Distributed management framework based on the blockchain technology for industry 4.0 environments

Asma Lahbib

INSTITUT POLYTECHNIQUE DE PARIS

NNT: 2020IPPAS017

Thèse de doctorat

Distributed Management Framework based on the Blockchain Technology for Industry 4.0 Environments

Thèse de doctorat de l'Institut Polytechnique de Paris préparée à **Telecom SudParis**

École doctorale n° ED 626 Ecole doctorale IP Paris (ED IPP)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Evry, le 23/11/2020, par

ASMA LAHBIB

Composition du jury:

Joaquin GARCIA-ALFARO Professeur, Telecom SudParis, France	Président
Mohamed Hamdi Maitre de Conférence, SupCom, Tunisie	Rapporteur
Riadh Dhaou Maitre de Conférence, INPT/ENSEEIHT, France	Rapporteur
Imen Ayari Directrice de l'Innovation factory, Talan, Tunisie	Examinatrice
Anis Laouiti Professeur, Telecom SudParis, France	Directeur
Khalifa Toumi Expert Blockchain, IRT SystemX, France	Coencadrant
Steven Martin Professeur, LRI, France	Invité
Cedric Adjih Chargé de Recherche, Inria Saclay, France	Invité

1

Abstract

The evolution of the Internet of Things (IoT) started decades ago as part of the first face of the digital transformation, its vision has further evolved due to a convergence of multiple technologies, ranging from wireless communication to the Internet and from embedded systems to micro-electromechanical systems. As a consequence thereof, IoT' platforms are being heavily developed, smart factories are being planned to revolutionize the industry organization and both security and trust requirements are becoming more and more critical. The integration of such technologies within the manufacturing environment and processes in combination with other technologies such as Cloud Computing, Cyber Physical Systems, Information and Communication Technologies as well as Enterprise Architecture has introduced the fourth industrial revolution referred to also as Industry 4.0. In this future world machines will talk to machines (M2M) to organize the production and coordinate their actions function of the information collected by different sensors and exchanged with other entities. However opening connectivity to the external world raises several questions about data security that was not an issue when devices were controlled locally and just few of them were connected to some other remote systems. That's why ensuring a secure communication between heterogeneous and reliable devices is essential to protect exchanged information from being stolen or tampered by malicious cyber attackers that may harm the production processes and put the different devices out of order. Without appropriate security solutions, these systems will never be deployed globally due to all kinds of security concerns. That's why ensuring a secure and trusted communication between heterogeneous devices and within dynamic and decentralized environments is essential to achieve users acceptance of such solutions. However, building a secure system does not only mean protecting the data exchange but it requires also building a system where the source of data and the data itself is being trusted by all participating devices and stakeholders.

In this thesis our research focused on four complementary issues, mainly (i) the dynamic and trust based management of access over shared resources within an Industry 4.0 based distributed and collaborative system, (ii) the establishment of a privacy preserving solution for related data in a decentralized architecture while eliminating the need to rely on additional third parties, (iii) the verification of the safety, the correctness and the functional accuracy of the designed framework and last (iv) the evaluation of the trustworthiness degree of interacting parties in addition to the secure storage and sharing of computed trust scores among them in order to guarantee their confidentiality, integrity and privacy.

By focusing on such issues and taking into account the conventional characteristics of both IoT and IIoT based environments, we proposed in this thesis a secure and distributed framework for resource management in Industry 4.0 environments. The proposed framework, enabled by the blockchain technology and driven by peer to peer networks, allows not only the dynamic access management over shared resources but also the distribute governance of the system without the need for third parties that could be their-selves vulnerable to attacks. Besides and in order to ensure strong privacy guarantees over the access control related procedures, a privacy preserving scheme is proposed and integrated within the distributed management framework. Furthermore and in order to guarantee the safety and the functional accuracy of our framework software components, we focused on their formal modeling in order to validate their safety

and compliance with their specification. Finally, we designed and implemented the proposal in order to prove its feasibility and analyze its performances.

Keywords: Blockchain, Security, Privacy, Access Control, Trust, IoT, IIoT

1

Resumé

En raison de l'avancement technologique dans les domaines des communications sans fil ainsi que ceux de l'informatique mobile et embarquée, les frontières entre le monde physique et le monde numérique se rétrécissent pour introduire une nouvelle technologie également appelée l'internet des objets (IdO). Cette technologie est en train de devenir une partie intégrante de nos vies. Allant des capteurs de puissance et capabilités réduites, aux appareils électroménagers intelligents, objets de télé-santé et véhicules et voitures connectées jusqu'aux usines interconnectées et intelligentes dans lesquelles les employés, les machines, les processus, les services et meme les produits interagissent entre eux de manière à fournir une meilleure adaptabilité dans la production ainsi qu'une allocation plus efficace des ressources, et ce, pour répondre plus rapidement au marché, d'une façon plus personnalisée et à moindre coût. L'intégration de cette technologie dans l'environnement et les processus de fabrication en combinaison avec d'autres technologies et paradigmes telles que le cloud computing (CC), les systèmes physiques cybernétiques (CPS), les technologies de l'information (IT) et de la communication ainsi que l'analyse des donnees et l'intelligence artificielle (IA) a introduit la quatrième révolution industrielle également appelée l'Industrie 4.0. Dans ce futur monde, les machines parleront aux machines (M2M) pour organiser la production et coordonner leurs actions en fonction des informations collectées et échangées entre les différents capteurs et entités. Cependant, l'ouverture de la connectivité a un monde externe soulève plusieurs soucis et questions sur la sécurité et la confidentialité des données produites et échangées qui n'étaient pas un problème lorsque les appareils étaient contrôlés localement et que seuls quelques-uns d'entre eux étaient connectés à d'autres systèmes distants. Les risques de sécurité liés à tels objets représentent des ressources potentielles pour les acteurs tiers et malveillants qui peuvent nuire aux processus de production au sein de l'usine et mettre les différents appareils hors service. Une fois compromises, ces ressources peuvent être utilisées dans des attaques à large échelle contre d'autres systèmes. C'est pourquoi il est essentiel d'assurer une communication sécurisée entre les différents appareils hétérogènes qui sont généralement déployés dans des environnements dynamiques et décentralisés pour protéger les informations échangées contre le vol ou la falsification par des cyberattaquants malveillants et par conséquent obtenir l'acceptation des utilisateurs de telles solutions. Dans cette direction, cette thèse est concentrée sur quatre questions complémentaires, principalement (1) la gestion dynamique de l'accès aux ressources partagées au sein d'un système distribué et collaboratif de l'industrie 4.0, (2) la protection des données personnelles et sensibles des utilisateurs tout au long des procédures de gestion d'access et tout en éliminant le besoin de s'appuyer sur des tiers supplémentaires, (3) la vérification formelle et la validation de la sécurité ainsi que l'exactitude du cadre conçu et enfin (4) la gestion de confiance des parties en interaction en plus du stockage sécurisé des informations relatives afin de garantir leur confidentialité, intégrité et traceabilité. En se concentrant sur telles questions et en tenant compte des caractéristiques conventionnelles des environnements IdO et IdO Industriels, nous avons proposé dans cette thèse un cadre générique sécurisé et decentralisé pour la gestion des ressources dans les environnements Industrie 4.0. Le cadre spécifié basé sur la technologie blockchain et piloté par un réseau peer to peer permet non seulement la gestion dynamique d'accès aux

ressources partagées mais aussi la governance distribuée du système, la protection de vie privée, la gestion de confiance et la vérification formelle des spécifications établies. Enfin, une conception et mise en œuvre de la plateforme sont assurées afin de prouver sa faisabilité et d'analyser ses performances.

Keywords: Blockchain, Securité, Controle d'access, Vie privée, Gestion de confiance, IdO, Industrie 4.0

1

Thesis Publications

Conference Papers

- **Lahbib Asma**, Khalifa Toumi, Sameh Elleuch, Anis Laouiti, and Steven Martin. Link reliable and trust aware RPL routing protocol for Internet of Things. In 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1-5. IEEE, 2017.

- **Lahbib Asma**, Khalifa Toumi, Anis Laouiti, Alexandre Laube, and Steven Martin. Blockchain based trust management mechanism for IoT. In 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-8. IEEE, 2019.

- **Lahbib Asma**, Khalifa Toumi, Anis Laouiti, and Steven Martin. DRMF: a Distributed Resource Management Framework for industry 4.0 environments. In 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), pp. 1-9. IEEE, 2019.

- **Lahbib Asma**, Abderrahim Ait Wakrime, Anis Laouiti, Khalifa Toumi, and Steven Martin. An Event-B Based Approach for Formal Modelling and Verification of Smart Contracts. In International Conference on Advanced Information Networking and Applications, pp. 1303-1318. Springer, Cham, 2020.

- **Lahbib Asma**, Khalifa Toumi, Anis Laouiti, and Steven Martin. Blockchain based Privacy Aware Distributed Access Management Framework for Industry 4.0. In 2021 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2021. (submitted)

Journal Papers

- **Lahbib Asma**, Khalifa Toumi, Anis Laouiti, and Steven Martin. Trust Based Routing Metric For RPL Routing Protocol IN The Internet Of Things, International Journal on AdHoc Networking Systems (IJANS), 2020

Acknowledgements

The time has come to end one of the most important stages of my life and on the eve of this moment I cannot let the occasion of the presentation of this report without expressing my thanks, my deep respect and gratitude to all those who were willing to provide necessary assistance to the success of this work.
I would like to begin by expressing my gratitude to Mr Mohamed HAMDI and Mr Riadh DHAOU for kindly accepting to review this manuscript. Also, my thanks go to the members of the jury: Mr Cedric ADJIH, Mrs Imen AYARI, and Mr Joaquin GARCIA-ALFARO for the interest they showed in my research and for the honor hey have granted to me by examining and evaluating this dissertation and hopefully this work will meet up to their standards.
I would never have been able to finish this thesis without the guidance, help and support of my colleagues, friends and family. I thank all those people who made this thesis possible and an unforgettable experience for me.
Firstly, I would like to express my deepest sense of gratitude to my advisors Mr Anis Laouiti and Mr Steven Martin for their continuous support of my PhD study, for their knowledge and immense encouragement and especially for their high personal and professional qualities. Their guidance helped me throughout the research and writing of this thesis. Thank you Anis for the advice, the encouragements, and the vote of confidence. I am lucky to have a mentor such as yourself.
I would like to express my special appreciation and gratitude to my co-advisor Mr Khalifa Toumi that made all of this possible. Thank you Khalifa for your interesting comments, advices, and reflections. You have provided me with valuable academic suggestions, relevant research ideas, and insightful inputs on my work. Thank you again for your patience and will to push me forward over those three years.
I have been honored to work with the members of RS2M Team at Telecom SudParis. Thank you: Ines, Nahit, Oumaima and Sameh for the pleasant working atmosphere you created. I am grateful to all of you, for your support as well as the fun times I have shared with you.
Besides, I would like to thank my family and friends. First and foremost, I am most grateful to my parents in Tunisia who continuously support and encourage me as always. They have provided me with a very stable family condition for so many years, which allows me to find and pursue my own life as I dreamed.
Finally, I have to thank my husband and love of my life, Houcem, for keeping things going, for always showing how proud he is of me and for standing by me through both the good times and the bad.
The last word goes for Youssef & Iyad, my lovely baby boys, who have been the light of my life for the last two years and who have given me the extra strength and motivation to get things done. This thesis is dedicated to them. ...

Contents

List of Figures

List of Tables

List of Abbreviations

6LoWPAN	IPv**6** **L**ow power **W**ireless **P**ersonal **A**rea **N**etwork
6TiSCH	IPv**6** over **Ti**me **S**lotted **C**hannel **H**opping
ABAC	**A**ttribute **B**ased **A**ccess **C**ontrol
AC	**A**ccess **C**ontrol
AMQP	**A**dvanced **M**essage **Q**ueuing **P**rotocol
CC	**C**loud **C**omputing
CoAP	**Co**nstrained **A**pplication **P**rotocol
CPS	**C**yber **P**hysical **S**ystem
CTL	**C**omputational **T**ree **L**ogic
DAO	**D**estination **A**dvertisement **O**bject
DAC	**D**iscretionary **A**ccess **C**ontrol
DAG	**D**irected **A**cyclic **G**raph
DDoS	**D**istributed **D**enial **o**f **S**ervice
DDS	**D**ata **D**istribution **S**ervice
DIO	**D**ODAG **I**nformation **O**bject
DNS	**D**omain **N**ame **S**ystem
DNS-SD	**D**omain **N**ame **S**ystem **S**ervice **D**iscovery
DODAG	**D**estination **O**riented **D**irected **A**cyclic **G**raph
DoS	**D**enial **o**f **S**ervice
DRMF	**D**istributed **R**esource **M**anagement **F**ramework
EA	**E**nterprise **A**rchitecture
EVM	**E**thereum **V**irtual **M**achine
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol
HTTPS	**H**yper**T**ext **T**ransfer **P**rotocol **S**ecure
IETF	**I**nternet **E**ngineering **T**ask **F**orce
ICT	**I**nformation and **C**ommunication **T**echnologies
IIoT	**I**ndustrial **I**nternet **o**f **T**hings
IoT	**I**nternet **o**f **T**hings
IP	**I**nternet **P**rotocol
JSON	**J**ava**S**cript **O**bject **N**otation
LCD	**L**iquid **C**rystal **D**isplay
LLN	**L**ow power and **L**ossy **N**etwork
LTE-A	**L**ong **T**erm **E**volution **A**dvanced
LTL	**L**inear **T**emporal **L**ogic
MAC	**M**andatory **A**ccess **C**ontrol
MANET	**M**obile **A**d-hoc **NET**work
MQTT	**M**essage **Q**ueuing **T**elemetry **T**ransport
NIZK	**N**on **I**nteractive **Z**ero **K**nowledge**P**roof
OIDC	**O**pen**ID** **C**onnect
OrBAC	**Or**ganization **B**ased **A**ccess **C**ontrol
P2P	**P**eer **2** **P**eer
PBFT	**P**ractical **B**yzantine **F**ault **T**olerance

PDP	**P**olicy **D**ecision **P**oint
PKI	**P**ublic **K**ey **I**nfrastructure
PoA	**P**roof **o**f **A**uthority
PoS	**P**roof **o**f **S**take
PoW	**P**roof **o**f **W**ork
QoS	**Q**uality **o**f **S**ervice
RBAC	**R**ole **B**ased **A**ccess **C**ontrol
REST	**RE**presentational **S**tate **T**ransfer
RPL	**R**outing **P**rotocol for **L**ow power and lossy networks
RR	**R**ound **R**obin
SANET	**S**ensor (and) **A**ctuator **NET**work
SHA	**S**ecure **H**ash **A**lgorithm
SC	**S**mart **C**ontract
TM	**T**rust **M**anagement
TMS	**T**rust **M**anagement **S**ystem
WSN	**W**ireless **S**ensor **N**etwork
XMPP	e**X**tensible **M**essaging and **P**resence **P**rotocol
ZKP	**Z**ero **K**nowledge **P**roof

To my lovely little baby boys Youssef & Iyad

Chapter 1
Introduction

Contents

1.1 Context and Motivation

The technological advancement in the fields of wireless communications and mobile computing has introduced a new technology also called IoT which, basically representing the integration of physical smart devices within the Internet infrastructure, has significantly emerged, grown and gradually affected daily lives of human beings promoting as a consequence thereof a new generation of innovative and valuable services provided by various application domains and industrial systems ranging from wearable devices, smart transportation, health care, smart grid and smart manufacturing systems, to mention a few.
Within such paradigm, smart connectivity, remote sensing and computation is an indispensable part. Their integration within the manufacturing environment and processes in combination with other technologies such Cloud Computing (CC), Cyber Physical Systems (CPS), Information and Communication Technologies (ICT) as well as Enterprise Architecture (EA), has introduced the fourth wave of the industrial revolution called also Industry 4.0.
In this regard, IoT is becoming an indispensable part for the design and the implementation of both Industry 4.0 and smart factories scenarios and applications. By connecting humans, machines, products and data processes, IoT will offer undoubtedly various new opportunities to the manufacturing systems making factories consequently more intelligent, greater resource efficient, higher reliable and correlated and obviously flexible and dynamic. As a consequence, the production process will be more efficient and flexible with products of higher quality. However, the deployment of such technologies in addition to the heterogeneous and constrained nature of IoT devices, is expected to intensify encountered security threats and issues. In fact, opening connectivity to the external world raises several questions about data and IT infrastructure security that were not an issue when devices and machines were controlled locally and just few of them were connected to some other remote systems. Launched threats and attacks can cause manufacturing disruptions, leading to defective products, production downtime, physical damage and even threaten lives.
According to a report from IBM X-Force Research in 2016, the manufacturing sector is the second most-attacked industry behind healthcare in 2015. Automotive manufacturers were the top targets for criminals, accounting for almost 30% of all cyberattacks, while chemical companies were attackers' second-favorite targets. The report sheds light also on the biggest cyber risks that organizations and manufacturing industries face today. In what follows, we cite some of the most popular security threats launched in last six years:

- German steel mill meltdown: In December 2014, attackers infiltrated a Germany steel mill facility and obtained control right of the cooperate and plant network, which led to the explosion of a furnace at that time [47].

