ms and 18.87 ms. In this case also, the additional cost caused by our approach is insignificant.

Finally, the time needed to find the block in the blockchain depends on the blockchain used. For example, the time needed to look for a transaction in Bitcoin

is not the same as for another blockchain less used, due to the number of blocks and transactions stored. However, optimization approaches can still be applied.

Regarding the financial cost related to the blockchain use, there are two cases: (1) if a private blockchain is used, then, there is no need for transaction payment. (2) if a public blockchain is used than, for each device, only one transaction is needed during the initialization phase and which contains the parameters that must be stored in the blockchain. Then, during the system functioning, the gateways do only reading operations from the blockchain, which are free-cost operations. The cost of one transaction depends on the blockchain used. However, it remains, generally, an insignificant cost. Moreover, according to studies like [202] and [203], the evolution of the cryptocurrencies rates will get more stable over time. Even better, *Ethereum* developers and community are working on regulating and stabilizing the amounts of fees related to smart contracts use[4] [3].

Consequently, considering these experimentations and results, one can conclude about the lightness and the low cost of our approach, while bringing more flexibility to the authentication system.

5.5 Conclusion

This chapter presented two blockchain-based authentication solutions for IoT systems. The proposed solutions take advantage of the inherent blockchain features to satisfy the IoT requirements imposed by the special nature and constraints of IoT systems.
The first proposed scheme is a lightweight authentication method, that does not require a special hardware, and thus is suitable for numerous IoT use case scenarios. This scheme was implemented in Java, tested and evaluated for two authentication scenarios: simple and mutual authentication. The results obtained confirmed its lightweightness and thus appropriateness for IoT devices.
The second proposed scheme is an adaptive approach that ensures the authentication and the authorization for IoT devices. In addition to meeting the general IoT requirements, this scheme addresses some practical limitations of existing solutions related to their deployment and usage in real use case scenarios. The performance and security evaluation of this scheme showed clearly its lightweightness and low cost.
The next chapter summarizes the research work done in this thesis and presents the directions for our future works.

[4] https://smartereum.com/6777/buterin-expresses-concern-over-stabilizing-ethereum/.

Chapter 6

General Conclusion and Future Directions

The penetration of Internet of Things (IoT) in the different aspects of our daily lives has significantly increased in the last years. Although the estimations might differ between the different studies [204] [13], but they all agree that the growth is exponential, and is expected to continue in the same manner for the upcoming years.
The idea behind IoT is the omnipresence of a variety of objects, where they are able to interact and cooperate with each other in order to provide a wide range of services. Thus, IoT introduces new opportunities such as the capability to monitor and manage devices remotely, analyze and take actions based on the information received from various real-time traffic data streams.
Consequently, with the advent of smart cities concept [14], IoT products are changing habits and cities by enhancing infrastructures, creating more effective and cost-efficient municipal services, improving transportation services by decreasing road traffic congestion, improving citizens' safety and providing smart health services [15].
IoT includes a large number of heterogeneous networked objects. Each object should be reachable and produce content that can be retrieved by any authorized user regardless of his/her location. It is important that only authenticated and authorized users (objects or people) can access IoT objects. Otherwise, they will be prone to numerous security risks such as information theft and identity spoofing. For these reasons and others, the security of IoT is a major challenge for the sustainability and competitiveness of companies and administrations. Indeed, security issues remain major obstacles to the worldwide adoption and deployment of IoT. In other words, users will not fully adopt IoT as long as IoT devices pose security and privacy risks. In fact, IoT is highly vulnerable to attacks for numerous reasons, such as:

1. Devices spend most of their time unattended in general, which makes them fairly easy to physically attack

2. Most of the communications are wireless, which makes Man-in-the-Middle attack, one of the most common attacks on such a system. Consequently, exchanged messages may be subject to eavesdropping, malicious routing, message tampering and other attacks which affect the security of the entire IoT

3. Multiple types of things (e.g., RFID tags) have limited resources in terms of energy and computation power, which prevent them from implementing advanced security solutions

IoT relies on the cooperation of nodes and multiple scenarios require that only trusted users can use the offered services. Thus, conventional security requirements such as authentication, data integrity, and sometimes confidentiality are critical to IoT objects, networks, and applications functioning. However, due to limitations and heterogeneity of objects' resources, existing security solutions are not fully adapted to IoT ecosystem.
Furthermore, in order to ensure these security requirements, the combination of multiple security technologies and solutions is needed which leads to high computation costs (in general one solution for authentication and integrity where at least one key sharing technique is applied, and another solution for confidentiality where another key sharing technique is applied).
Consequently, it is necessary to propose new lightweight security solutions or to adapt existing ones to meet the requirements of IoT.

This thesis consisted of two main parts: the literature review and the thesis contributions.

The first part (chapters 2 and 3) surveyed the literature of Internet of Things in general with a focus on the security and special emphasis on the authentication in IoT systems. This in-depth survey established in its first part, the necessary technical background for the IoT, their proposed architecture(s), the advances of research and development efforts, the technologies and platforms available for the market and the core IoT application domains. In addition, it allowed to identify the main challenges facing the IoT systems such as their resource-constrained nature, the heterogeneity of devices, scalability, big data management and most importantly the security of such systems. The security challenges constituted the general domain of this research work, and thus, the second part of the literature review has drawn the general IoT security context with special emphasis on the IoT authentication field; the specific domain of this research work. This has been done by analyzing the general IoT security objectives, issues, and challenges, and then shedding the light on those related to the authentication security goal through a comprehensive survey of the research works in such field. This led the identification of a number of open-issues that need to be addressed to fill the existing gaps in this domain .

The second part of this thesis built on the outcomes of the first part, in a sense that it addresses issues that are still unsolved (partially or completely) in the state of the art, especially from a practical (i.e. deployment) perspective (lightweightness, heterogeneity of IoT systems, scalability, node mobility support, etc.), and proposes three contributions to advance the literature in this research field.
The first contribution is a lightweight authentication scheme for IoT objects based on ECC and Isogeny offering two security services: authentication and keys generation. Our scheme relies on One Time Password (OTP), a very promising solution for IoT given its robustness and simplicity. Basically, OTP method is used to ensure authentication without having to use a third party server. The scheme did not use OTP only for authentication, but, also to generate a One Time Key (OTK) derived from OTP in order to be used by encryption algorithms. Hence, our scheme enhances data confidentiality in addition to ensuring the authentication while being more efficient in terms of computation than existing solutions. Furthermore, this approach has the originality of not relying on a Counter or a Timestamp as for synchronous OTP-type approaches, neither on a challenge generation and management from the server side as for the asynchronous OTP-type solutions. Furthermore, a special attention was given to the trade-off between the performance and the resources available on IoT devices. Extensive evaluations through implementation showed promising results making the new scheme as a good candidate for a lightweight authentication method for IoT applications.