- The Mirai attack: In October 2016, the Mirai IoT botnet [49] infected numerous IoT devices and initiated distributed Denial of Service (DoS) attacks through flooding DNS servers, which results in the large-scale Internet network paralysis.
- The Jeep hack: In July 2015, a team of researchers was able to take total control of a Jeep SUV over Sprint cellular network by exploiting a firmware update vulnerability. They discovered that they could control the speed of the vehicle as well as veer it off from the road [11].
- The Hackable Cardiac Devices from Abbott medical center: Implantable cardiac pacemakers from Abbott (formerly called St. Jude) medical center in Chicago have vulnerabilities that could allow a hacker to access them, to deplete the battery or administer incorrect pacing or shocks to patients.

Without appropriate security solutions, various IoT and IIoT systems will never be deployed globally due to all kinds of security concerns. That's why ensuring a secure and trusted communication between heterogeneous devices and within dynamic and decentralized environments is essential to achieve users acceptance and to protect exchanged information from being stolen or tampered by malicious cyber attackers that may harm the production processes and put the different devices out of order. However, building a secure system does not only mean protecting the data exchange but it requires also building a system where the source of data and the data itself is being trusted by all participating devices and stakeholders.
To improve privacy and security within such environments, a more decentralized approach is seen as the solution to allow the longterm growth of both IoT and IIoT. This refers mainly to the fact that participating entities when interacting and collaborating with each other, do not have to rely and to trust external services or third-parties to manage access rights over data they produce or they share. In this context, several proposals have been made for ensuring distributed systems in IoT environments and hence harvest scalability and security advantages. However, these last cannot easily solve the integrity, immutability, traceability, and notarization issues required for most use cases within such environments.

1.2 Problematic

In this context, the research problem that we address in this thesis is:

How to ensure a distributed collaborative system within an Industry 4.0 based environment, where the access is dynamically and trustfully managed over shared resources and where both users and data privacy are preserved during collaboration?

We decompose this research problem into four sub-problems.

- RP1: How to design a dynamic access control framework within a distributed collaborative system where collaborating parties are more empowered to control access over resources they share by defining their own policies and dynamically reconfiguring them in response to time, events and more importantly to entities' changing behavior and attitudes. This approach requires a trust management mechanism to be considered and integrated within the access control model for evaluating access requester entities' behavior. Such consideration would guarantee as a consequence thereof the dynamicity of security policies insofar that they would be defined and validated function of the access requester entity's behavior.

- RP2: How to ensure strong privacy guarantees over the access control related procedures regarding the access requester's sensitive attributes as well as the shared access control policies? When interacting parties from different sites and with diversified competing and conflicting interests, share common resources for which access is controlled and managed via a distributed and a consensus based management framework, their privacy may be breached. This is mainly due to the fact that their access history related details once recorded in a distributed storage structure, may conduct to tracing their access activities and learning their authorization functionality pattern. Also and in case of malicious competitors existence, although these last cannot read shared data files and resources related information, they can read stored access control policies, and thus they can deduce information about an organization current activity details, they can even try to satisfy the access condition in order to get authorized to the corresponding resource

- RP3: How to evaluate the trustworthiness degree of interacting parties and how to securely store and share computed trust scores among them in order to guarantee their confidentiality, integrity and privacy? To evaluate entities trustworthiness, exchanging trust information is crucial to reach an accurate assessment. Besides, trust information may contain sensitive information about trust providers and targets. Hence, there is a need to secure not only the network but also to ensure integrity and trustworthiness of exchanged information. Consequently secure sharing and storage of trust information is inevitably crucial for its confidentiality, integrity and immutability especially in IoT enabled industries environments where devices collect a large amount of data conveying important information for critical decision making.

- RP4: How to guarantee the safety, the correctness and the functional accuracy of the designed framework? Generally speaking, customers and end users acceptance of any kind of proposed technologies and frameworks depends mainly on the fact that their software components provide services that behave correctly and satisfy their requirements. The safety, correctness and reliability of such systems include two dimensions. One is to regard developed software components as a

static program that has not been put into use. The correctness of the program is a prerequisite for ensuring their security and reliability. The other one is the security issues that may arise during their execution. By considering these two dimensions in a comprehensive way, the security and reliability of developed software components can be greatly improved.

1.3 Objectives and Main Contributions

Fig. 1.1 depicts the positioning of the main contributions of this PhD thesis. It illustrates the problems tackled, the approaches adopted to solve it and the different solutions to meet our goal. In the following, we present the main objectives of this thesis. We address each objective with one contribution.

- State of the Art: We present security issues and requirements in IoT and Industry 4.0 environments. Therefore we comprehensively review existing blockchain applications in such areas. Specifically, we present the current research trends in each of the related industrial sectors, as well as successful commercial implementations. Then, we emphasis on the major security concerns prevalent in the industrial domain with some countermeasures proposed in the literature. Based on this comparison, we have shown that existing solutions leave some areas where some remaining open issues are encountered and a contribution is needed, those areas where contributions are needed are the ones we tackle throughout this thesis, making them our core contribution to the state of the art.

- First Contribution: we proposed a Distributed Resource Management Framework for Industry 4.0 Environments. The proposed framework utilizes blockchain to keep a living document trace about the flow of resources being distributed and shared among collaborating parties while using the OrBAC access control model to implement distributed, fine grained, flexible and secure resource access authorization.

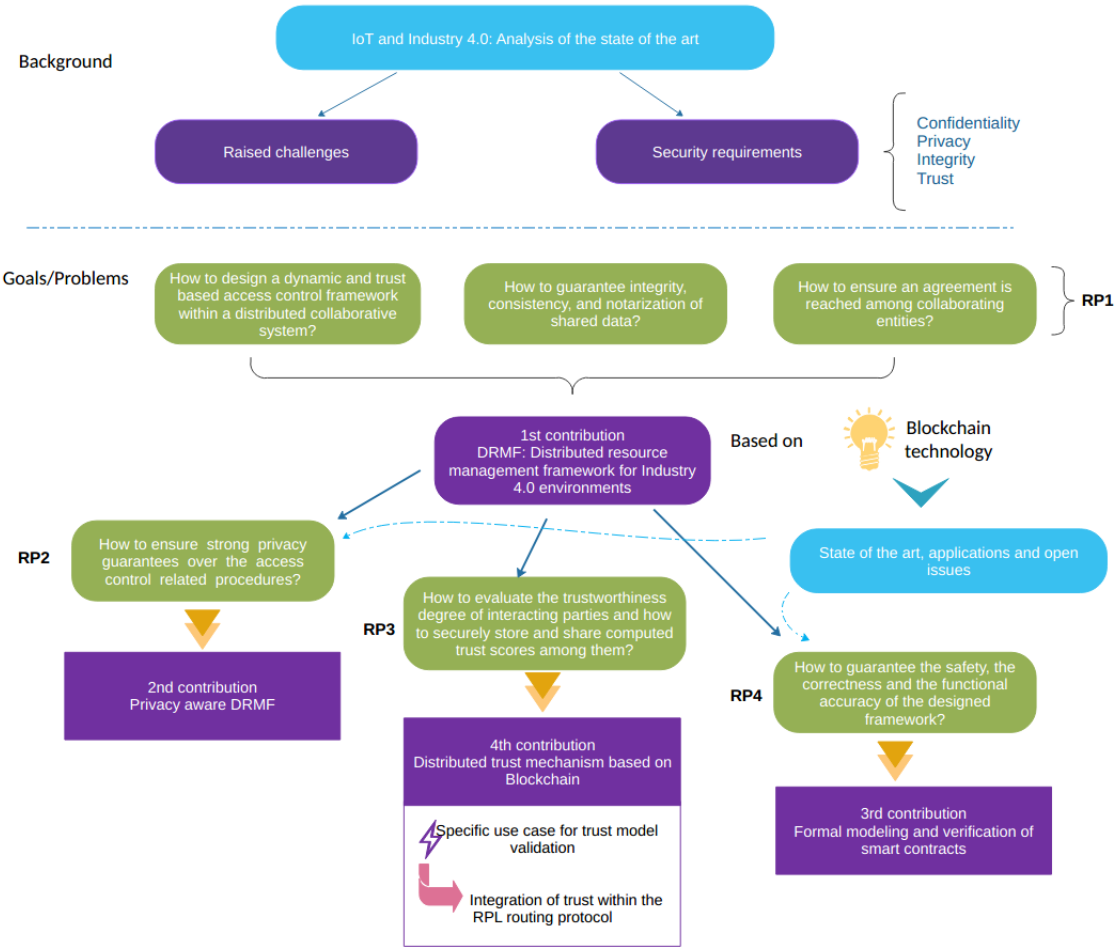

Figure 1.1 – Thesis objectives and contributions

Moreover, and in order to better support the security requirement, this framework adds the notion of trust management to the access control model where a trust framework is integrated to evaluate access requester entities' behavior. Finally our proposal is conceived to support distributed and dynamic governance of the system through the registration of new entities requesting mining permissions. To demonstrate the application of the framework, a case study is provided in which an Ethereum private blockchain network is configured and a set of smart contracts are implemented. However, adopting the blockchain technology to handle shared resources management and controlling the access made over them is not straightforward and additional critical issues are raised regarding the privacy of access requester entities as well as the safety and the reliability of the proposed framework. We deal with such issues in the following contributions.

- Second Contribution: We ensure strong privacy guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies. The proposed scheme will be integrated within our

DRMF framework to preserve the anonymity of both the access requester entities as well as the collaborating parties, by this way the transparency feature of our framework will be maintained while guaranteeing and preserving the privacy of its users.

- Third Contribution: As a third contribution, and as blockchain smart contracts could be exposed to many security problems and attacks caused by vulnerabilities contained within, reasoning about the correctness, the safety and the functional accuracy of smart contracts before their deployment on the blockchain network is a need. In this context model checking tools are well adopted for the formal verification of smart contracts in order to assure their execution as parties' willingness as well as their reliable and secure interaction with users. For this contribution, we use Event-B formal verification method to formally model written smart contracts in order to verify and validate their safety, correctness and functional accuracy in addition to their compliance with their specification for given behaviors. To illustrate the proposed approach, its application to a realistic industrial use case is described.

- Fourth Contribution: We propose a secure trust management system based on the blockchain technology so that we can take advantages of security features it provides regarding reliability, traceability and information integrity. Blockchain based trust management can provide tamper proof data, enable a more reliable trust information integrity verification, and help to enhance its privacy and availability during sharing and storage. For this purpose, we design and implement a blockchain based trust architecture to collect trust evidences, to define a trust score for each device and to securely store and share them with other devices within the network by embedding them into blockchain transactions. Results from performance evaluation demonstrate that our proposal provides security features including tamper-proof and attacks resiliency, reliability in addition to a low complexity for IoT scenarios and applications. For assessing the trustworthiness degree of each entity within the network and deriving a trust score to be shared between involved entites and considered within the decision making process, a generic trust mechanism is defined. For the evaluation and the experimentation of this last, we have chosen to use our solution within a specific use case which is the RPL routing protocol to ensure secure routing by protecting the network against misbehaving, selfish and malicious nodes regarding the routing procedure.

These contributions have given rise to several publications in international conferences and revues namely:

1.4 Manuscript Organization

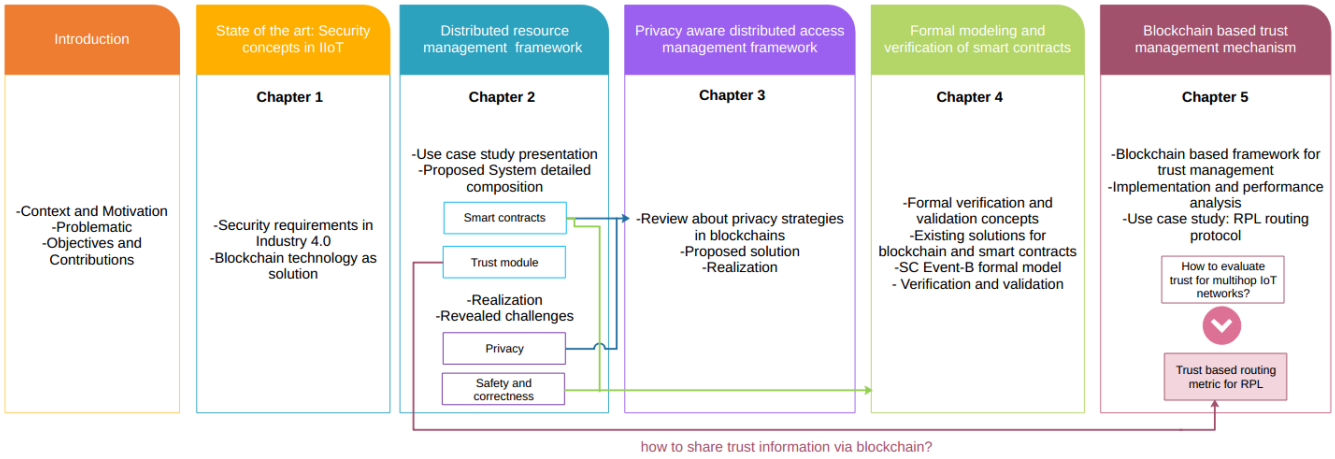

Figure 1.2 – Thesis plan

As it is illustrated in Fig. 1.2, the remaining of this dissertation is organized on five chapters as follows. Chapter 1 provides a comprehensive study of the state of the art of security related concepts in the IoT and Industry 4.0 domains. Chapters 2, 3, 4, and 5 detail our proposed contributions as presented in 1.3

Chapter 2

State of the Art: Security and Trust concepts in Industrial Internet of Things (IIoT)

Contents

Introduction

Internet of Things (IoT) is the interconnection of objects sensing, communicating and interacting with each other on a cooperative basis to meet a standard goal. The integration of such paradigm within the manufacturing environment and processes in combination with other technologies has introduced Industry 4.0 that represents the fourth industrial revolution. In this scenario, security requirements represent a crucial issue whose satisfaction is a key to achieve users acceptance of such technologies. These last include data confidentiality, integrity and authentication, identity management, privacy and trust among the different devices. Besides, data gathered by industrial IoT devices may contain confidential, sensitive and private information, in addition to the emergence of security threats that aim to exploit the weaknesses of current IoT and industrial infrastructures. Indeed, most state of the art IoT infrastructures are heavily centralized with single points of failure, which hinder scalability and wide adoption of based solutions and applications. To improve privacy and security within such environments, a more decentralized approach is seen as the solution to allow the longterm growth of both IoT and IIoT. This refers mainly to the fact that participating entities when interacting and collaborating with each other, do not have to rely and to trust external services or third-parties to manage access rights over data they produce or they share. In this context, several proposals have been made for ensuring distributed systems in IoT environments and hence harvest scalability and security advantages. However, these last cannot easily solve the integrity, immutability, traceability, and notarization issues required for most use cases within such environments until the blockchain technology has emerged as a prominent perspective to develop security solutions in decentralized, collaborative and trustless domains. In recent years, this technology has attracted significant scientific interest in research areas beyond the financial sector, among them IoT and Industry 4.0. This last is seen as the missing link toward building a truly decentralized, traceable, trustless, and secure environment for such domains. In this context, our aim in this chapter is to shape a comprehensive picture of the current state of the art efforts made in this direction. As the domain is vast and various, we first recall the basic concepts of IoT and Industry 4.0. We list then the main challenges related to their appearance, in particular security and trust issues. Therefore we provide a detailed description of what a blockchain is, how a blockchain network operates, what are its main characteristics and concepts, and how blockchain based systems achieve the characteristics of decentralization, security, and auditability. From there, we comprehensively review existing blockchain applications in Industry 4.0. Specifically, we present the current research trends in each of the related industrial sectors, as well as successful commercial implementations. Further, we emphasis on the major security concerns prevalent in the industrial domain with some countermeasures proposed in the literature and finally we present currently open issues raised with the adoption of blockchains in Industry 4.0.