The second and third contributions of the thesis are based on blockchain; the decentralized distributed ledger. This choice is based on the inherent features of blockchain (e.g., scalability, robustness, resilience) to address the requirements of authentication solutions as well as the limitations of some existing solutions.
The second contribution is a lightweight authentication scheme, supporting simple and mutual authentication scenarios, without the need for specific hardware which makes it applicable in a multitude of IoT use cases. The proposed scheme was evaluated through a real implementation relying on Ethereum blockchain and using different devices in order to confirm its feasibility and evaluate its initial performances. The results obtained confirm its suitability to such environments.
The third contribution provided the authentication and the authorization for IoT devices. It has an important advantage of being adaptive in a sense that it adapts to heterogeneous authentication techniques in order to ensure flexible and more resilient security. The security of the proposed solution was evaluated analytically, while the performance evaluation was done experimentally through a real implementation. The results of its evaluation showed clearly its lightness and low cost.

Future research directions

The future research directions can be seen as the continuity in the research works done in the three contribution of this thesis:

1. First contribution:
   (a) The use of Isogeny is a key element that improves the security and robustness of the first contribution. In the proposed scheme, we have used the Isogeny of Translation as a Proof-of-Concept. Other types of Isogeny shall be considered and evaluated.
   (b) The definition of a mechanism for the retransmission of lost and non-authenticated messages
   (c) The security of the proposed protocol especially against Denial of Service attacks that attempt to saturate the server's CPU by sending fake OTPs that the server must verify.
2. Second Contribution:
   (a) The development of the proposed approach in order to support the authorization management.
   (b) The comparison of the performance of this approach with existing approaches.
3. Third Contribution:
   (a) The design and specification of the storage of protocol data elements in the blockchain and the related operations with special emphasis on the security of such data.

Bibliography

[1] VL Shivraj, MA Rajan, Meena Singh, and P Balamuralidhar. One time password authentication scheme based on elliptic curves for internet of things (iot). In *Information Technology: Towards New Smart World (NSITNSW), 2015 5th National Symposium on*, pages 1–6. IEEE, 2015. (Cited on pages viii, 12, 13 and 49.)

[2] Mohamed Tahar Hammi, Erwan Livolant, Patrick Bellot, Ahmed Serhrouchni, and Pascale Minet. A lightweight mutual authentication protocol for the iot. In *International Conference on Mobile and Wireless Technology*, pages 3–12. Springer, 2017. (Cited on pages viii, 13, 49 and 50.)

[3] Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. *Computers & Security*, 78:126–142, 2018. (Cited on pages viii, 9, 11, 13, 52, 89, 95 and 97.)

[4] Mountain View, David M'Raihi, Frank Hoornaert, David Naccache, Mihir Bellare, and Ohad Ranen. HOTP: An HMAC-Based One-Time Password Algorithm. RFC 4226, December 2005. (Cited on pages viii, 13, 63 and 76.)

[5] Mountain View, Johan Rydell, Mingliang Pei, and Salah Machani. TOTP: Time-Based One-Time Password Algorithm. RFC 6238, May 2011. (Cited on pages viii, 13, 64 and 76.)

[6] Damian Christie. IoT Standards – Why So Many? `https://www.linkedin.com/pulse/iot-standards-why-so-many-damian-christie`, 2016. (Cited on pages 1 and 18.)

[7] Wordpress. The Internet of Things Protocol stack – from sensors to business value. `https://entrepreneurshiptalk.wordpress.com/2014/01/29/the-internet-of-thing-protocol-stack-from-sensors-to-business-value/`, 2014. (Cited on pages 1 and 20.)

[8] LoRa-Alliance. LoRaWAN, What is it. A Technical Overview of LoRa and LoRaWAN. Technical report, LoRa Alliance. Technical Marketing Workgroup 1.0, November 2015. (Cited on pages 1, 21 and 22.)

[9] Padraig Scully. 5 Things To Know About The IoT Platform Ecosystem. https://iot-analytics.com/5-things-know-about-iot-platform/, 2016. (Cited on pages 1, 24 and 26.)

[10] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu. Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 285–292, June 2015. (Cited on pages 1, 28 and 29.)

[11] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid - the new and improved power grid: A survey. *IEEE Communications Surveys Tutorials*, 14(4):944–980, Fourth 2012. (Cited on pages 1, 29 and 30.)

[12] CNXSOFT. Comparison Table of Low Power WAN Standards for Industrial Applications. http://www.cnx-software.com/2015/09/21/comparison-table-of-low-power-wan-standards-for-industrial-applications/, 2015. (Cited on pages 3 and 23.)

[13] Knud Lasse Lueth. Iot 2019 in review: The 10 most relevant iot developments of the year. Online, January 2020. (Cited on pages 7 and 99.)

[14] Rida Khatoun and Sherali Zeadally. Smart cities: concepts, architectures, research opportunities. *Communications of the ACM*, 59(8):46–57, 2016. (Cited on pages 7, 8 and 99.)

[15] Badis Hammi, R Khatoun, Sherali Zeadally, Achraf Fayad, and Lyes Khoukhi. Internet of Things (IoT) Technologies for Smart Cities. *IET Networks*, 2017. (Cited on pages 7, 9, 15 and 99.)

[16] krebsonsecurity inc. Hacked cameras, dvrs powered today's massive internet outage. Online, October 2016. (Cited on page 8.)

[17] John Biggs. Hackers release source code for a powerful DDoS app called Mirai. https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-appcalled-mirai/, October 2016. (Cited on pages 8 and 35.)

[18] Ken Borgendale Andrew Banks, Ed Briggs and Rahul Gupta. Mqtt version 5.0, March 2019. (Cited on page 8.)

[19] Z. Shelby, K. Hartke, and C. Bormann. Constrained application protocol (coap). RFC 7252, June 2014. (Cited on pages 8, 18 and 47.)

[20] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core, March 2011. RFC 6120, Internet Engineering Task Force. (Cited on page 8.)

[21] Achraf Fayad, Badis Hammi, and Rida Khatoun. An adaptive authentication and authorization scheme for iot's gateways: a blockchain based approach. In *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, pages 1–7. IEEE, 2018. (Cited on page 9.)

[22] Amir-Mohammad Rahmani, Nanda Kumar Thanigaivelan, Tuan Nguyen Gia, Jose Granados, Behailu Negash, Pasi Liljeberg, and Hannu Tenhunen. Smart e-health gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, pages 826–834. IEEE, 2015. (Cited on page 11.)

[23] Emmanouil Vasilomanolakis, Jörg Daubert, Manisha Luthra, Vangelis Gazis, Alex Wiesmaier, and Panayotis Kikiras. On the security and privacy of Internet of Things architectures and systems. In *Secure Internet of Things (SIoT), 2015 International Workshop on*, pages 49–57. IEEE, 2015. (Cited on page 11.)

[24] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer Networks*, 76:146–164, 2015. (Cited on page 11.)

[25] Kim Thuat Nguyen, Maryline Laurent, and Nouha Oualha. Survey on secure communication protocols for the internet of things. *Ad Hoc Networks*, 32:17–31, 2015. (Cited on page 11.)

[26] Jorge Granjal, Edmundo Monteiro, and Jorge Sá 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. (Cited on page 11.)

[27] Debiao He and Sherali Zeadally. An analysis of rfid authentication schemes for internet of things in healthcare environment using elliptic curve cryptography. *IEEE internet of things journal*, 2(1):72–83, 2015. (Cited on pages 11 and 42.)

[28] Mohammed El-hajj, Ahmad Fadlallah, Maroun Chamoun, and Ahmed Serhrouchni. A survey of internet of things (iot) authentication schemes. *Sensors*, 19(5):1141, Mar 2019. (Cited on pages 11, 45 and 48.)

[29] Haodong Wang, Bo Sheng, Chiu C Tan, and Qun Li. Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control. In *The 28th international conference on distributed computing systems*, pages 11–18. IEEE, 2008. (Cited on pages 12 and 52.)

[30] Parikshit N Mahalle, Bayu Anggorojati, Neeli R Prasad, and Ramjee Prasad. Identity authentication and capability based access control (iacac) for the internet of things. *Journal of Cyber Security and Mobility*, 1(4):309–348, 2013. (Cited on page 12.)

[31] Chen-Xu Liu, Yun Liu, Zhen-Jiang Zhang, and Zi-Yao Cheng. The novel authentication scheme based on theory of quadratic residues for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013. (Cited on page 12.)

[32] Sharu Bansal and Dilip Kumar. Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication. *International Journal of Wireless Information Networks*, 2020. (Cited on page 16.)

[33] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du. Research on the architecture of internet of things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 5, pages V5–484–V5–487, 2010. (Cited on page 16.)

[34] Ibrahim Mashal, Osama Alsaryrah, Tein-Yaw Chung, Cheng-Zen Yang, Wen-Hsing Kuo, and Dharma P. Agrawal. Choices for interaction with things on internet and underlying issues. *Ad Hoc Networks*, 28:68 – 90, 2015. (Cited on page 17.)

[35] I. Chen, J. Guo, and F. Bao. Trust management for soa-based iot and its application to service composition. *IEEE Transactions on Services Computing*, 9(3):482–495, 2016. (Cited on page 17.)

[36] Rajesh Khanna, Pallavi Sethi, and Smruti R. Sarangi. Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017:9324035, 2017. (Cited on page 17.)

[37] International Telecommunication Union. Next Generation Networks - Frameworks and functional architecture models. Recommendation ITU-T Y.2060. Overview of the Internet of things. Technical report, International Telecommunication Union, 2012. (Cited on page 17.)

[38] Oladayo Bello, Sherali Zeadally, and Mohamad Badra. Network layer inter-operation of device-to-device communication technologies in internet of things (iot). *Ad Hoc Networks*, 57:52 – 62, 2017. Special Issue on Internet of Things and Smart Cities: security, privacy and new technologies. (Cited on pages 18 and 32.)

[39] Hanna Okkonen, Oleksiy Mazhelis, Petri Ahokangas, Pasi Pussinen, Mervi Rajahonka, Riikka Siuruainen, Seppo Leminen, Alexey Shveykovskiy, Jenni Myllykoski, and Henna Warma. Internet-of-things market, value networks, and business models: state of the art report. *Computer science and information systems reports. TR, Technical reports 39.*, 2013. (Cited on page 18.)

[40] Xue Li, Jing Liu, Quan Z. Sheng, Sherali Zeadally, and Weicai Zhong. Tms-rfid: Temporal management of large-scale rfid applications. *Information Systems Frontiers*, 13(4):481–500, 2011. (Cited on page 18.)

[41] Quan Z Sheng, Xue Li, and Sherali Zeadally. Enabling next-generation rfid applications: Solutions and challenges. *Computer*, 41(9), 2008. (Cited on page 18.)

[42] LoRa-Alliance. Wide Area Networks For IoT. https://www.lora-alliance.org, 2017. (Cited on page 21.)

[43] LoRa Alliance Technical Committee. Lorawan 1.1 specification. Online, February 2017. (Cited on page 21.)

[44] LoRa Alliance. LoRa Technology. https://www.lora-alliance.org/What-Is-LoRa/Technology, 2017. (Cited on page 21.)

[45] Keith E Nolan, Wael Guibene, and Mark Y Kelly. An evaluation of low power wide area network technologies for the Internet of Things. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 439–444. IEEE, 2016. (Cited on pages 21 and 22.)

[46] Atmel. Ata8520 datasheet. Online, November 2015. (Cited on page 22.)

[47] Vangelis Gazis, Manuel Görtz, Marco Huber, Alessandro Leonardi, Kostas Mathioudakis, Alexander Wiesmaier, Florian Zeiger, and Emmanouil Vasilomanolakis. A survey of technologies for the internet of things. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 1090–1095. IEEE, 2015. (Cited on page 24.)

[48] Miyuru Dayarathna. Comparing 11 IoT Development Platforms. https://dzone.com/articles/iot-software-platform-comparison, 2016. (Cited on page 24.)

[49] iot analytics.com. IoT Platforms: Market Report 2015-2021. Technical report, IoT Analytics Insights for the Internet of Things, 2016. (Cited on page 24.)

[50] Amazon Web Services. AWS IoT. https://aws.amazon.com/iot-platform/, 2017. (Cited on page 25.)