2.1 IoT and Industry 4.0: Background and Basic Concepts

In this section we will point out first the basic principles of Internet of things (IoT) and Industry 4.0. We will overview then the main challenges and issues related to their appearance and we will focus on some important recurring topics that could be integrated within these concepts in order to obtain benefits in specific application scenarios.

2.1.1 Internet of Things

Definition

As an arising technology, IoT is expected to offer promising solutions that will revolutionize not just the conduct but also the services to be provided across several industries such as health-care, transportation, energy and manufacturing. Building upon a complex network connecting billions of devices, objects, services and humans into a multitechnology, multi-protocol and multi-platform infrastructure, this paradigm main vision is to create an intelligent world while bridging the physical world, sensing/actuating, processing, analytics, to the digital, cyber, and virtual worlds on a global scale [63].
In other words, IoT could be defined as the intelligent connection of objects that equipped with sensors, collect data and take decisions locally or collectively. These objects communicate with each other without human interaction, however they just need to have Internet connectivity in order to retrieve and send their data to be kept in a database or even in a Cloud infrastructure for further processing that requires many other networks to be realized, for this reason the IoT can be viewed as a network of networks of things [59].
For example, production processes will be organized and monitored remotely, machines will talk to machines to coordinate their actions function of the information collected by different sensors and exchanged with other entities among the production line in order to control the corresponding value chain.
It is clear here that the fact of exploiting IoT basic technologies including sensor networks, embedded technologies, communication standards and Internet protocols will impact the nature of involved objects, making them capable of communicating and interacting with other external entities while sharing information and coordinating decisions.

Reference Architecture

Although several architectures have been proposed to model the Internet of Things, the basic one is still the well-known three-layer architecture [84, 179] consisting of the application, the network, and the perception layers.

As illustrated in Fig. 2.1, the perception layer is made up of smart devices, actuators

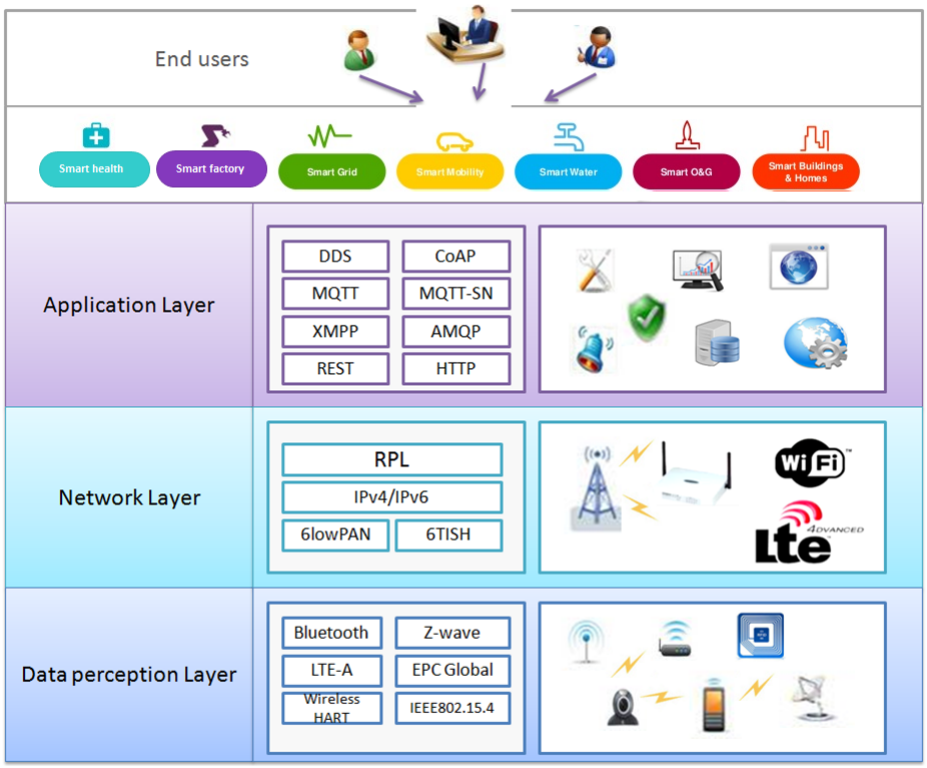

Figure 2.1 – Internet of Things reference architecture

and wireless sensing devices. Its main tasks are perceiving, identifying, collecting information and automatic control. To ensure such functions, several standards and communication protocols were proposed such as IEEE 802.15.4, Bluetooth, LTE-A, WirelessHART [160], ISA100.11a, etc.
As a second layer, the network layer ensures the processing, the addressing, and the information transmission and routing from the perception layer to the application layer safely and reliably through the use of infrastructure protocols such as 6LowPAN [89], 6TiSCH [165], IPv4/IPv6, RPL[178], etc.
Finally, the application layer takes in charge the control and the management of transmitted information, the activation of relative events and the generation of requested services by both customers and end users. To do so, several application and service discovery protocols were proposed such as DDS [132], COAP [31], AMQP [172], MQTT [99], XMPP [143], REST, HTTP, mDNS, DNS-SD.

2.1.2 Industry 4.0

Definition

Currently, the emerging technologies such as Big Data and data analytics, Cloud Computing, machine learning, and Internet of things are being frequently integrated into the manufacturing environments and processes introducing as a consequence thereof the fourth industrial revolution sometimes referred to as Industry 4.0 [62, 150, 32]. This

concept has introduced what has been called the Smart Factory, the revolutionized version of traditional factory in which Cyber Physical Systems (CPS) communicate over the IoT to monitor the physical processes of the factory and to make decentralized decisions.
This revolution includes the introduction of highly flexible and greatly efficient supply chains, manufacturing on demand, logistics operations and production processes, the provision of new services and the allowance of mass-customization and virtual production.
Thus, the core idea of Industry 4.0, similarly for the Smart Factory, is the use and the integration of emerging information technologies within industrial and manufacturing processes what could make production operates in an efficient, flexible and economic manner with constantly high quality and low cost. As a consequence thereof, everything in and around the manufacturing supply chain will be interconnected such as machines, data, processes, suppliers, customers, distributors, even the product itself.
By this way, data about business operations will be shared between involved entities and locations, production lines will be remotely monitored and automatically handled, machines will communicate with each other to organize the production, to adapt their functioning to both operating conditions and received orders , and also to coordinate their actions function of the information collected by the different sensors regarding their location, their status, as well as the encountered faults, exceptions and problems.
As seen, digitalized manufacturing will result in a wide range of changes and benefits to manufacturing processes, outcomes and business models [62].

Reference Architecture

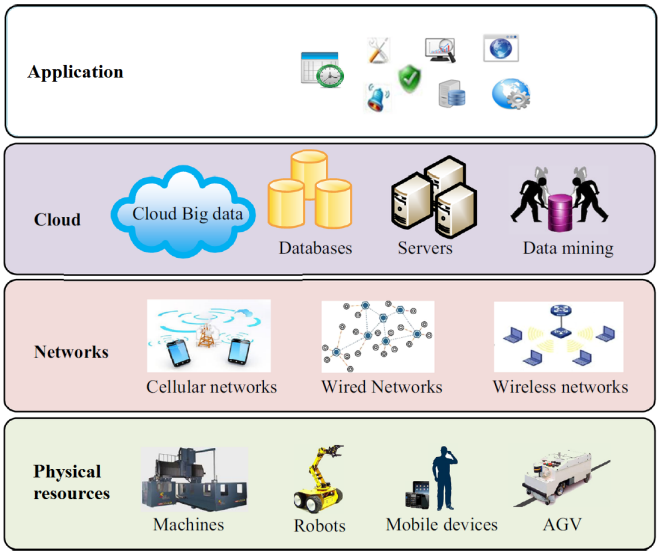

Figure 2.2 – Layout of the Smart Factory of Industry 4.0

Fig. 2.2 illustrates the general layout of a Smart Factory within Industry 4.0. This framework is composed of four main layers [176, 93], including the physical resource layer, network layer, Cloud layer, and application layer.
The physical resources layer comprises smart machines (e.g. robots, mobile devices, workmen, AGVs (Automated Guided Vehicles), etc.), smart products and smart conveyors. These resources acquire and compute data while communicating with each other through the industrial network for the completion of mechanical tasks and the achievement of the system-wide goal.
These resources communicate not only with each other but also with the data servers and the industrial Clouds through the industrial networks. These last are made up of cellular, wireless, wired and other industrial networks transmitting data in realtime among the involved entities.
The Cloud layer is responsible for storage, analytics, mining, computation, high performance processing and so on. Once activated and operated, the physical resources begin to collect and produce huge amounts of information data transferred to the Cloud via the network layer in order to be processed after by application systems. Hence the Cloud is an important infrastructure that provides the bridge between the networked resources and the application layer.
This last links users, workmen, and management to the smart factory systems. Through the terminals they use such as computers, LCD screens, smart phones and tablets, they can access the statistics provided by the Cloud, apply a different configuration and provide key parameters according to their needs by choosing some different options or perform maintenance and diagnosis of the production process, even remotely through the Internet.

Key Technologies

Various technologies or techniques can be used for implementing Industry 4.0. These technologies include Cyber Physical Systems (CPS), IoT, Cloud Computing, Big Data and Data Analytics, Mobile Computing, Industrial Information Integration and other related technologies [101, 181, 173].

- Big Data : The volume, the variety and the velocity of data collected, produced and communicated between devices within industries will be more and more important than ever seen before. Smart factory devices such as wireless sensors, cameras, programmable controllers, digital gauges, RFID readers, bar codes and energy monitoring components (e.g. temperature controllers, hour-meters, counters, air flow meters, etc) plus services, supervision and management applications are assumed to produce continuously and in a single day larger amounts of industrial data than they used to do in a full month just ten years ago. Currently, the produced data has reached a total volume of more than 1000 exabytes annually and is expected to increase 20-fold in the next ten years [15]. Moreover,

the data produced will be diverse, complex, heterogeneous and comes in many varied types and formats such as text, images, audio, video, scans, digital and analog data and much more. That's why the comprehensive evaluation, the efficient data analysis and the real-time actionable insight and intelligence gaining shall be guaranteed within the industrial production processes. Meanwhile, the 5 Vs of Big Data namely Variety, Volume, Velocity, Value and Veracity have introduced several issues especially adequate hardware and software requirements data processing, the urgent need for on-line detection ability, and the necessity for interdisciplinary techniques [185].
An entire generation of innovative new technologies has sprung up to address these issues. The utilization of Big Data is now feasible with the Cloud Computing that enables to effectively manage the scalability, the performance, the manageability, the reliability and the computing processing of such a system. Many others techniques and technologies can be used such as : Genetic algorithms, Machine learning, Regression analysis, Sentiment analysis and Social network analysis.

- Cloud Computing : According to the National Institute of Standard and Technologies, Cloud Computing can be defined as "a model for enabling ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". [96] While the concept of Cloud Computing has existed for several years and has gradually evolved to reach various IT environments, the technology has only started to take its place in the manufacturing industry as a solution to store and manage the explosive growing volume of production and control related data. Cloud based manufacturing can contribute significantly to the realisation of Industry 4.0 that enables modular and service oriented manufacturing where systems orchestration and sharing of services and components are important considerations [164].
  Besides, a smart industry related operations involve numerous decision making processes that require large amounts of information plus intensive computation. On the one side, and while these kind of industries require several computing and storage resources such as servers for databases and decision-making units, they have suffered before from an inefficient data exchange and sharing, low productivity and less optimal utilisation of manufacturing resources. The integration of Cloud Computing within such environments would provide undoubtedly an effective solution to such problems. All data can be stored and processed in private or public Cloud servers. By this way complex decision-making tasks can be supported and manufacturers can effectively increase the agility, ensure efficient processing, ameliorate performance and drive profitability.[182, 137]

When talking about agility, manufacturers with enabling Cloud Computing, will be able to adapt to changing and fluctuating market demands by modulating their production capacity, realigning their processes, products and equipments and supporting automatic scaling of resources and processing capabilities. This could consequently increase productivity and reduce relatively operating costs especially when using a private Cloud solution.

- Cyber Physical Systems (CPS): Cyber Physical Systems are defined according to [101] as "industrial automation systems that integrate innovative functionalities through networking to enable connection of the operations of the physical reality with computing and communication infrastructures". These last are considered as the core foundation of Industry 4.0 that

2.1.3 Research challenges

The integration of IoT technologies within the industrial environments makes production processes and operations operate in an efficient, intelligent and flexible manner with constantly high quality and low cost. However, such integration will create also various challenges that have gained increasing attention from the public and the research area [84] [161] [116] [21]. In the following, we will cover five main challenges coming from both IoT and IIoT unconventional characteristics namely scalability, heterogeneity, reliability, dynamic changes, and security as presented in Fig. 2.3. We deal specifically with such challenges just as they are the closest to our work alongside this thesis.

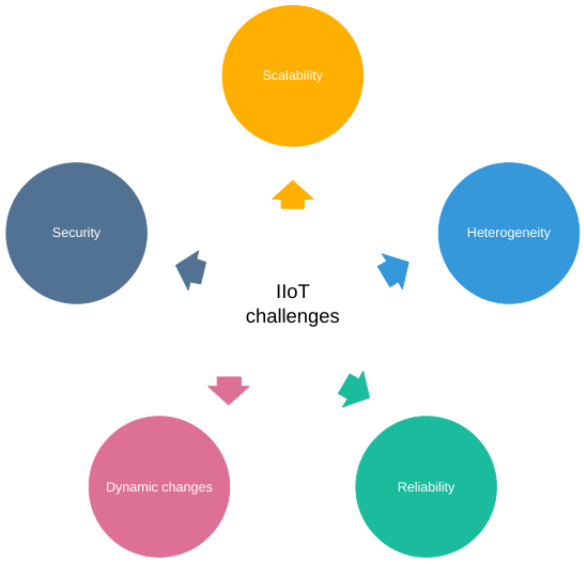

Figure 2.3 – Industrial IoT raised challenges

Scalability

With the explosive growth of Internet connected devices, both IoT and IIoT based applications and services must be able to support the increasing number of connected objects, end users as well as the application features, processing and analytics capabilities without causing any significant decrease nor degradation in the quality of the service offered to their customers. Scalability is therefore essential to meet the inherent features of such systems. In such vast networks of interconnected objects, designing related frameworks such as authentication, authorization and access control mechanisms should take into full consideration the scalability feature of such environments so that all participating and involved entities from organizations and humans to devices, assets and services should be identified and authenticated to grant access and authorization tokens to entities requesting to use their resources at anytime and from anywhere. These mechanisms therefore should be extensible in size, structure, and number of users and resources. Besides, the unbounded number of connected entities exposes them to potential threats and attacks that imposes to move towards distributed approaches and infrastructures without centralized control of any security authority or management system.

Heterogeneity

The IoT interacts with a large number of devices presenting very different technologies, services and capabilities from the computational and communication standpoints thus making them incompatible. Differences between those devices can be the operating system, the connectivity, the I/O channels and the performance which will lead certainly to different computational power, storage capacity and energy consumption. Since IoT devices would be connected through an interface in common in order to communicate all together, the management of their heterogeneity should be guaranteed at both architectural and protocol levels [18]. Thus the task of standardization needs to be considered to ensure interoperability among devices and also to standardize the communication among the network.

Reliability

The reliability within the Smart factory is an important evaluation factor that reflects the performance of the whole system insofar as it evaluates both data and results consistency as well as the stability of the offered services.
In an industrial environment, the reliability is concerned with how much data is

received successfully at the receiver end with minimum delay. However, the reliability of the transmitted data is affected by the environment dynamic topology where packets transmission is susceptible to link availability, interference, channel state change and protocol overheads. Therefore, high communication reliability is essential to provide accurate and precise supervision of industrial processes.
In this context, authors in [93] have presented some of the approaches used to increase the reliability of wireless sensor networks used in industries such as redundancy, frequency-hopping and interference minimization. In addition, many other works [171, 15, 101] have assumed that the use of the fifth generation (5G) mobile technology will address effectively the industrial requirements associated with Industry 4.0 based Smart factory by reducing the communication latency, increasing the longevity of devices battery life and more importantly improving the reliability of communication in indoors as well as in outdoors.

Dynamic changes in industrial environments

In the context of IoT, states often describe devices' behaviors. Transitions between states are quite common and frequent, e.g., started and standby, sleeping and waking up, leaving and joining networks. Besides, the number of connected devices can also evolve. Environments in which IoT devices operate are subject to contextual changes. The characteristic of dynamic changes is the intrinsic properties of the IoT. However, many threats emerge due to dynamic changes in IoT systems. For instance, in intelligent transportation systems with characteristics of the high mobility of connected vehicles, rapidly changing network topology and unbounded network size, hackers could even hijack a moving car and take the control. Particularly, IoT devices, such as vehicles or wearable devices equipped with strong mobility, often make great demands on across domain authentication and authorization to prevent malicious attacks from adversaries. Therefore, the secure IoT infrastructure should be able to resilient to this dynamic changes environment and provide a peer-to-peer authentication and authorization services.

Security

Many Other challenges have been discussed in the context of IoT but one of the most important ones is the security challenge [116, 88], in fact traditional security mechanisms could not be used directly within IoT applications due to its different technologies, standards and communication stacks [8]. In addition, the existence of such a large network with a high number of interconnected entities will definitely imply various scenarios of attacks and eavesdropping which could threaten those entities and put them in danger thus harming the corresponding users.

To cope with this challenge, cyber security systems must offer adapted mechanisms to protect the collected data from physical devices and this since it may include and manage sensitive user information. This means that data confidentiality, integrity, and availability should be provided by the IoT system [159] which could be done by considering encryption primitives [121, 184], redundancy techniques as well as authentication [90], access control and authorization [107, 72] mechanisms in order to prevent unauthorized users to access the system.
Recently, many other security challenges have been arised especially with the full increasing implications of ubiquitous connectivity and on the other hand as regards to the frequent integration of IoT services and applications to carry out daily activities.For example, data providers can behave deceitfully by providing false information. Users personal and sensitive data could be collected, accessed and interpreted by third parties what could make data providers hesitant about sharing their information. Hence both privacy and trust represent real and major issues that may limit the potential and the development of IoT applications. The increasing amount of production data uploaded and shared between smart devices deployed in heterogeneous and distributed architectures and communicating with each other independently within and beyond the factory site put the corresponding industrial system at a greater cyber risk that need to be seriously considered in the near future. For example, attackers can manipulate and infiltrate industrial systems, malware injection can disturb their functioning and put them out of action, which could cause significant damage to the whole production area.

2.2 Security requirements for Industry 4.0 based Infrastructures

In addition to the previously mentioned challenges, we still have to sort out specific security requirements in order to build secure infrastructures and frameworks for Industrial IoT based environments. Some researchers have already introduced many works related to security requirements in IoT. For instance, in the context of IoT, [159] focuses more on security and privacy requirements in terms of confidentiality, authentication, access control, and trust. Moreover, in [108], have identified security challenges in each layer of IoT and potential attacks like replay, DoS, man-in-the-middle, and eavesdropping attacks. Similarly, in [9] authors have provided a taxonomy of security threats and attacks in IoT that were classified in terms of application, architecture, communication, and data. The presented security vulnerabilities are then discussed for hardware, network, and application components and demonstrated in different application scenarios such as smart environment, healthcare, etc. More specifically, [183] presented a comprehensive

survey that focused on the specific issue of trust management in IoT. In the context of Industrial IoT and Industry 4.0 and with the growing trend of integrating IoT in industries and manufacturing sectors, a significant number of surveys have been realized to highlight research advancements done in the domain of Industrial IoT. However just few ones have discussed security concerns and predominant attacks prevalent in the industrial domain. For example, in [155] authors have surveyed IIoT specific security problems and discussed eventual corresponding solutions. In anoher work [134], authors have outlined basic requirements for secure decentralised industrial IoT infrastructures and therefore examined threats and vulnerabilities targeting such systems.
From the already mentioned research papers, we can deduce that security requirements in both IoT and IIoT should not only cover inherent characteristics of related systems and environments but should also deal with several components and layers ranging from devices and networks to services and applications during both the design phase and the runtime phase. In this context, this section states the main security requirements for a secure Industrial IoT infrastructure.

2.2.1 Confidentiality

Confidentiality may be defined as the guarantee that data is protected from being accessed, retrieved, cracked, altered and misused by unauthorized and malicious parties. This security requirement represents a fundamental issue in IoT and IIoT scenarios whereby not only users but also authorized objects, machines and services may access data. This last may represent an asset or a resource to be protected in order to maintain the competitiveness, preserve the privacy of related holders and safeguard the market values [116]. This requires addressing two important aspects: first, the definition of an access control mechanism and second, the definition of an object authentication process (with a related identity management system). However in such environments, current solutions for access control and identity management may not be applicable due to several limitations such as the amount of data generated, the dynamic change of such networks, the agreement needed to be established in case of shared resources.

2.2.2 Integrity

Integrity may be defined as the guarantee that data is protected from unauthorized alteration and tampering. These measures provide assurance in the accuracy and the completeness of data either collected and stored at the device layer, or transmitted and exchanged between entities at the network layer. For maintaining integrity, several countermeasures may be put in place []. For instance, access

control and rigorous authentication can help in preventing authorized users from making unauthorized changes. Hash verification and digital signatures on the other side can help in ensuring transactions authentication and guaranteeing that files have not been modified or corrupted.

2.2.3 Privacy

With the increasing use and efficiency of data processing, information privacy has become the predominant concern nowadays, that's why it is considered as one of the major issues that need to be addressed in IoT based applications and specifically when designing and developing trust management systems.
Although privacy is a very broad and diverse notion for which the literature offers many definitions and perspectives [138] but in the context of IoT and IIoT, little attention have been paid to privacy preserving issues and techniques. In fact, many IoT solutions and services provide sensitive and personal information about users and data providers such as personal preferences, private information, opinions, interests, etc. This information could be later revealed and made known to all the accessing users, it could even be accessed and misused by a third party. That's why privacy preserving is so important to protect data providers' personal information such as identity information, location, mobility traces, habits from any other parties in order to keep a certain degree of anonymity, unlinkability and data secrecy.

2.2.4 Availability

Availability may be defined as the guarantee that all services and devices are timely and uninterruptedly accessible by authorized users. Some of the most fundamental threats to availability are non malicious in nature and include hardware failures, unscheduled software downtime and network bandwidth issues. Malicious threats include various forms of sabotage intended to cause harm to an organization by denying users access to the information system.
In the context of IoT, availability requirements, as specified by [140], are highly tied to reliability requirements. IoT systems need to display sufficient resiliency to sustain availability under desired levels as well as they need to guarantee a certain level of performance requested by their applications.

2.2.5 Trust

In the current literature, various trust definitions have been proposed. these last range from specific scenarios to wide and general systems. Meanwhile, the trust

meaning across existing proposals differs from one work to another insofar that each one of them has considered trust from different perspectives. According to [120], trust is defined as the subjective expectation of others future behavior as expected by their evaluators on the basis of the history of their encounters. Another definition was given in [69] where trust was defined as the firm belief that other entities are competent enough to act in a secure, dependent, and reliable way within a specified context.
A trust management system is often needed to produce reaction based on the real time evaluation of entities behaviors during established interactions in addition to feedbacks and recommendations gathered from other entities. These last aggregated together form an overall trust score that once shared and propagated over the network, participating entities could decide whether to continue or not the collaboration.
This concept plays a key role in IoT in general and has a great importance for industrial IoT based environments. It is essential when participating devices, without being previously interacted with each other, want to cooperate and to use provided services with a certain degree of trust among themselves. It is needed also to achieve trustworthy data during collection, exchange, analysis, fusion and mining phases which is inevitably crucial in IoT and especially in smart factories where devices continuously collect data with great amounts and important information within that is needed for critical decision making. It is needed as well for many other decision making situations such as access control, intrusion detection, key management, isolating misbehaving nodes for effective routing, authenticating devices before interaction and other purposes.
Regarding this set of security requirements, in this thesis, we will focus more on trust mechanisms. For this purpose, we will detail more this requirement in Section 2.3.

2.3 Trust management in IoT

We have presented in the previous paragraph the existing trust definitions, in the following we will give a generic definition for trust in the context of IoT. Then, we will describe trust properties, the operations considered within the trust management process, and therefore we will review how proposed works have dealt with the concept of trust management within each operational block in IoT systems.
In the context of IoT, trust could be defined as a relationship established between two entities: a trustor and a trustee, where the trustor is the evaluating entity while the trustee is the evaluated one. The trustor when needing to collaborate with the trustee, to use the services or the information it provides, when it receives outcomes through the established communication while coming in touch to it, will

evaluate its competence to act just as predicted for a specific period and within a specified context. Thus this relationship is always related to a time value which corresponds to the trust evaluation time, a context where the relationship resides in, and a parameter on which the evaluation depends.
As a part of this thesis focuses on the concept of trust management, we will present in this section more details about the notion of trust, and we will review how proposed works have dealt with this last during the trust composition process in IoT systems and exceptionally in smart industries.

2.3.1 Trust properties

Due to the different requirements of IoT applications and regarding the nature of the harsh industrial environment, the concept of trust can be viewed in a different way and accordingly its evaluation can depend on several properties and characteristics. In this section, we will define the main trust properties as illustrated in Fig. 2.4.

Figure 2.4 – Trust properties

Trust is unidirectional

When evaluating the trustworthiness of a specific entity, the trustor generally rely on the evidence it has about the trustee. This evidence may be acquired either through its direct observations, either through the policies it specifies, either from the recommendations it receives from the neighboring nodes, or other means.
Obviously, the trustee may not necessarily know the trustor or has established previous interaction with him and therefore it may not trust him. Even in the case where previous interactions have been existed between the two entities, the disposition of the trustee, its willingness to trust, its perception on the trustor's

performance and benevolence may differ.
Hence trust may not be mutual or reciprocal . This property could be formalized as follow:

$$\exists(e_i, e_j) : trust(e_i \longmapsto e_j)_t^c \neq trust(e_j \longmapsto e_i)_t^c$$

Trust may not be transitive

[79, 42] If the trustor e_i needs to evaluate the trustworthiness of e_j to collaborate with for the first time, e_k which e_i trusts, comes and recommends e_j, in this case should e_i consider the received recommendation and trust e_j?
The answer may be yes and no. This fact is so large to be as simple as this. In fact many other factors come into play to determine whether trust could be transitive or not such as the context in which the trustor e_i trusts the recommender e_k, the community to which each entity belongs and the relationships that exists between them.
This property could be formalized as follow:

$$trust(e_i \longmapsto e_k)_t^c \wedge trust(e_k \longmapsto e_j)_t^c \implies trust(e_i \longmapsto e_j)_t^c$$

Trust is dynamic

Present entities in IoT environments are generally resource constrained and mobile in nature and due to these two characteristics, the network topology changes at every instant and consequently data in transmit will be typically incomplete and can change rapidly. To accommodate the different changes related to the environment in question, various approaches were considered in the literature.
Some have considered event driven scheme [124, 41, 142], where trust is updated after a communication or an event is made between two entities or after ratings and feedbacks are sent to the evaluating entity.
Others have considered time-driven scheme [36, 39, 38], as trust evidence is collected periodically, trust needs to be updated at each period, past evidence can be considered also while updating trust either by applying aggregation techniques or by using exponential time decay function.

Trust is social based

The exploitation and the integration of social concepts within IoT environments has introduced SIoT, the Social Internet of Things paradigm [**bib322**, 17]. This last is mainly related to the fact that objects within IoT environments are able to establish social relationships in an autonomous way with regard to their owners.

The evaluation and the computation of trust therefore will depend not only on the trustee's competence and capability to perform the requested task, but also on its commitment and on the type of the relationship towards the trustor.

Trust is subjective

The evaluation of trustworthiness depends not only on the behavior of the trustee, its performance, reputation or technical properties but also on how this evidence is perceived and interpreted by the evaluating entity. Hence trust can be influenced by the subjective properties of both the evaluating and the evaluated entities [183] such as the evaluated entity's honesty, benevolence, goodness and on the other side, the evaluating's confidence, expectation, belief and willingness to trust, etc.

2.3.2 Trust operational blocks

The design, the implementation and the development of a trust management model generally goes through a succession of operations that are considered essential for trust computation in dynamic networks [67].
In Fig.2.5 we present the possible trust operational blocks that need to be implemented within a trust framework for IoT. These phases include: (i) the computation of a trust value based on a specific parameters, considering some metrics and using certain factors, (ii) the computed values are propagated over the network in order to establish trust between entities having neither prior knowledge nor previous interaction, (iii) As trust values are sent by several evaluating entities and since they are propagated through multiple paths, they need to get aggregated into one single value which will be used in a later trust compositions, (iv) Whenever a change occurs due to the network dynamicity, trust could be predicted potentially using the present and past trust values, (v) The trust values will be applied into the network in order to achieve the desired purposes.

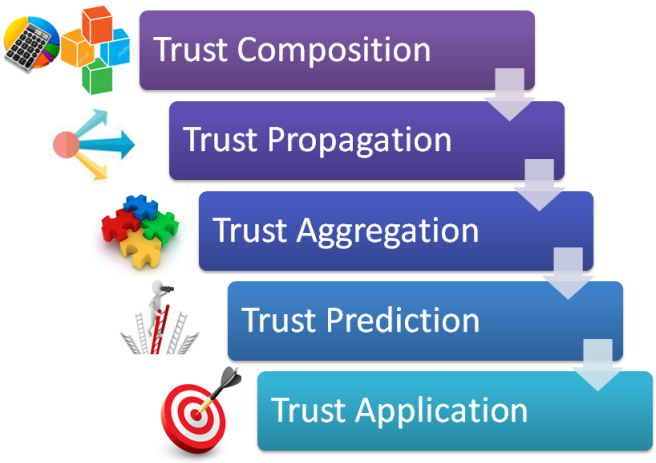

Figure 2.5 – Trust operational blocks

These five phases the trust composition process is made up of are detailed in the following paragraphs.