[51] Oracle. Oracle Internet of Things Cloud Service. https://aws.amazon.com/iot-platform/, 2017. (Cited on page 25.)

[52] Microsoft. Microsoft Research, Lab of Things. http://www.lab-of-things.com, 2013. (Cited on page 25.)

[53] OpenRemote. OpenRemote is the Open Source Middleware for the Internet of Things. http://www.openremote.com, 2016. (Cited on page 25.)

[54] KAA. KAA: The truly open-source Kaa IoT Platform. https://www.kaaproject.org, 2014. (Cited on page 25.)

[55] ThingsBoard. ThingsBoard: Open-source IoT Platform. https://thingsboard.io, 2017. (Cited on page 25.)

[56] Plotly. Plotly: Visualize Data, Together. https://plot.ly, 2017. (Cited on page 25.)

[57] IBM. IBM Watson IoT Platform. https://internetofthings.ibmcloud.com/#/, 2017. (Cited on page 25.)

[58] KII. KII Platform. https://en.kii.com, 2016. (Cited on page 25.)

[59] Echelon. Echelon: Industrial Internet of Things. http://www.echelon.com/izot-platform, 2017. (Cited on page 25.)

[60] Axeda. Axeda Machine Cloud. https://www.ptc.com/en/axeda, 2017. (Cited on page 25.)

[61] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431 – 440, 2015. (Cited on page 24.)

[62] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*, 1(4):349–359, 2014. (Cited on page 26.)

[63] 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. (Cited on page 26.)

[64] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008. (Cited on page 27.)

[65] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*, 1(2):112–121, April 2014. (Cited on page 27.)

[66] M. T. Lazarescu. Design of a wsn platform for long-term environmental monitoring for iot applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(1):45–54, March 2013. (Cited on page 27.)

[67] D. C. Bogatinoska, R. Malekian, J. Trengoska, and W. A. Nyako. Advanced sensing and internet of things in smart cities. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 632–637, May 2016. (Cited on page 27.)

[68] R. Fujdiak, P. Masek, P. Mlynek, J. Misurec, and E. Olshannikova. Using genetic algorithm for advanced municipal waste collection in smart city. In *2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pages 1–6, July 2016. (Cited on page 27.)

[69] Z. Pala and N. Inanc. Smart parking applications using rfid technology. In *RFID Eurasia, 2007 1st Annual*, pages 1–3, Sept 2007. (Cited on page 28.)

[70] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication Systems*, 50(4):217–241, 2012. (Cited on page 28.)

[71] R. Lu, X. Lin, H. Zhu, and X. Shen. Spark: A new vanet-based smart parking scheme for large parking lots. In *INFOCOM 2009, IEEE*, pages 1413–1421, April 2009. (Cited on page 28.)

[72] C. C. Y. Poon, Yuan-Ting Zhang, and Shu-Di Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine*, 44(4):73–81, April 2006. (Cited on page 28.)

[73] Marcus Handte, Stefan Foell, Stephan Wagner, Gerd Kortuem, and Pedro José Marrón. An Internet-of-Things Enabled Connected Navigation System for Urban Bus Riders. *IEEE internet of things journal*, 3(5):735–744, 2016. (Cited on page 28.)

[74] J. Liu, X. Li, X. Chen, Y. Zhen, and L. Zeng. Applications of internet of things on smart grid in china. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, pages 13–17, Feb 2011. (Cited on page 29.)

[75] International Transport Forum. Urban Mobility System Upgrade How shared self-driving cars could change city traffic. Technical report, OECD/International Transport Forum, 2015. (Cited on page 29.)

[76] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011. (Cited on page 31.)

[77] Ken Traub, Greg Allgair, Henri Barthel, Leo Burstein, John Garrett, Bernie Hogan, Bryan Rodrigues, Sanjay Sarma, Johannes Schmidt, Chuck Schramek, et al. The EPCglobal architecture framework. *EPCglobal Ratified specification*, 2005. (Cited on page 32.)

[78] Sang-Do Lee, Myung-Ki Shin, and Hyoung-Jun Kim. EPC vs. IPv6 mapping mechanism. In *Advanced Communication Technology, The 9th International Conference on*, volume 2, pages 1243–1245. IEEE, 2007. (Cited on page 32.)

[79] Dong Geun Yoon, Dong Hyeon Lee, Chang Ho Seo, and Seong Gon Choi. RFID networking mechanism using address management agent. In *Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on*, volume 1, pages 617–622. IEEE, 2008. (Cited on page 32.)

[80] PC Jain and KP Vijaygopalan. RFID and wireless sensor networks. *Proceedings of ASCNT–2010, CDAC, Noida, India*, pages 1–11, 2010. (Cited on page 32.)

[81] EPCGlobal. Ratified Standard Specification with Approved, Fixed Errata. EPCglobal Object Name Service (ONS) 1.0.1. Technical report, EPCglobal Inc, May 2008. (Cited on page 32.)

[82] GS1. Ratified Standard. GS1 Object Name Service (ONS) Version 2.0.1. Technical report, GS1, January 2013. (Cited on page 32.)

[83] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010. (Cited on pages 32 and 33.)

[84] Oladayo Bello and Sherali Zeadally. Intelligent device-to-device communication in the internet of things. *IEEE Systems Journal*, 10(3):1172–1182, 2016. (Cited on page 32.)

[85] TV Lakshman and Upamanyu Madhow. The performance of tcp/ip for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking (ToN)*, 5(3):336–350, 1997. (Cited on page 32.)

[86] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute, 2011. (Cited on page 33.)

[87] Saint John Walker. Big data a revolution that will transform how we live work and think, 2014. (Cited on page 33.)

[88] Rob Kitchin. The real-time city? big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014. (Cited on pages 33 and 34.)

[89] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014. (Cited on page 33.)

[90] J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo. Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, 22(6):122–128, 2015. (Cited on page 34.)

[91] Michael Batty. Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279, 2013. (Cited on page 34.)

[92] Zaheer Khan, Ashiq Anjum, and Saad Liaquat Kiani. Cloud based big data analytics for smart future cities. In *Proceedings of the 2013 IEEE/ACM 6th international conference on utility and cloud computing*, pages 381–386. IEEE Computer Society, 2013. (Cited on page 34.)

[93] Yogesh Simmhan, Saima Aman, Alok Kumbhare, Rongyang Liu, Sam Stevens, Qunzhi Zhou, and Viktor Prasanna. Cloud-based software platform for big data analytics in smart grids. *Computing in Science & Engineering*, 15(4):38–47, 2013. (Cited on page 34.)