Trust Composition Before producing a trust score and judging an entity either trustworthy or not, the trust management system should collect enough information about a specific entity in order to define its trust score regarding its behavior as it will be considered by other network entities.
The questions to be asked here:

- What approaches to use to determine trust?
- Which kind of aspects to consider?
- How to compute the gathered information and to provide a final trust value?

The process to be realized within this block is illustrated in Fig 2.6

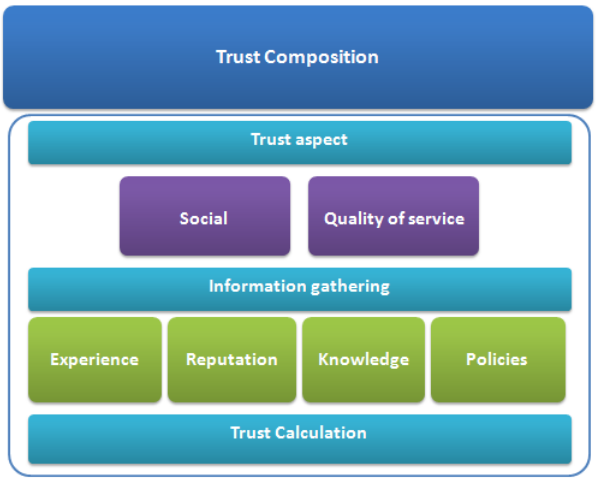

Figure 2.6 – Trust composition process

As seen, the first step within trust composition is to determine and to specify which kind of aspect the information gathering will focus on. Here we can distinguish between social trust and technical trust (or QoS trust) [71]. Social trust derives from established social relationships among IoT entities while quality of service trust is determined by the belief on entities ability to guarantee a certain quality of service in response to a service request. When reviewing the current literature, we can interpret that almost all trust management systems have considered both technical (QoS) and social trust as an aspect on which the information gathering focuses for the trust evaluation phase. This fact is mainly related to the introduction of the social Internet of Things [62, 150] according to which participating entities are capable of setting up social relationships that are generally affected by those existing between their corresponding owners. Among social properties, friendship [40, 136, 125] and community of interest [22, 23, 24, 40], are the most used ones for the evaluation of social trust. This refers to the fact that having a good friendship, sharing a common interest or belonging to the same community implies a good confidence on the provided service and the established communication. For the technical trust, we have seen that this last considers several parameters such as the energy consumption level [37, 177] and the device related capability [142, 136, 177, 2] in order to address respectively the energy efficiency and the resource constraints challenges present in almost IoT based environments. As a next step and once the aspect is determined, the information gathering process can be launched. This last may be based on several approaches such as experience, reputation, knowledge and policies which represents the trust parameters [14, 68, 169]. The experience parameter corresponds to each node's interpretation of the previous interactions and events established with its immediate neighbors at a specific period of time. These evaluations will be kept within each node and updated at regular periods and regarding regular events. Moreover, they will be propagated as trust recommendation part to other network nodes. Subsequently, the past gathered trust information will be regularly kept and considered after as the knowledge part of trust. Another approach to evaluate trust is to use well-defined languages and semantics as policies to make trust decisions. For the information gathering process, we have seen that some works in the current literature have considered the context while assessing the trustworthiness degree of each entity within the network [142, 136, 177]. The context awareness is an important feature that should be considered while designing a trust management system insofar that some nodes could act honestly in such a context and maliciously in other one.

Finally a last step within the trust composition process is the trust calculation where a trust value will be computed and modeled according to a specific method such as probability, mean, difference, etc.

Trust Propagation In Fig. 2.7, we illustrate the functional sub-blocks corresponding to the trust propagation process. In fact, once trust is composed regarding a specific entity, the trust value will be propagated to the network entities what would effectively optimize the resource utilization as entities will spend no more resources to recompute trust.
The main questions here are:

- How to propagate trust values?
- Shall an entity propagate trust autonomously and periodically to other entities within its neighborhood?
- Shall it wait until it receives a request from another entity to propagate trust?
- Or shall it send it to a centralized entity for further processing and storage?

Generally, there are three schemes of trust propagation namely centralized, distributed, and decentralized. where in the distributed scheme, each entity will record locally the trust information and provide it either on request from relying nodes or communicate it autonomously to other entities it interacts and collaborates with. On the other hand, in the centralized scheme, a third party (i.e. a centralized server, a cloud platform, a virtual service, etc.), came into play to propagate the trust information over the network in order to make it publicly available. As it is seen here, the process of trust propagation could be launched either on demand from the relying entities or autonomously, freely and independently by the evaluating ones.

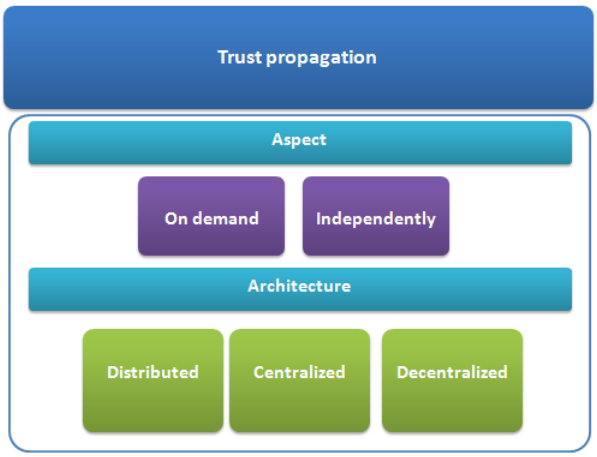

Figure 2.7 – Trust propagation process

After reviewing the existing works, we have seen that existing proposals mostly consider the distributed scheme for trust scores propagation [22, 23, 24, 40]. These

last are sent either upon demand to nodes requesting their use or autonomously and independently to serve within the trustworthiness evaluation process of target nodes. The consideration of such a scheme seems to be the most convenient to the inherent requirements of IoT systems where devices scattered over a wide area could have always access to a centralized system to send and get trust information regarding nodes to be assessed. Moreover no work far discusses how to ensure the integrity, the validity and the authenticity of the transmitted data during the propagation phase. Instead, existing works assume that collecting information from a large number of entities and executing aggregation operations on the exchanged trust related information will result in a relatively accurate assessment.

Trust Aggregation Generally speaking, when trust values regarding a specific entity are requested, different evaluating entities will send their assessed values which will be obviously propagated through multiple paths within the network, thus different trust values will be received and multiple versions of each value will be created as well. For this reason an aggregation technique is needed to combine the received value into a final one.
The main question to be asked here is:

- Which technique to use for trust aggregation?

In the current literature, many aggregation techniques have been presented. According to [79], these last include belief theory, weighted sum, fuzzy logic, regression analysis and Bayesian inference.

Trust Prediction Whenever a change occurs due to the network high dynamicity, trust could be predicted potentially using the entities' present and past trust values. Moreover, when two entities lose contact and there is no edge between them, it will be so important to be able to estimate the trust relationship nature to be established before it took place in the reality.
The main questions to be raised here are:

- Which approach to consider for the trust prediction?
- How to guarantee the accuracy of trust prediction?
- How to trust the prediction in itself and take actions based on it?

Various approaches have been considered in the literature, some of them used the kalman-filter approach to predict the trust system future state [35, 27], others have focused on the unsupervised methods to predict trust [163], while several ones have used the principles of machine learning [127] and data mining [86] as a basis for creating a prediction enabled trust model.

Trust Application Applications of trust management are enormous in mobile networks and particularly in IoT environments where trust management systems are often designed to handle effectively many security services as illustrated in Fig 2.8. These services include : intrusion detection, key Management, secure routing, malicious nodes detection, quality of information assessment, access control management etc.

Figure 2.8 – Trust design purposes and applications

2.4 Research Statement

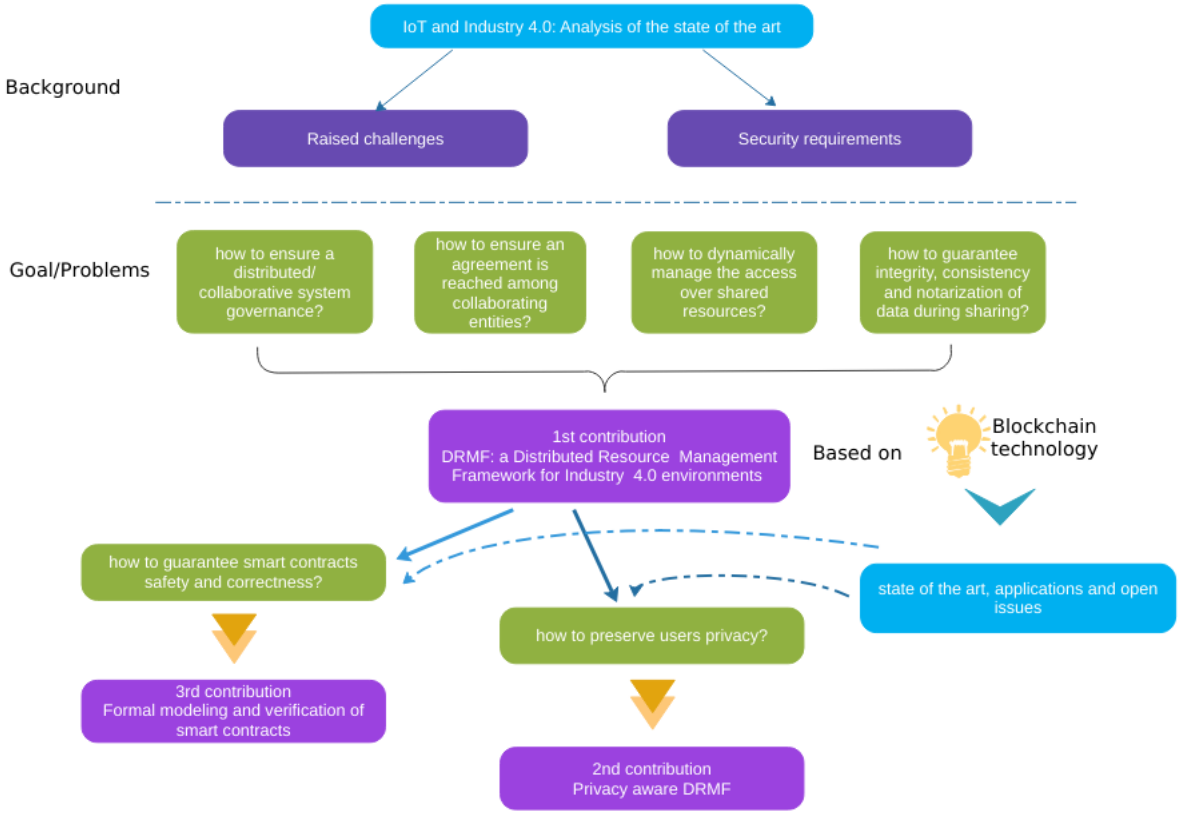

Figure 2.9 – Thesis objectives and contributions

From summarized IoT and IIoT challenges and security requirements, we can see, that in order to build our IoT secure infrastructure for meeting presented security requirements while taking into account mentioned IoT unconventional characteristics, several questions need to be considered. Therefore, we summarize our research problems as follow:

- How to ensure a distributed collaborative system?
  we remind here that, the aim of IoT is mainly to have smart objects communicate over the Internet to collect comprehensive data and provide personalized automation services, with little deliberate human interaction. Towards this aim, current IoT platforms are built on a centralized model where a central server or broker provides services like data handling, device coordination, and authorization. This approach necessitates high-end servers and proves to be unsuitable for scenarios where objects are required to autonomously exchange data and where end-to-end communications do not have to go through a centralized server for performing automation services.
- How to ensure an agreement is reached among collaborating entities? When talking about entities collaborating and working altogether alongside a production process for example and sharing for this purpose a set of common resources, an agreement should be reached among them in order to ensure a distributed and dynamic governance and management of the overall system where all relevant parts can make a comprehensive decision of joining entities registration and shared resources management without the need for a third party that could be itself vulnerable to attacks.
- How to dynamically manage the access over shared resources?
  Where end users and collaborating parties are more empowered to control access over their own devices as well as over resources they share by defining their own policies and dynamically reconfiguring them in response to time, events and more importantly to entities' changing behavior and attitudes.
- How to guarantee integrity, consistency and notarization of data during sharing?
  We have seen in the previous section how security is one of the most critical concerns to IoT applications. Particularly, the integrity and consistency of collected and shared data is the basic guarantee for securing established operations. In this direction, effective mechanisms need to be considered to protect IoT communications for confidentiality, integrity, authentication and non-repudiation of information flows. Moreover, interacting entities need to

be identified to ensure the data integrity from the origin, which conventionally relies on trusted third parties that may lead to breaches of privacy and security in the considered environment.

2.4.1 The solution: Decentralizing IoT and IIoT networks in a secure way through the blockchain technology

A decentralized, resilient, fault tolerant, secure and censorship resistant approach to both IoT and IIoT networking would solve many of the encountered issues as described above. The concept of a distributed scenario is not novel. In fact, many academic and industrial studies consider it as one of the most promising approaches that can push the dream of the IoT into the real world [78]. Currently, the blockchain technology represents one of the most suitable candidate technologies able to support a secure and distributed ecosystem for both the IoT and the Industrial IoT. The inherent features of such technology make it a natural fit to developing distributed and secure frameworks for these environments. That's why we propose to integrate it within our work in order to take advantages of security features it provides. Fig. 2.9 depicts the positioning of the main contributions of this thesis. The figure illustrates the problem tackled, the approaches adopted to solve it and the different solutions to meet our goal. As a first contribution and as illustrated in Fig. 2.9, we propose to use the blockchain technology within a resource management framework for ensuring reliability, traceability, control, information integrity and notarization. The proposed framework utilizes blockchain to keep a living document trace about the flow of resources being distributed and shared among collaborating parties while maintaining distributed system governance and implementing dynamic, fine grained, flexible and secure resource access authorization. More details about the proposed approach are given in the next chapter, Chapter 3. To better understand the core feature of the blockchain technology as well as its working principles, we will investigate in the next section its key challenges and benefits. The state of the art of blockchain technologies in terms of consensus mechanisms, enabling technologies and main concepts are analyzed as well.

2.5 Blockchain in Industry 4.0

From existing IoT and Industry 4.0 based solutions, distributed models could not only improve the scalability of the considered system but also to ensure the security and the privacy of involved entities compared with centralized and decentralized models. This refers mainly to the fact that participating entities when

interacting or collaborating with each oher, do not have to rely and to trust external services or third-parties to handle data they produce and share given the fact that these last can misuse it or in worst case scenarios, share it with mass-surveillance programs. In this context, several proposals have been made for ensuring distributed systems in IoT environments and hence harvest scalability and security advantages. For instance, Webinos [64] and Alljoyn [12] are two pioneers for practicing distributed models in IoT environments. However, these last cannot easily solve the integrity, immutability, traceability, and notarization issues required for most use cases within such environments until the blockchain technology has emerged as a prominent perspective to develop security solutions in decentralized, collaborative and trustless environments. By using blockchains, we could remove intermediaries, allow users and devices to manage their data without relying on third parties and especially we can reach high levels of harmony in the traceability and sharing of information. By providing effective sharing of historical information, this technology guarantees transactions transparency and traceability.
In the area of IoT applications such as smart manufacturing, tracing historical data is crucial. For instance, by reviewing data, we can identify key factors that might impact product quality. By improving processes, a higher quality will then be achieved. By filtering through the data, we can discover weak spots in production and so on.
We will provide in this part a detailed description of what a blockchain is, how a blockchain network operates, what are its main characteristics and concepts, how smart contracts allow us to radically redefine how interactions between transacting parties on a network can be set up and automated, what are the different application scenarios a blockchain is used for in IoT and Industry 4.0, and finally what are the main security challenges these applications face once the blockchain technology deployed within.

2.5.1 Definition

Originally designed for keeping a financial ledger and meeting the purpose of cryptocurrency applications, the blockchain paradigm can be extended to provide a generalized framework for managing any movements of data related to goods, devices, information records, etc. This last could be defined as a distributed ledger of transactions across a decentralized network whereby records of all established interactions are registered providing thereof a proof of existence, of ownership and modification of this data during interaction [43, 48, 50].
These transactions are hold within blocks chained together through cryptographic hashes contained within their headers in order to ensure immutability insofar that

blocks once chained, data contained within will be available and could never be changed or altered. Each block references the hash of the block that came before it. This establishes a link between the different blocks, thus creating a chain of blocks, or blockchain as it is illustrated in Fig. 2.10. Any node with access to this ordered, back-linked list of blocks can read it and figure out what is the world state of the data that is being exchanged on the network.

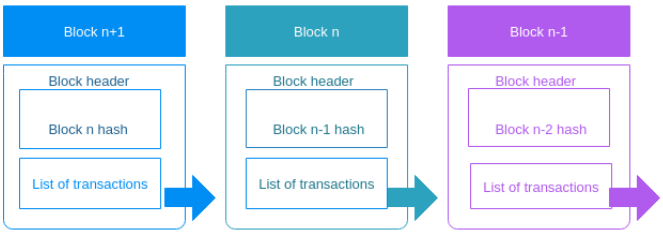

Figure 2.10 – Blockchain design structure

2.5.2 Types of Blockchain

Based on several criteria and how they are used in different application scenarios, blockchain systems can be classified into three main types namely public, private, and consortium blockchains [10, 43]. These three are compared in Table 2.1 and described as follows:

- Public Blockchains: This category provides an open platform that allows users from different organizations and backgrounds to join, transact, mine and perform read and write operations on the blockchain. There are no restrictions on any of these factors and anyone can send transactions, maintain a copy of the distributed ledger and engage in validating and adding new blocks to the chain where the nomination of permissionless blockchains.
  Moreover the blockchain is open and transparent, there are no specific validator pre-selected set of nodes and all users can publish new blocks to the blockchain by solving either computationally expensive puzzles, or staking one's own cryptocurrency. The availability of the copy of the entire blockchain, synchronized with all of the nodes, makes it secure and immutable.
  Moreover, this kind of blockchain is tamper-resistant insofar that each transaction has a processing fee attached to it, which prevents the public ledger from being hacked since it would be too costly to tamper its contents.

- Private Blockchains: This kind of blockchain is mainly set up to facilitate the private sharing and exchange of data among a group of known members

within a single organization. Private blockchains are also called permissioned blockchains insofar that external users cannot have access nor participate unless they are authorized to do. Users' participation is decided either by a set of rules or by the network that controls access. This inclines the network more toward centralization, while derogating the elementary blockchain features of complete decentralization and openness as defined. In a private blockchain system, once nodes become part of the network, they contribute in running a decentralized network, with each node maintaining a copy of the ledger and collaborating to reach a consensus for updating. But, unlike public blockchain, the writes are restricted.

– Consortium Blockchains: This kind of blockchain could be considered as a partly private and permissioned blockchain, in which no single organization handle consensus process and block validation but rather a set of preselected set of nodes. These nodes decide who can participate in the network and who can partake in the consensus mechanism. Thus, it is a partially centralized system, owing to the control by some selected validator nodes. Similar to private blockchains, this kind of blockchain does not involve processing fees, and it is not computationally expensive to publish new blocks [10]. While it does provide auditability and lower latency in transaction processing, it does not entirely guarantee immutability and irreversibility, since the control of the consortium is made by a majority of nodes which can lead to tampering with the blockchain.

	Public blockchain	**Consortium blockchain**	**Private blockchain**
Registration authorities	Anyone	Multiple entities (organizations)	Single entity Defined before initializing the network
Access	Public read/write	Can be restricted	Can be restricted
Identity	Pseudo-anonymous	Approved participants	Approved participants
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Participation in consensus	All nodes	Selected nodes in multiple organizations	Single organization
Transaction Speed	Slow	Lighter and faster	Lighter and faster

Table 2.1 – Types of blockchains

2.5.3 Blockchain Technology Features and Working Principles

In order to understand the potential applications of blockchains in IoT and Industry 4.0, it is important to have a clear understanding of the main concepts and working principles of blockchains. In the following, we introduce the main features and working principles involved in achieving immutability, security, and integrity of blockchain stored data.

Consensus Mechanisms

To ensure that all entities agree on the transactions and the order in which these are listed on the newly validated block so that they have the same copy of the ledger, an agreement is required to maintain the blockchain architecture and to ensure its functioning and consistence. Otherwise, the individual copies of the blockchain will diverge and we will end up with forks. In such a case participating nodes will have a different view of the world state and the network will no longer be able to maintain a unique authoritative chronology unless this fork is resolved. A distributed consensus mechanism is therefore needed in every blockchain network in order to make sure that an agreement is reached between the set of predefined

entities to support a decision making. The type of the considered consensus mechanism depends mainly on the blockchain network as well as the characteristics and the capabilities of the participating entities.
To reach consensus among validating entities, several ways could be considered [190, 20]. Below, a brief introduction to a few of them is given.

- PoW (Proof of Work): this consensus strategy is used in the Bitcoin network [122] in addition to many other cryptocurrencies to confirm transactions and produce new blocks to the chain. In such a strategy, Publishing new blocks to the blockchain is called "mining", and miners engage in a race to find a nonce that, when hashed with the hash of a block, produces a resultant smaller than a predefined threshold. In the decentralised network, all participants 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. After that, transactions in the new block would be validated 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. Subsequently, the miner claims the processing fees associated with the transactions stored within the block as a reward for mining. In PoW consensus, the computationally expensive block creation and transaction fees secure the network against DDoS attacks and false block creation.

- PoS (Proof of Stake): this consensus strategy is an alternative approach for PoW that requires less CPU computations for mining. Instead of a competition between participating nodes to solve the next block, a node is chosen for mining task based on its proportional stake in the network which is its wealth in terms of that cryptocurrency. This last will usethen a digital signature to prove its ownership over the stake instead of solving a complicated hash problem. Meanwhile, the more currency forgers held, the greater chance they have to generate the next block. Moreover, in this method, all coins are available from the first day and no mining reward or coin creation exists, instead, miner nodes are rewarded only with the transaction fees.
  Although this method eliminates the computational requirements of proof of work, it creates new problems, insofar that is contingent upon nodes with the highest amount of stake which somehow makes the blockchain centralized.

- PBFT (Practical byzantine fault tolerance): this consensus strategy is based on a replication algorithm to tolerate byzantine failures. All entities, in this method, should participate in the voting process in order to validate and to

add the next block. Here, the consensus is reached when more than two-thirds of all nodes agree upon that block. Meanwhile, PBFT can tolerate malicious behavior from up to one-third of all nodes to perform normally. For instance, in a system with one malicious node, there should be at least 4 nodes to reach a correct consensus. Otherwise, consensus is not reached.
Besides, in such mechanism, the consensus is reached quicker and in an economic manner compared to proof of work. Also, it does not require owning assets similar to proof of stake to take part in the consensus process, which make it well suited to private blockchain networks like Hyperledger projects that are controlled by a third-party.
However, it is not the best choice for permissionless, public blockchains due to their limited scalability and comparatively low tolerance towards malicious activities. PBFT has high throughput, low latency, and low computational overhead, all of which are desirable for IoT networks. However, its high network overhead makes it unscalable for large networks, thus it could be applied only to small IoT networks.

– RR (Round Robin): This mechanism is mostly used in private blockchain networks, where mining is restricted only to select identifiable entities. This prevents the problem that such blockchains might have, which is the monopolization of the mining process. More specifically within this consensus strategy permitted entities create blocks in rotation in order to generate a valid blockchain, and every entity in a given time window can only create a finite number of blocks calculated using a network parameter called mining diversity that determines the number of blocks that should be wait for before attempting to mine again.

Smart Contracts

A smart contract is an executable code deployed and residing at a specific unique address on the blockchain network. This last is triggered by addressing a transaction to it. The main aim of a smart contract is to automatically execute the terms of an agreement once specified conditions are met. It include a set of data which are the state variables and code corresponding to the executable functions. These last are executed when transactions are made, broadcast to the network and addressed to its address. Called smart contract then runs independently and automatically in a prescribed manner on every node in the network, according to the data that was included as input in the related transaction, as a result an eventual return value is shown to the outside.
When a blockchain supports smart contracts, it does allow for multi-step processes to occur between mutually distrustful counterparties. The transacting entities get

to: (a) inspect the code and identify its outcomes before deciding to engage with the contract, (b) have certainty of execution since the code is already deployed on a network that neither of them controls fully, and (c) have verifiability over the process since all the interactions are digitally signed. The possibility of a dispute is eliminated since the participants cannot disagree over the final outcome of this verifiable process they engaged in.
Smart contracts operate as autonomous actors, whose behavior is completely predictable. As such they can be trusted to drive forward any on-chain logic that can be expressed as a function of on-chain data inputs, provided that the data they need to manage is within their own reach.
Smart contracts can be developed and deployed in different blockchain platforms where each one of them offers distinctive features for development supported by different high-level programming languages. In Ethereum blockchain platform, advanced and customized smart contracts are supported with the help of Turing complete programming language. The code of an Ethereum contract is in a low-level, stack-based bytecode language referred to as Ethereum virtual machine (EVM) code. Users define contracts using high-level programming languages compiled into EVM code. The most widespread language is Solidity [53] which is a JavaScript style contract-oriented, statically-typed, high-level programming language designed for implementing smart contracts.

Peer To Peer Networks (P2P)

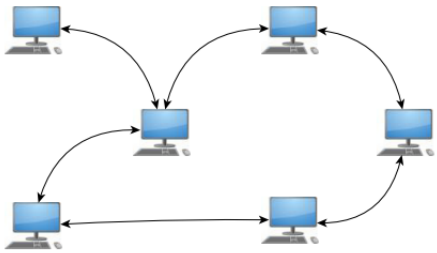

Figure 2.11 – Peer to peer model

The continuous and widespread growth of Internet based applications in terms of number of users and computational resources, has challenged the centralised nature of the client-server paradigm which has led to the emergence of peer to peer networks. These last provide a good substrate for creating large-scale data sharing, content distribution and application-level multicast applications [102]. Fig. **??** illustrates the general model of a peer to peer network. As shown in Fig **??**, there is no hierarchical organization or central governing authority within the network.

Instead a set of autonomous entities called peers form self-organizing overlay networks that are overlayed on the IP networks, offering a set of various properties as follows:

1. Self-organisation: where nodes are able to interact with each other without any central control. This is the main property of peer to peer systems that makes the difference against the client-server paradigm.
2. Decentralised resource usage: available resources of nodes (CPU, storage, and bandwidth) are distributed and shared with the best effort regarding its load distribution.
3. Fault tolerance: where the failure of a single node within the peer to peer network must not compromise the correct operation of the whole system.

When it comes to blockchains, one of the main goals of this technology is to minimize the number of intermediaries being involved in the process. As a result, the use of the peer to peer architecture was a necessity to offer security, decentralization, and censorship resistance.

Cryptographic Techniques

In blockchain, cryptography technology is mainly used to protect user privacy, transaction information, ensure data consistency and immutability, and guarantee the blockchain as a distributed ledger with tamper proof and public verifiability [174]. In what follows, we will present an overview of the commonly and widely used cryptographic primitives and algorithms in blockchain platforms.

- Hash functions: A hash function is a computational method that maps data of arbitrary and indeterminate size to a fixed size string and is characterized by susceptibility, unidirectionality, non-invertibility, collision resistance, and high sensitivity [187]. Hash functions are generally used to guarantee data integrity and immutability and to ensure that it has not been illegally tampered with. In the context of blockchains, hash functions can be used to perform block and transaction integrity verification where the hash value of the information of the previous block is stored in the header of each block, and any user can compare the calculated hash value with the stored hash value. In turn, the integrity of the information of the previous block is detected. The most popular hash function used in blockchains is SHA256, which is one of the algorithms from a family of cryptographic hash functions named SHA (Secure Hash Algorithms).

- Digital Signatures: Besides hash functions, digital signatures are another inevitable cryptographic primitive in blockchains. Generally speaking, these primitives are used for ensuring source authentication, source non-repudiation and integrity. A digital signature scheme usually consists of two algorithms: a signature generation algorithm and a verification one. The generated signature is mainly dependant on a signature key that is kept secret, and controlled by the signer. For the verification phase, a public key is used for verifying the received signature. In the context of blockchains, ECDSA and EdDSA are the two digital signature schemes frequently used in blockchains [174]. In principle, both of them are based on the hardness of the elliptic curve version of discrete logarithm problem. ‘
- Other primitives: To enhance more transactions and identites privacy, some other primitives are applied in some blockchain based applications such as ring signatures, multi-signatures, non interactive zero knowledge proof, commitment proof and so on [60, 75].

Blockchain key characteristics

In this paragraph, we will briefly delineate the main features and key carachteristics of blockchain that will make it an attractive technology for addressing the aforementioned issues encountered with IoT and Industry 4.0 based applications.

- **Decentralization:** Blockchain technology utilizes the idea of distributed and decentralized computing and storage. There is no centralized database in a blockchain and participating can engage in transactions with each other without the need to place trust upon a central third party to maintain records of data exchange or perform authorization. This eliminates the many to one traffic flows and overcome the problem of having a single point of failure.
- **Immutability:** Since all new entries made in the blockchain are agreed upon by peers via decentralized consensus, the blockchain is censorship-resistant and is nearly impossible to tamper. Similarly, all previously held records in the blockchain are also immutable and, in order to alter any previous records, an attacker would need to compromise a majority of the nodes involved in the blockchain network. Otherwise, any changes in the blockchain contents are easily detected.
- **Traceability:** Blockchain reaches high levels of harmony in the traceability and sharing of information. Providing effective sharing of historical information, this technology guarantees transactions transparency and traceability. In the area of IoT applications such as smart manufacturing, tracing historical data is crucial. For instance, by reviewing data, we can identify key

factors that might impact product quality. By improving processes, a higher quality will then be achieved. By filtering through the data, we can discover weak spots in production and so on.

- **Auditability:** All participating peers hold a copy of the distributed ledger, consequently they can access all timestamped transaction records. This transparency allows peers to look up and verify transactions involving specific blockchain addresses. Blockchain addresses are not associated with identities in real life, so the blockchain provides a manner of pseudo-anonymity. While records of a blockchain address cannot be traced back to the owner, specific blockchain addresses can indeed be held accountable, and inferences can be made on the transactions a specific blockchain address engages in.

- **Security:** Security is comprised mainly of confidentiality, integrity, and availability. Blockchain uses hash functions to chain blocks which ensures integrity and immutability insofar that blocks once connected, data contained within cannot be subsequently altered. For the availability requirement, this last is fulfilled inherently in blockchain given its distributed nature and data is always available. Coming to the confidentiality requirement, this last is achieved in permissioned blockchain solutions were users are permitted to access just the data they are authorized to access through the use of permissions. Without forgetting the fact of encrypting transactions before being linked to the existing ledger.