[94] Internet of Things: Privacy and Security in a Connected World. Technical report, US Federal Trade Commission (FTC), 2015. (Cited on page 37.)

[95] ENISA. Iot security standards gap analysis. techreport, European Union Agency for Cybersecurity, January 2019. (Cited on page 37.)

[96] NIST-CSRC. Computer security resource center - online glossary. Online, 2020. (Cited on pages 38, 39, 40 and 41.)

[97] Musa G. Samaila, Miguel Neto, Diogo A. B. Fernandes, Mário M. Freire, and Pedro R. M. Inácio. *Security Challenges of the Internet of Things*, pages 53–82. Springer International Publishing, Cham, 2017. (Cited on pages 39 and 40.)

[98] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla. Challenges of multi-factor authentication for securing advanced iot applications. *IEEE Network*, 33(2):82–88, 2019. (Cited on page 39.)

[99] Dipankar Dasgupta, Arunava Roy, and Abhijit Nag. *Multi-Factor Authentication*, pages 185–233. Springer International Publishing, Cham, 2017. (Cited on page 39.)

[100] Internet-Society. Internet society policy brief:iot privacy for policymakers. Online, September 2019. (Cited on page 40.)

[101] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, August 2010. v0.34. (Cited on page 41.)

[102] Vu Tuan Anh, Pham Quoc Cuong, and Phan Cong Vinh. Context-aware mobility based on pi-calculus in internet of thing: A survey. In *Context-Aware Systems and Applications, and Nature of Computation and Communication*, pages 38–46. Springer, 2019. (Cited on page 41.)

[103] Richard K. Lomotey, Joseph Pry, and Sumanth Sriramoju. Wearable iot data stream traceability in a distributed health information system. *Pervasive and Mobile Computing*, 40:692 – 707, 2017. (Cited on page 41.)

[104] Karol Furdik, Ferry Pramudianto, Matts Ahlsén, Peter Rosengren, Peeter Kool, Song Zhenyu, Paolo Brizzi, Marek Paralic, and Alexander Schneider. Food traceability chain supported by the ebbits iot middleware. In Herbert Kotzab, Jürgen Pannek, and Klaus-Dieter Thoben, editors, *Dynamics in Logistics*, pages 343–353, Cham, 2016. Springer International Publishing. (Cited on page 41.)

[105] Thomas K. Dasaklis, Fran Casino, and Constantinos Patsakis. Defining granularity levels for supply chain traceability based on iot and blockchain. In *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, COINS '19, page 184–190, New York, NY, USA, 2019. Association for Computing Machinery. (Cited on page 41.)

[106] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda. Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, pages 1–4, 2018. (Cited on page 41.)

[107] Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: Perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, 2014. (Cited on page 42.)

[108] Kai Han, Jun Luo, Yang Liu, and Athanasios V Vasilakos. Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, 51(7):107–113, 2013. (Cited on page 42.)

[109] David J Malan, Matt Welsh, and Michael D Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 71–80. IEEE, 2004. (Cited on page 42.)

[110] Mo Li, Zhenjiang Li, and Athanasios V Vasilakos. A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12):2538–2557, 2013. (Cited on page 42.)

[111] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996. (Cited on page 42.)

[112] J Hoffstein, N Howgrave-Graham, J Pipher, JH Silverman, and W Whyte. NTRUEncrypt and NTRUSign: efficient public key algorithms for a post-quantum world. In *Proceedings of the International Workshop on Post-Quantum Cryptography (PQCrypto 2006)*, pages 71–77, 2006. (Cited on page 42.)

[113] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006. (Cited on page 42.)

[114] Laurent Eschenauer and Virgil 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. (Cited on pages 42 and 43.)

[115] Gunnar Gaubatz, J-P Kaps, Erdinc Ozturk, and Berk Sunar. State of the art in ultra-low power public key cryptography for wireless sensor networks. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 146–150. IEEE, 2005. (Cited on page 42.)

[116] Adrian Perrig, Robert Szewczyk, Justin Douglas Tygar, Victor Wen, and David E Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002. (Cited on page 42.)

[117] Arvinderpal S Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 324–328. IEEE, 2005. (Cited on page 42.)

[118] TV Landegem and H Viswanathan. Anywhere, anytime, immersive communications [j]. *Enriching Communications*, 2(1):1–6, 2008. (Cited on page 43.)

[119] Mohamed Amine Ferrag, Leandros A Maglaras, Helge Janicke, Jianmin Jiang, and Lei Shu. Authentication protocols for internet of things: a comprehensive survey. *Security and Communication Networks*, 2017, 2017. (Cited on pages 44 and 45.)

[120] Mohammed El-hajj, Maroun Chamoun, Ahmad Fadlallah, and Ahmed Serhrouchni. Taxonomy of authentication techniques in internet of things (iot). In *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*, pages 67–71. IEEE, 2017. (Cited on page 44.)

[121] S. Agrawal and P. Ahlawat. A survey on the authentication techniques in internet of things. In *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, pages 1–5, 2020. (Cited on pages 44 and 45.)

[122] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer. Implementation and characterization of a physical unclonable function for iot: A case study with the tero-puf. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):97–109, 2018. (Cited on page 44.)

[123] Ygal Bendavid, Nasour Bagheri, Masoumeh Safkhani, and Samad Rostampour. Iot device security: Challenging "a lightweight rfid mutual authentication protocol based on physical unclonable function". *Sensors*, 18(12):4444, Dec 2018. (Cited on page 44.)

[124] J. Yang, Y. Lin, Y. Fu, X. Xue, and B. A. Chen. A small area and low power true random number generator using write speed variation of oxidebased rram for iot security application. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017. (Cited on page 44.)

[125] S. U. Hussain, M. Majzoobi, and F. Koushanfar. A built-in-self-test scheme for online evaluation of physical unclonable functions and true random number generators. *IEEE Transactions on Multi-Scale Computing Systems*, 2(1):2–16, 2016. (Cited on page 44.)

[126] Hala Hamadeh, Soma Chaudhuri, and Akhilesh Tyagi. Area, energy, and time assessment for a distributed tpm for distributed trust in iot clusters. *Integration*, 58:267 – 273, 2017. (Cited on page 44.)

[127] H. Tan, G. Tsudik, and S. Jha. Mtra: Multiple-tier remote attestation in iot networks. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2017. (Cited on page 44.)

[128] C. Lesjak, D. Hein, and J. Winter. Hardware-security technologies for industrial iot: Trustzone and security controller. In *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pages 002589–002595, 2015. (Cited on page 44.)

[129] S. Pinto, T. Gomes, J. Pereira, J. Cabral, and A. Tavares. Iioteed: An enhanced, trusted execution environment for industrial iot edge devices. *IEEE Internet Computing*, 21(1):40–47, 2017. (Cited on page 44.)

[130] G. Ayoade, V. Karande, L. Khan, and K. Hamlen. Decentralized iot data management using blockchain and trusted execution environment. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 15–22, 2018. (Cited on page 44.)

[131] Ashok Kumar Das, Sherali Zeadally, and Debiao He. Taxonomy and analysis of security protocols for internet of things. *Future Generation Computer Systems*, 89:110 – 125, 2018. (Cited on page 44.)

[132] Kai Zhao and Lina Ge. A survey on the internet of things security. In *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, pages 663–667. IEEE, 2013. (Cited on page 45.)

[133] Yahya Atwady and Mohammed Hammoudeh. A survey on authentication techniques for the internet of things. In *Proceedings of the International Conference on Future Networks and Distributed Systems*, 2017. (Cited on page 45.)

[134] Mohammad Wazid, Ashok Kumar Das, Rasheed Hussain, Giancarlo Succi, and Joel JPC Rodrigues. Authentication in cloud-driven iot-based big data environment: Survey and outlook. *Journal of Systems Architecture*, 97:185–196, 2019. (Cited on page 45.)

[135] Thomas Kothmayr, Corinna Schmitt, Wen Hu, Michael Brünig, and Georg Carle. Dtls based security and two-way authentication for the internet of things. *Ad Hoc Networks*, 11(8):2710 – 2723, 2013. (Cited on page 47.)

[136] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu. A robust authentication scheme for observing resources in the internet of things environment. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 205–211, 2014. (Cited on page 47.)

[137] Sanaz Rahimi Moosavi, Tuan Nguyen Gia, Amir-Mohammad Rahmani, Ethiopia Nigussie, Seppo Virtanen, Jouni Isoaho, and Hannu Tenhunen. Sea : A secure and efficient authentication and authorization architecture for iot-based healthcare using smart gateways. *Procedia Computer Science*, 52:452–459, 2015. QC 20150618. (Cited on page 47.)

[138] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. Internet of things security: A survey. *Journal of Network and Computer Applications*, 88:10 – 28, 2017. (Cited on page 48.)

[139] Sheetal Kalra and Sandeep K. Sood. Secure authentication scheme for iot and cloud servers. *Pervasive and Mobile Computing*, 24:210 – 223, 2015. Special Issue on Secure Ubiquitous Computing. (Cited on page 48.)

[140] Saru Kumari, Marimuthu Karuppiah, Ashok Kumar Das, Xiong Li, Fan Wu, and Neeraj Kumar. A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers. *The Journal of Supercomputing*, 74(12):6428–6453, 2018. (Cited on page 48.)

[141] C. Schmitt, M. Noack, and B. Stiller. Chapter 13 - tinyto: two-way authentication for constrained devices in the internet of things. In Rajkumar Buyya and Amir [Vahid Dastjerdi], editors, *Internet of Things*, pages 239 – 258. Morgan Kaufmann, 2016. (Cited on page 48.)

[142] J. L. Hernández-Ramos, M. P. Pawlowski, A. J. Jara, A. F. Skarmeta, and L. Ladid. Toward a lightweight authentication and authorization framework for smart objects. *IEEE Journal on Selected Areas in Communications*, 33(4):690–702, 2015. (Cited on page 48.)

[143] S. Raza, L. Seitz, D. Sitenkov, and G. Selander. S3k: Scalable security with symmetric keys—dtls key establishment for the internet of things. *IEEE Transactions on Automation Science and Engineering*, 13(3):1270–1280, 2016. (Cited on page 48.)

[144] H. Khemissa and D. Tandjaoui. A lightweight authentication scheme for e-health applications in the context of internet of things. In *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 90–95, 2015. (Cited on page 48.)

[145] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos. A lightweight authentication mechanism for m2m communications in industrial iot environment. *IEEE Internet of Things Journal*, 6(1):288–296, 2019. (Cited on page 48.)

[146] K Aravindhan and RR Karthiga. One time password: A survey. *International Journal of Emerging Trends in Engineering and Development*, 1(3):613–623, 2013. (Cited on page 49.)

[147] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert. Sms-based one-time passwords: attacks and defense. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 150–159. Springer, 2013. (Cited on page 49.)

[148] Dan Boneh et al. Twenty years of attacks on the rsa cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999. (Cited on page 49.)

[149] Dinei Florêncio and Cormac Herley. One-time password access to any server without changing the server. In *International Conference on Information Security*, pages 401–420. Springer, 2008. (Cited on page 49.)

[150] Yunjin Lee and Howon Kim. Insider attack-resistant otp (one-time password) based on bilinear maps. *International Journal of Computer and Communication Engineering*, 2(3):304, 2013. (Cited on page 49.)

[151] Vipul Goyal, Ajith Abraham, Sugata Sanyal, and Sang Yong Han. The n/r one time password system. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 1, pages 733–738. IEEE, 2005. (Cited on page 49.)

[152] Ashok Kumar Mohan and T Gireesh Kumar. Secure seed-based sturdy otp via convenient carry-on device. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, pages 447–455. Springer, 2015. (Cited on page 49.)

[153] Mohamed Tahar Hammi, Erwan Livolant, Patrick Bellot, Ahmed Serhrouchni, and Pascale Minet. A lightweight iot security protocol. In *Cyber Security in Networking Conference (CSNet), 2017 1st*, pages 1–8. IEEE, 2017. (Cited on page 49.)

[154] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 618–623. IEEE, 2017. (Cited on page 49.)

[155] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Blockchain in Internet of Things: challenges and solutions. *arXiv preprint arXiv:1608.05187*, 2016. (Cited on page 49.)

[156] Thomas Hardjono and Ned 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. (Cited on page 50.)

[157] Mohamed Tahar Hammi, Patrick Bellot, and Ahmed Serhrouchni. BCTrust: A decentralized authentication blockchain-based mechanism. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pages 1–6. IEEE, 2018. (Cited on page 50.)