2.5.4 Blockchain for IoT and IIoT: Applications

Integrating Blockchain technologies into Industry 4.0 based applications, such as those involving Industrial IoT (IIoT) and IoT has proved to be quite beneficial [115]. For instance, several attempts have been made for harnessing the power of blockchain technology to face security issues in Industrial systems, to facilitate data collection and storage techniques, and to trace the flow of data being exchanged and accessed. The adoption of such technology has empowered several IIoT applications to enhance their security, transparency and to add additional features within their pre-existing systems. In this section, we present the current research trends in Industry 4.0 related sectors specifically the manufacturing, the supplychain and the healthcare sectors and we discus how the most relevant applications have used blockchain to add decentralization, and to enforce security, safety, privacy and other benefits to their systems.

Manufacturing Industry

Within a smart factory environment, manufacturing data alongside the production process have to be shared between different related departments and across complex networks of machines, products, human operators, and value chain participants that may be deployed in heterogeneous, distributed and unknown sites. These data have to be released, timely updated, and securely accessed and shared among the different parties within and outside the factory walls which is a very tedious process and requires a certain level of trust to be established in advance between the different parties. Security hence is one of the most important concerns in such environments, given that vulnerabilities introduced during manufacturing can be hard to detect and harder to react especially with open connectivity to the external world in addition to the increasing amount of shared data. For example, attackers can manipulate and infiltrate industrial systems, malware injection can disturb their functioning and put them out of action, which could cause significant damage to the whole production area. Also, users with different motivations, roles and in different contexts can utilize the terminals to interact with data storage systems. They may query data or attempt to take control of some physical resources (i.e. terminals and the communication channel are the target). Or an attacker can seek to collect available data through harvesting and exfiltration of sensitive information from these terminals.
By integrating the blockchain technology, immutable records of manufacturing data can be stored and uploaded on the shared distributed ledger and is accessible to blockchain participants while guaranteeing transparency, auditability and traceability of both the flow of data as well as their access history.
Some existing applications and use cases of blockchain in the manufacturing industry are summarized below.

1. Logistics management: One important area is using the blockchain technology for logistics management in order to ensure fair pricing and guarantee timely delivery of row material and supplies for its productions. In addition, it helps to ensure efficient and on time delivery of their products to meet their customers' need. In [3], authors have proposed a decentralized distributed system that uses the blockchain technology to collect, store and manage key product information of each individual product throughout its life cycle what would create a secure, shared record of transactions for each individual product along with specific product information.

2. Social manufacturing networks: A second area is using the blockchain technology to enable social manufacturing networks among manufacturing parties who collaborate altogether alongside the production process in order to

effectively and securely support sharing of their manufacturing resources. Forming a social manufacturing network can efficiently enhance manufacturers' competitive capabilities and help them to produce more precise and professional services and products that meet their customers' needs. However, there are always high concerns of security, fairness, and effectiveness in such collaborative network. In this context, authors in [97] developed a blockchain based product credit assurance mechanism under social manufacturing context in order to securely, fairly, and effectively manage and regulate the cross-enterprise collaborations among socialized manufacturing resources. This management is achieved in a peer-to-peer fashion, without involving a third party and using both smart contracts plus a credit system.

3. Cloud manufacturing: Another direction is utilizing the blockchain technology to support cloud manufacturing operations. This last is a new service-oriented manufacturing model designed to utilize concepts from cloud computing, Internet of Things, service-oriented computing, and virtualization in order to transfer the manufacturing resources, capabilities and operations into a set of services that can be smartly integrated and managed according to the customers' demands [26]. The integration of the blockchain technology within such a paradigm has encountered problems it faces especially those regarding centralization, scalibility and resiliency to vulnerabilities which affect the system such as cyber-attacks and security problems. In this context, authors in [26] has introduced a decentralized cloud manufacturing system based on the peer to peer network developed via the blockchain technology in order to enable all evolving set of parties to maintain a safe, permanent, and tamper-proof ledger of connections without a central authority. Subsequently, this concept was expended in [25] where authors have focused on machine level connection, and shop floor data sharing based on blockchain technology.

4. Security and privacy: A last but not least direction is using the blockchain technology for enforcing security and privacy requirements within industrial systems. In this direction, several works have been made in an attempt to use the blockchain technology as a basis for designing access control systems, identity management tools and preserving the privacy of their participants as well. For instance in [129], FairAccess, a blockchain based access control framework is presented for IoT. This last makes use of bitcoin stack and introduces a new type of transactions to manage access permissions. Access requests are granted or denied through defining access control transactions using smart contracts, which takes advantage of the immutability of the blockchain and makes the auditing of access control policies transparent.

In [95] a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0 was presented. The proposed framework leverages the underpinning characteristics of blockchain as well as several cryptographic materials to realize a decentralized, privacy preserving solution. Another approach was presented in [77] where authors have proposed a credit-based proof-of-work (PoW) mechanism for IIoT devices. The proposed mechanism is based on directed acyclic graph (DAG)-structured blockchains which is further used to develop a data authority management method. This method regulates access to sensor data thereby protecting sensitive data confidentiality as well as data privacy.
On the other hand and with regards to blockchain based identity management systems, these last are gaining much attention to propose new solutions for digital identities. In this context, several efforts have been made in order to develop blockchain based identity systems such as Uport [103], Shocard [158], Civic [149], Sovrin [166], etc. For instance, Uport, which is a core component of the Consensys Ethereum ecosystem [192], aims at building decentralized applications to solve the digital identity problem. It mainly uses smart contracts to design digital identity models and ensures reliability and usability of identities through a set of operations (i.e., keys revocation and identities recovery). For its part, Sovrin takes a different approach and provides a complete full stack to manage identities from the distributed ledger to devices. It adds the identity layer for every entity on the Internet and operates as a global public utility designed to provide permanent, private and trustworthy identities. Sovrin establishes a public permissioned blockchain in a peer-to-peer network in which nodes are divided into authenticated validator nodes (permissioned) and observer nodes to ensure high performance and scalability.

Supply Chain Industry

Traditional supply chain applications focus generally on the management of the delivery of raw material, products, and services between providers, customers and consumer destinations. This can run across multiple organizations and entities or just be part of a single organization. The integration of the blockchain technology within such applications can provide powerful support for their well conduct. Reminding that within such applications, it is so important to provide a set of functions for planning, scheduling, coordinating, monitoring, tracing and validating the performed related operations. Which can be efficiently, securely and transparently supported by the blockchain technology. Using a shared distributed ledger to verify,permanently store and audit logistics transactions can also reduce

time delays, management costs, and help to check the authenticity of the items by tracing their origin. In addition, applying smart contracts will facilitate agreements between the companies involved and create binding contracts faster and with lower costs.
In this context, authors in [19] have described how a reliable, transparent, authentic and secure system can be created by integrating blockchain into the supply chain architecture. They have also studied the benefits of introducing the blockchain to the supply chain and the challenges to be encountered. Another approach was presented in [118] where authors have proposed a blockchain based food supply chain using the proof-of-object (PoO) based authentication protocol. For food traceability, an RFID tag is attached to the food product in order to track its location throughout its lifetime within the supply chain. Tracking and monitoring generated data are then stored within the distributed ledger thus assuring better quality of the food to be exchanged. Moreover in [100], involved companies in the supply chain are connected to the blockchain to share logistic related data with different visibility levels while simultaneously verifying its authenticity and integrity. For the architecture deployment, a simulation model is used to recreate such a supply chain and integrate it within the Ethereum blockchain.
Furthermore and regarding commercial implementations, important blockchain based commercial projects in the domain of supply chain management are discussed below:

1. Food Trust: This project created by the IBM company [30] is a modular solution build on blockchain technology for food traceability, supply chain transparency and auditability. The project is using the IBM blockchain platform in addition to Hyperledger Fabric Project architecture to track food production flows. The tracking information in this project includes storage temperatures, expiration date, shipping details, origination farm details, batch number and much more relevant data when the food being delivered worldwide. The data are digitally connected to food items and the information is entered into the blokchain along with every step of the process.

2. IBM Blockchain-TradeLens: IBM Blockchain provides solutions that cover all aspects of supply chain management, with a specific focus on logistics. Transparency and traceability are the most critical aspects of logistics, and IBM Blockchain can streamline business exchanges, transactions and trading associations with secure, worldwide business systems and networks. With solutions like TradeLens, new and open blockchain-fueled platforms built to support worldwide trade major logistics players are profiting by a common and shared ledger that is validated promptly with each system member

[151]. The outcomes are streamlined stock administration, greater collaboration, improved resource usage and much more.

3. OriginTrail: The main objective of this project is to bring transparency to complex international supply chains since 2013 [109]. It is a platform, which is already in use in the food industry, that lets its users know the whereabouts of their food products.

4. BlockVerify: It is a blockchain based anti-counterfeit solution presenting transparency in the supply chains. It is effectively being utilized in diamonds, pharmaceuticals, and a couple of electronic industries.

We summarize in Table 2.2 the main benefits of using the blockchain technology in each of the discussed industrial sectors. Therefore in Table **??** we present a summary of the different case studies reviewed in this Section giving details of the blockchain technology employed in each work. For each case study, the problem addressed and the solution proposed are discussed. Technical details of blockchain such as consensus algorithm used, the blockchain platform used, transaction data which goes into each block, and the status of smart contracts usage have also been discussed.

Industrial Sector	Benefits of utilizing the Blockchain technology in each industrial domain
Manufacturing Industry	– Reduce manufacturing costs by improving manufacturing supply chain management. – Enable social manufacturing networks. – Support Cloud manufacturing operations. – Enforce security and privacy requirements. – Enhance anti-counterfeiting and copyright protection.
Supply Chain Industry	– Reduce time delays, human errors and management costs. – Help to check the authenticity of items by tracing their origin. – Support logistic operations scheduling, coordinating and monitoring. – Enable agreements to be established between involved parties. – Make a transparent and effective framework for dealing with all archives engaged within the logistic procedure.

Table 2.2 – Benefits of utilizing the blockchain technology in Industry 4.0 related sectors

2.5.5 Security Discussions on Blockchain based IoT and Industry 4.0 Applications

Recently, the concept of industrial IoT is slowly gaining popularity with its practical application in various domains and application areas. This has lead to the discovery of various security vulnerabilities and threats whitin the industrial environment such as replay, Man in the middle and DDoS attacks [156]. Along with these security risks, blockchains bring their own set of security vulnerabilities. Although these last are considered as one of the most promising technologies for securing both IoT and Industrial IoT applications, they may suffer from several

vulnerabilities and security threats. In this section we will present first the different vulnerabilities of blockchain systems, open research challenges are then highlighted and discussed.

Security vulnerabilities of blockchain systems

Double spending :
Although the consensus mechanism of blockchain can validate transactions, it is still impossible to avoid double spending. Double spending refers to that a consumer uses the same cryptocurrency more than once for validating transactions. For example, an attacker could leverage race attack for double spending. The existence of such kind of attacks in Bitcoin were first analysed in [82] where authors showed how, with some reasonable assumptions and without the need of special computation nor much network overhead, an attacker has a great probability of succeeding with a double-spending attack. Moreover, authors showed also how basic countermeasures such as waiting a few seconds before accepting the payment or adding observers that report back to the payee are not enough on their own to avoid these types of attacks. Consequently, proposition to modify the Bitcoin protocol rules was made so that nodes forward double-spending transactions instead of dropping them.

51% vulnerability :
The blockchain relies on the distributed consensus mechanism to establish mutual trust. However, the consensus mechanism itself has 51% vulnerability, which can be exploited by attackers to control the entire blockchain. More precisely, in PoW based blockchains, if a single miner's hashing power accounts for more than 50% of the total hashing power of the entire blockchain, then the 51% attack may be launched. Hence, the mining power concentrating in a few mining pools may result in the fears of an inadvertent situation.. In PoS based blockchains, 51% attack may also occur if the number of coins owned by a single miner is more than 50% of the total blockchain. By launching the 51% attack, an attacker can arbitrarily manipulate and modify the blockchain information. Specifically, an attacker can exploit this vulnerability to conduct the following attacks [54]:

- Reverse transactions, and launch double spending attack.
- Exclude transactions, and change their ordering.
- Tamper with the blockchain information.
- Hamper normal mining operations of other miners.
- Impede the confirmation operation of normal transactions.

Private key leakage :
When using blockchain, the user's associated private key is considered to be his identity and security credential, which is generated and maintained by the user instead of third-party services. In [113] a vulnerability in ECDSA (Elliptic Curve Digital Signature Algorithm) scheme which is used to generate the private key of the user is discovered. If the private key is generated with limited randomness, it can be recovered by an attacker. Once the user's private key is lost, it will not be able to be recovered. If the private key is stolen by criminals, the user's blockchain account will face the risk of being tampered by others. Since the blockchain is not dependent on any centralized third party trusted institutions, if the user's private key is stolen, it is difficult to track the criminal's behaviors and recover the modified blockchain information. As a result, many attacks can be launched by exploiting such vulnerability such as: key attack, replay attack, tampering attack, impersonation attack, modification attack, and man-in-the-middle attack.

Vulerable smart contracts :
While smart contracts are becoming widely recognized as the most successful application of the blockchain technology that could be applied into various industries and for different purposes, their implementation has posed several challenges insofar that they could handle large amount of money and digital assets in addition to their ability to manipulate critical data and transactions related information which makes them attractive targets of security threats and attacks that could lead to significant problems like money losses, privacy leakage and data breach. Besides, smart contracts may have several security vulnerabilities caused by program defects. In [16], an investigation of smart contracts vulnerabilities is conducted. These last include 12 types of bugs that was classified into 3 categories regarding the target in which the vulnerability arise namely Source code vulnerabilities, EVM bytecode vulnerabilities and Blockchain mechanism caused vulnerabilities. Another investigation was made in [104] where a symbolic execution tool called Oyente is proposed in order to find potential security bugs. In this work, authors have confirmed that among 19366 Ethereum smart contracts, 8833 are tested vulnerable. Detected bugs of smart contracts include timestamp dependence, transaction ordering dependence, mishandled exceptions and reentrancy vulnerability. The existence of such vulnerabilities made smart contracts an easy target to security threats and attacks aiming at tampering and stealing assets they handle. An example of attack was in June 2016, the DAO (the world's largest crowdfunding project deployed on the Ethereum) was attacked by hackers, causing more than 3 million ETH separated from the DAO resources pool which is worth around $60 million.

Transaction Privacy Leakage :
In the current blockchain environments, the issue is not only that data is permanently stored on a ledger, never to be erased or altered, but that by nature it exists on a blockchain which is irreversibly shared with the entire network what makes it easily accessible by each participating user in case of public blockchain or by authorized ones in case of private blockchain solutions which increases privacy concerns regarding exchanged transactions content. Unfortunately, privacy protection measures in blockchain are not very robust. In [119], authors have empirically evaluated two linkability weaknesses in Monero's mixin sampling strategy, and discovered that 66.09% of all transactions do not contain any mixins which will lead to the privacy leakage of both its sender and its content.

Security and privacy enhancement solutions for blockchain systems

In the following, we will give an overview of some of the practical achievements for enhancing the security and the privacy of blockchain systems.