[158] Umair Khalid, Muhammad Asim, Thar Baker, Patrick C. K. Hung, Muhammad Adnan Tariq, and Laura Rafferty. A decentralized lightweight blockchain-based authentication mechanism for iot systems. *Cluster Computing*, 2020. (Cited on page 50.)

[159] B. K. Mohanta, A. Sahoo, S. Patel, S. S. Panda, D. Jena, and D. Gountia. Decauth: Decentralized authentication scheme for iot device using ethereum blockchain. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 558–563, 2019. (Cited on page 50.)

[160] J. Won, A. Singla, E. Bertino, and G. Bollella. Decentralized public key infrastructure for internet-of-things. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 907–913, 2018. (Cited on page 50.)

[161] Guntur Dharma Putra, Volkan Dedeoglu, Salil S. Kanhere, and Raja Jurdak. Trust management in decentralized iot access control system, 2019. (Cited on page 50.)

[162] P.S. Febin Sheron, K.P. Sridhar, S. Baskar, and P. Mohamed Shakeel. A decentralized scalable security framework for end-to-end authentication of future iot communication. *Transactions on Emerging Telecommunications Technologies*, n/a(n/a):e3815, November 2019. e3815 ETT-19-0503.R1. (Cited on page 51.)

[163] Ashutosh Dwivedi, Gautam Srivastava, Shalini Dhar, and Rajani Singh. A decentralized privacy-preserving healthcare blockchain for iot. *Sensors*, 19(2):326, Jan 2019. (Cited on page 51.)

[164] Almudena Alcaide, Esther Palomar, José Montero-Castillo, and Arturo Ribagorda. Anonymous authentication for privacy-preserving iot target-driven applications. *Computers & Security*, 37:111 – 123, 2013. (Cited on page 51.)

[165] R. Xu, Y. Chen, E. Blasch, and G. Chen. Blendcac: A blockchain-enabled decentralized capability-based access control for iots. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1027–1034, 2018. (Cited on page 51.)

[166] Pelin Angin, Melih Burak Mert, Okan Mete, Azer Ramazanli, Kaan Sarica, and Bora Gungoren. A blockchain-based decentralized security architecture for iot. In Dimitrios Georgakopoulos and Liang-Jie Zhang, editors, *Internet of Things – ICIOT 2018*, pages 3–18, Cham, 2018. Springer International Publishing. (Cited on page 51.)

[167] Axel Moinet, Benoît Darties, and Jean-Luc Baril. Blockchain based trust & authentication for decentralized sensor networks. *CoRR*, abs/1706.01730, 2017. (Cited on page 51.)

[168] Abdallah Zoubir Ourad, Boutheyna Belgacem, and Khaled Salah. Using blockchain for iot access control and authentication management. In Dimitrios Georgakopoulos and Liang-Jie Zhang, editors, *Internet of Things – ICIOT 2018*, pages 150–164, Cham, 2018. Springer International Publishing. (Cited on page 52.)

[169] Arjen K Lenstra. Key lengths. Technical report, Wiley, 2006. (Cited on page 58.)

[170] Christof Paar and Jan Pelzl. *Understanding Cryptography - A Textbook for Students and Practitioners.* Springer, 2010. (Cited on page 58.)

[171] Dindayal Mahto and DILIP YADAV. Rsa and ecc: A comparative analysis. *International Journal of Applied Engineering Research*, 12:9053–9061, 01 2017. (Cited on page 58.)

[172] Elaine Barker and Quynh Dang. Nist special publication 800-57 part 1, revision 5: Recommendation for key management: Part 1 – general, May 2020. (Cited on page 58.)

[173] D McGrew, K Igoe, and Margaret Salter. Fundamental elliptic curve cryptography algorithms. RFC 6090, February 2011. (Cited on page 60.)

[174] Lily Chen, Dustin Moody, Andrew Regenscheid, and Karen Randall. Sp 800-186-draft: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters, October 2019. (Cited on page 60.)

[175] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, 1(1):36–63, 2001. (Cited on page 61.)

[176] Sec 1: Elliptic curve cryptography. version 2.0. *Standards for Efficient Cryptography*. (Cited on page 61.)

[177] Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017. (Cited on page 61.)

[178] Philippe Guillot. *Courbes elliptiques: une présentation élémentaire pour la cryptographie*. Hermes science-Lavoisier, 2010. (Cited on pages 61 and 62.)

[179] Dr. Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, February 1997. (Cited on page 63.)

[180] NIST. Fips 140-3: Security requirements for cryptographic modules. Technical report, apr 2019. (Cited on page 64.)

[181] Dieter Bong and Andreas Philipp. *Securing the Smart Grid with Hardware Security Modules*, pages 128–136. Springer Fachmedien Wiesbaden, Wiesbaden, 2012. (Cited on page 64.)

[182] S. Turner, D. Brown, K. Yiu, R. Housley, and T. Polk. RFC 5480: Elliptic Curve Cryptography Subject Public Key Information. *IETF, March*, 2009. (Cited on page 68.)

[183] SEC 2: Recommended Elliptic Curve Domain Parameters. Version 2.0. *Standards for Efficient Cryptography*, January 2010. (Cited on page 68.)

[184] M. Lochter and J. Merkle. RFC 5639: Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. *IETF, March*, 2010. (Cited on page 68.)

[185] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, pages 17–36, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. (Cited on page 69.)

[186] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. (Cited on pages 80, 82 and 83.)

[187] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14:352, 10 2018. (Cited on pages 80, 81 and 83.)

[188] Matthew Beedham. All you need to know about bitcoin network nodes, March 2019. (Cited on page 80.)

[189] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009. (Cited on pages 82 and 83.)

[190] Arati Baliga. Understanding blockchain consensus models, 2017. (Cited on pages 82 and 83.)

[191] bitshares 2.0: distributed consensus,"bitshares whitepapers", 2015. (Cited on page 83.)

[192] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186, USA, 1999. USENIX Association. (Cited on page 83.)

[193] Shengyun Liu, Christian Cachin, Vivien Quéma, and Marko Vukolic. Xft: Practical fault tolerance beyond crashes. *CoRR*, abs/1502.05831, 2015. (Cited on page 83.)

[194] David Schwartz, Noah Youngs, Arthur Britto, et al. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5(8), 2014. (Cited on page 83.)

[195] David Mazières. The stellar consensus protocol: A federated model for internet-level consensus, 2015. (Cited on page 83.)

[196] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2014. (Cited on page 84.)

[197] B. K. Mohanta, S. S. Panda, and D. Jena. An overview of smart contract and use cases in blockchain technology. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4, 2018. (Cited on page 84.)

[198] Christian Cachin. Architecture of the hyperledger blockchain fabric, 2016. (Cited on page 84.)

[199] Kristin Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications*, 11(1):62–67, 2004. (Cited on page 89.)

[200] Erik De Win, Serge Mister, Bart Preneel, and Michael Wiener. On the performance of signature schemes based on elliptic curves. In *International Algorithmic Number Theory Symposium*, pages 252–266. Springer, 1998. (Cited on page 89.)

[201] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93, 2005. (Cited on page 95.)

[202] Kenji Saito and Mitsuru Iwamura. How to Make a Digital Currency on a Blockchain Stable. *arXiv preprint arXiv:1801.06771*, pages 1–15, 2018. (Cited on page 97.)

[203] Ousmène Jacques Mandeng. Cryptocurrencies, Monetary Stability and regulation. Technical report, 2018. (Cited on page 97.)

[204] C STAMFORD. Gartner says by 2020, more than half of major new business processes and systems will incorporate some element of the internet of things. Technical report, Gartner, Inc,, 2016. (Cited on page 99.)

Publications

Journal articles

- Badis HAMMI, Rida KHATOUN, Sherali ZEADALLY, Achraf FAYAD, Lyes KHOUKHI. Internet of Things (IoT) Technologies for Smart Cities. In *IET Journals*, 2017
- Badis HAMMI, Achraf FAYAD, Rida KHATOUN, Sherali ZEADALLY. A Lightweight ECC-Based Authentication Scheme for Internet of Things (IoT). In *IEEE Systems Journal*, 2020.

International conferences

- Achraf FAYAD, Badis HAMMI, Rida KHATOUN. An adaptive authentication and authorization scheme for IoT's gateways: a blockchain based approach. In *Proceedings of the 2018 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC'2018)*, Shanghai, China. October 2018.
- Achraf FAYAD, Badis HAMMI, Rida KHATOUN, Ahmed Serhrouchni. A Blockchain-based Lightweight Authentication Solution for IoT. In *Proceedings of the 2019 International Conference on Cyber Security In Networking, (CSNet'2019)*, Quito, Euador. October 2019.

ECOLE
DOCTORALE

Titre : Protocole d'authentification sécurisé pour les objets connectés

Mots clés : Objets connectés, securité, authentification, autorisation

Résumé : L'interconnexion de ressources privées sur une infrastructure publique, la mobilité des utilisateurs ainsi que l'émergence des nouvelles technologies (réseaux véhiculaires, réseaux de capteurs, Internet des objets (Internet of Things (IoT)), etc) ont ajouté des nouvelles exigences en terme de sécurité du côté client et serveur. L'IoT semble répondre à des usages bien accueillis par le grand public. On trouve ainsi les applications de l'IoT dans tous les domaines de la vie quotidienne. Les objets connectés sont ainsi prévus en très grand nombre et à très grande échelle. La sécurité est l'élément majeur qui renforcera d'une manière certaine une acceptation encore plus grande de l'IoT par les citoyens et les entreprises. Par ailleurs, le déploiement à grande échelle des objets connectés, sera convoité par les attaques de tout bord. Les cyberattaques opérationnelles sur les réseaux traditionnels seront projetées vers l'internet des objets. La sécurité est ainsi critique dans ce contexte au vu des enjeux sous-jacents.
Les travaux de recherche menés dans cette thèse visent à faire avancer la littérature sur l'authentification des objets connectés, en proposant trois schémas d'authentification qui répondent aux besoins des systèmes IoT en termes de sécurité et de performance. L'OTP (One-Time Password) est une méthode d'authentification qui représente une solution prometteuse pour les environnements des objets connectés et les villes intelligentes. Ce travail de recherche étend le principe OTP et propose une méthode d'authentification légère utilisant une nouvelle approche de la génération OTP qui s'appuie sur la cryptographie à courbe elliptique (ECC) et l'Isogénie pour garantir la sécurité des systèmes sous-jacents. Les résultats de performance obtenus démontrent l'efficacité de notre approche en termes de sécurité et de performance.
Par la suite, nous nous appuyons sur les caractéristiques de la blockchain pour proposer deux solutions d'authentification : premièrement, une solution d'authentification légère basée sur Ethereum, et deuxièmement, une approche adaptative d'authentification et d'autorisation. Nous avons fourni une implémentation de ces approches. Nous avons également mené une évaluation approfondie qui montre clairement la capacité des solutions proposées à répondre aux différentes exigences de sécurité avec un coût léger en terme de performance.

Title : Secure authentication protocol for Internet of Things

Keywords : Internet of Things, security, authentication, autorization

Abstract : The interconnection of private resources on public infrastructure, the user mobility and the emergence of new technologies (vehicular networks, sensor networks, Internet of Things (IoT), etc.) have added new requirements in term of security on the server side as well as the client side. Examples include the processing time, mutual authentication, client participation in the choice of security settings and protection against traffic analysis. IoT is in widespread use and its applications cover many aspects of today's life, which results in a huge and continuously increasing number of objects distributed everywhere.
Security is no doubt the element that will improve and strengthen the acceptability of IoT, especially that this large scale deployment of IoT systems will attract the appetite of the attackers. The current cyber-attacks that are operational on traditional networks will be projected towards the Internet of Things. Security is so critical in this context given the underlying stakes ; in particular, authentication has a critical importance given the impact of the presence of malicious nodes within the IoT systems and the harm they can cause to the overall system.
The research works in this thesis aim to advance the literature on IoT authentication by proposing three authentication schemes that satisfy the needs of IoT systems in terms of security and performance. One-Time Password (OTP) is an authentication scheme that represents a promising solution for IoT and smart cities environments. This research work extends the OTP principle and proposes a new approach to generate OTP based on Elliptic Curve Cryptography (ECC) and Isogeny to guarantee the security of such protocol. The performance results obtained demonstrate the efficiency and effectiveness of our approach in terms of security and performance.
We also rely on blockchains in order to propose two authentication solutions : first, a lightweight blockchain-based authentication scheme for IoT systems based on Ethereum, and second, an adaptive blockchain-based authentication and authorization approach for IoT use cases. We provided a real implementation of our proposed solutions. The extensive evaluation provided, clearly shows the ability of our schemes to meet the different security requirements with a lightweight cost in term of performance.

Institut Polytechnique de Paris
91120 Palaiseau, France