Hawk :
As it was described in the previous paragraph, privacy leakage is a serious threat to blockchain systems where not only transactions but also smart contracts related data are visible and accessile to blockchain participants, such as contract's bytecode, invoking parameters, etc. For preserving smart contracts privacy, authors in [87] have proposed Hawk, a novel framework for developing privacy preserving smart contracts. Leveraging Hawk, developers can write private smart contracts, and it is not necessary for them to use any code encryption or obfuscation techniques. Furthermore, the financial transaction's information will not be explicitly stored in blockchain. When programmers develop Hawk contract, the contract can be divided into two parts: private portion, and public portion. The private data and financial function related codes can be written into the private portion, and codes that do not involve private information can be written into the public portion. The Hawk contract is compiled into three pieces. (1) The program that will be executed in all virtual machines of nodes, just like smart contracts in Ethereum. (2) The program that will only be executed by the users of smart contracts. (3) The program that will be executed by the manager, which is a special trustworthy party in Hawk. The Hawk manager is executed in Intel SGX enclave, and it can see the privacy information of the contract but will not disclose it. Hawk can not only protect privacy against the public, but also protect the privacy between different Hawk contracts. If the manager aborts the protocol of Hawk, it will be automatically financially penalized, and the users will gain compensation. Overall, Hawk can largely protect the privacy of users when they are using blockchains.

Oyente :
In [104], authors have proposed Oyente to detect bugs in Ethereum smart contracts. Currently open source for public use, Oyente leverages symbolic execution to analyze the bytecode of smart contracts and it follows the execution model of EVM. Since Ethereum stores the bytecode of smart contracts in its distributed ledger, this tool can be used to detect bugs in deployed contracts. For the execution process, this tool takes both the smart contract's bytecode and the Ethereum global state as inputs. Firstly, based on the bytecode, the Control Flow Graph (CFG) BUILDER will statically build the corresponding CFG of the smart contractin question. Then, and according to the Ethereum state and CFG information, EXPLORER conducts simulated execution of smart contract leveraging static symbolic execution. In this process, CFG will be further enriched and improved because some jump targets are not constants. Instead, they should be computed during symbolic execution. The CORE ANALYSIS module uses the related analysis algorithms to detect four different vulnerabilities as described earlier. Therefore the VALIDATOR module validates the detected vulnerabilities and vulnerable paths. Confirmed vulnerability and CFG information will finally be output to the VISUALIZER module, which can be employed by users to carry out debugging and program analysis.

Town Crier :
To enable smart contracts' interaction with off-chain data source, authors in [188] have proposed Town Crier (TC) which is an authenticated data feed system for this data interaction process. Since smart contracts deployed in blockchain cannot access network directly, they cannot get data through HTTPS. TC exactly acts as a bridge between HTTPS enabled data source and smart contracts. The basic architecture of TC is mainly composed of 3 layers namely: TC contract, TC server and TC target which are HTTPS-enabled websites. The TC contract is the front end of the TC system, which acts as API between users' contracts and TC server. The main function of the TC server is to obtain the data requests from users' contracts, and obtain the data from target HTTPS-enabled websites. Finally, the TC server will return a datagram to the users' contracts in the form of digitally signed blockchain messages. TC can largely protect the security of the data requesting process. The core modules of TC are respectively running on decentralized Ethereum, SGX-enabled enclave, and HTTPS-enabled website. Furthermore, the enclave disables the function of network connection to maximize its security. Relay module is designed as a network communication hub for smart contracts, SGX enclave environment, and data source websites. Therefore, it achieves isolation between network communication and the execution of TC's core program. Even if the Relay module is attacked, or the network communication packets are

tampered, it will not change the normal function of TC. TC system provides a robust security model for the smart contracts' off-chain data interaction, and it has already been launched online as a public service.

Fair Access :
As presented earlier, FairAccess is a multi-layered framework that focuses on privacy, reliability and integrity in its design as a blockchain enabled IoT architecture [129]. This framework uses smart contracts on which IoT users rely to selectively associate role based privileges to people requesting access to their data, in exchange for monetary or service incentives. Moreover, Fairaccess has transaction definitions for granting and revoking access to users' IoT data, for decentralized access control. For storage, FairAccess adds a separate storage layer where data is stored in off-chain, decentralized storage systems.

Open research issues

From the analysis of the selected literature, a series of insights can be derived concerning the limitations of the blockchain technology in IIoT and Industry 4.0 based environments. In the following, we will discuss some of the pressing security related open issues in the adoption of blockchain in Industry 4.0 scenarios.
The first issue to speak about is the privacy and the confidentiality of shared data. In this direction, several anonymisation and encryption-based mechanisms can be adopted to protect established transactions content as well as the real identities of their partakers [60, 75]. However, such primitives depend generally on the implementation, the context and the environment of the system in question. A two well-known problems in blockchain data privacy are transaction privacy and identity privacy. To preserve such privacy, several efforts have been made. These last focused mainly on the following aspects:

- Obfuscating transactions relationships to resist linking or tracing analysis.
- Hiding real identities of both the sender and the receiver via complicated cryptographic primitives.
- Blinding the transaction content whilst retaining the verifiability and computability.

Adopted strategies for preserving identities and transactions privacy can be adapted to different requirements of a range of application areas such as financial applications, e-voting systems and healthcare services. However, just few studies have focused on using privacy preservation approaches within blockchain based systems in the context of smart factories environments.

A second issue is related to smart contracts that similarly to programs, frequently contain errors, which can cause hefty losses. To better deal with such issue, reasoning about the correctness, the safety and the functional accuracy of smart contracts before their deployment on the blockchain network is critical and no important than ever. How to write reliable smart contracts was presented in [98], where two aspects were considered for the correctness verification and security insurance of smart contracts including programming correctness and formal verification.
Formal verification provides a powerful technology for the correctness verification of the established specification of smart contracts. To strictly verify the required security of smart contracts, we need to formalize the semantics as well as the security properties of interest, especially at the level of the executed bytecode. A number of security properties should be formally defined for smart contracts, including atomicity of invocation, independence, and integrity.
Another important issue is related to data ownership and protection policies insofar that attackers and malicious users can modify the ownership details to make their ownership invalid. So, preventing the ownership data from unauthorized access is of utmost importance to make the system safe and secure.

Returning to Fig. 2.9, we argued in Section 2.4 how the blockchain provides an elegant solution to problems encountered within IoT enabled industries. Therefore, we analyzed in Section 2.5 the state of the art of the blockchain technology and we surveyed the latest research work conducted on blockchain applicability in multiple IIoT specific industries. However and while blockchains have great potential in establishing a decentralized and secure fabric for IoT enabled industries scenarios and applications, its integration within such systems has raised significant additional fears and concerns about the privacy of participating entities as well as the correctness and safety of smart contracts. On the basis of these considerations, our focus in the rest of this thesis is to ensure first strong privacy guarantees over our DRMF framework as illustrated in Fig. **??**. The corresponding contribution will be detailed in Chapter . Therefore, we will be interested in a second step in behavior based formal verification of smart contracts in order to verify their compliance with the specification for given behaviors. More details about the proposed approach are given in Chapter 5.

Conclusion

We have presented, throughout this chapter, the basic concepts of IoT and Industry 4.0 environments and we listed the main challenges and issues related to their appearance, particularly security and trust requirements. We pointed out then the importance of decentralizing such architectures through the use of blockchains.

Therefore and in a second part of this chapter, our focus lied mainly on enhancing their security aspect in order to achieve trustworthy data during transmission, storage and sharing among the different participating parties within the network guaranteeing their transparency, integrity, authenticity, privacy, confidentiality, and authorization. Where the focus on the blockchain technology. We will provide in this chapter a detailed description of what a blockchain is, how a blockchain network operates, what are its main characteristics and concepts, how smart contracts allow us to radically redefine how interactions between transacting parties on a network can be set up and automated, and what are the main security features that their introduction adds to a system once deployed within.

Chapter 3

DRMF: A Distributed Resource Management Framework for Industry 4.0 Environments

Contents

INTRODUCTION

While smart factories are becoming widely recognized as a fundamental concept of Industry 4.0, their implementation has posed several challenges insofar that they generate, process, and exchange vast amounts of security critical and privacy sensitive data, which makes them attractive targets of attacks and unauthorized access. Introduced attacks and vulnerabilities during manufacturing, in such environments, can be hard to detect and harder to react especially with open connectivity to the external world in addition to the increasing amount of data shared between devices deployed in heterogeneous, distributed and unknown sites while collaborating altogether along the production process. Security requirements in such scenario include integrity, confidentiality, traceability and notarization of exchanged data in the one hand plus access control, privacy and trust in the other one. In this direction, this chapter mainly focuses on the following security properties:

- Confidentiality, tracability and notarization: Entities and resources related information are generally recorded in a different storage structure. Hence, there is a risk that these structures can be subject to unauthorized access and modification. Consequently, traceability and auditability of both the flow of data and their access history records will be challenging.

- Dynamic access management: Dynamic reconfiguration of access rules in response to time, events and more importantly to entities' changing behavior and attitudes.

- Distributed system governance: Distributed management of the shared resources plus the authorization of new entities willing to partake in the system without the need for a third party that could be vulnerable to attacks.

In this chapter, we present DRMF, a distributed resource management framework for Industry 4.0 environments. This last utilizes blockchain based smart contracts technology to support shared resources management and usage between collaborating parties while dynamically manage access authorization over considered resources. Moreover, and in order to better support the security requirement, this framework adds the notion of trust management to the access control model. Here a trust framework is integrated to evaluate access requester entities' behavior guaranteeing thereof dynamicity of security policies insofar that they would be defined and validated function of the access requester entity's behavior.
The rest of this chapter is organized as follows.
Section 2 presents the proposed case study. Section 3 discusses the different challenges of existing access control systems and presents related proposals carried out in the area of blockchain based access control. Thereafter, in Section 4. an overview of the proposed framework is presented and a complete detail of its composition and functioning

principles is provided. Finally Section 5 delves into the implementation of the proposed scheme, a set of experimental results validating our approach are shown.

3.1 Use case study

Throughout this chapter, we will project and illustrate our approach with a scenario example inspired from a real world case study of existing applications related to Industry 4.0.
As a case study, let us take the example of three automaker factories SF1, SF2 and SF3 respectively responsible for: (i) the manufacturing of mechanical and electrical components, (ii) the assembly operations and (iii) the test plus the performance and quality control. These factories over time interact all together along the production processes and alongside with automotive suppliers, as an example a raw material supplier SP1 and a component supplier SP2 proposing, buying and shipping goods and products through transportation partners as it is shown in Fig.3.1.
These parties while looking to ensure sophisticated shipping and logistics operations, agreed to invest in shipping and logistic equipments as well as technologies and personnel to manage their day-to-day trucking operations instead of relying on third parties to ensure them. Obviously, the fact of using their own resources for shipping operations will increase the efficiency of processes, the immediate availability of vehicles, the reduction of cost and the increase of profits insofar that they could rent these resources to other companies and help them load, deliver and unload their items and products. Another important advantage within such decision is the fact of preserving their privacy as well as those related to their resources especially when the third part is not trustful enough so that they could rely on it to ensure such service which could lead to several issues as they put a non trustful third part in control of one of the business functions with the most impact on the smooth running of production processes and the greatest effect on their customers satisfaction.

That's why having their own shipping and logistic resources is really important to overcome such limitations. We notice here that to share such resources among several parties collaborating and working all together and especially that could not always have a strong confidence established in advance, different challenges should be firstly resolved as follows:

- Fully distributed management framework: where we don't need to pass through a third part or to involve several services to manage shared resources and to perform common processes.

- High security level: that guarantees integrity, confidentiality, traceability and auditability of both established transactions and access records and procedures.

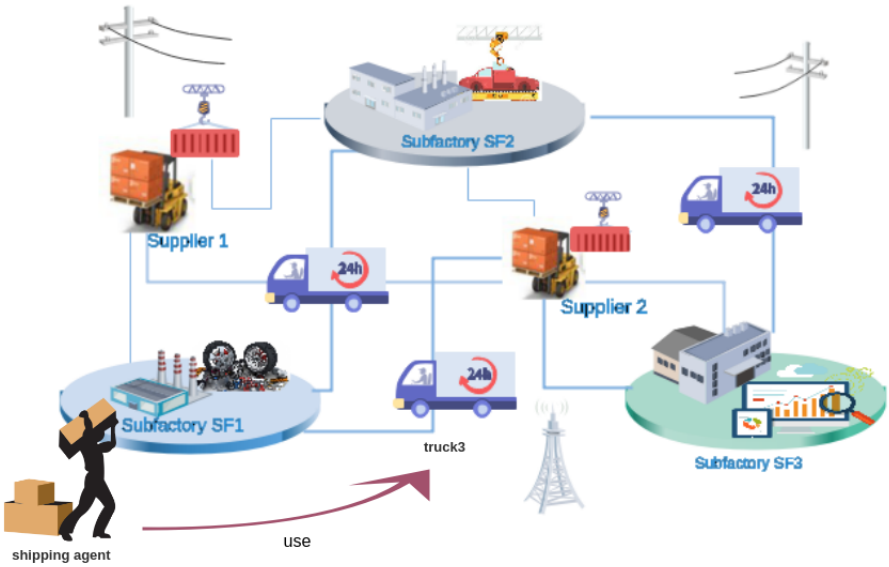

Figure 3.1 – use case study

- Dynamic access control: insofar that security rules can change dynamically in response to time, events and more importantly to involved parties' changing behavior and attitudes.
- Distribute governance: where collaborating parties could join, leave the system and partake in the consensus mechanism any moment they want without worrying neither about the well conduct nor about the security of common processes and shared resources obviously after a consortium is established between the collaborating parties.

3.2 Access control schemes raised issues and problem statement

In this section, we will give first a brief introduction to different access control schemes, we will point out then potential security issues that they may have and we will review research proposals carried out in the current literature to solve such issues.

3.2.1 Access control systems and related issues

An access control model is often used to protect system resources by checking the access made over them in a given access context. The decision is made according to rights subjects have to perform the access they request. These rights are expressed by means of security policies, which consist of a set of conditions evaluated against the current access context to make the access decision each time an access request is received.

In the current literature several models were proposed to define access rights and to control access requests [130, 168], among them: Mandatory Access Control (MAC) [128], Discretionary Access Control (DAC) [146], Role Based Access Control (RBAC) [61], Attribute-based Access Control (ABAC) [186], and Organization Based Access Control (OrBAC) [80]. Many derivatives have been deduced as well from these models in order to resolve a specific need.

Classical access control systems

Discretionary Access Control (DAC) and Mandatory Access Control (MAC), also known as lattice based access control or multilevel security are the first models of access control that have emerged in the early 1970's.

- Discretionary Access Control (DAC): a decentralized access control model where subjects are allowed to decide access rights on objects they own through a set of security rules defined by the triple (S,A,O) meaning that the subject S is permitted to perform the action A on the object O. Access controls within DAC are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
- Mandatory Access Control (MAC): a centralized access control model where resource owners are not permitted to decide who gets access to it, but instead a manager or a central authority has the authority to set access rights to any object by assigning security levels and labels to subjects and objects respectively such as confidential, secret, and top secret. A subject can access only objects that correspond to a security level equal to or lower than theirs in the hierarchy.

Role Based Access Control (RBAC)

RBAC [147] is one of the most widely used access control models that has been standardized and extended in many ways. The key concept of RBAC is the role assignment where roles assigns a collection of permissions to users for the purpose of granting access as it is illustrated in Fig. 3.2. Access is granted by creating a role assignment, and access is revoked by removing a role assignment. The use of roles in RBAC makes it easier and simpler to add, remove, and adjust permissions than assigning permissions to users individually. Moreover, and for efficiency, roles can be structured hierarchically so that some roles can inherit permissions from each other.
However, RBAC suffers from some inflexibility where access permissions could not change in correspondence to network dynamics in changing environments, as in the case of smart factories environments, where production processes information, resource attributes and devices behaviors change as time goes by. Second, and as smart factories consist of a huge number of autonomous and complex systems where each of which

has its own administration service, it seems to be so challenging the fact of managing thousands of roles and mapping them to operations and users respectively and thus the setting up of the RBAC model may end up being an infinite process for big organizations or systems.

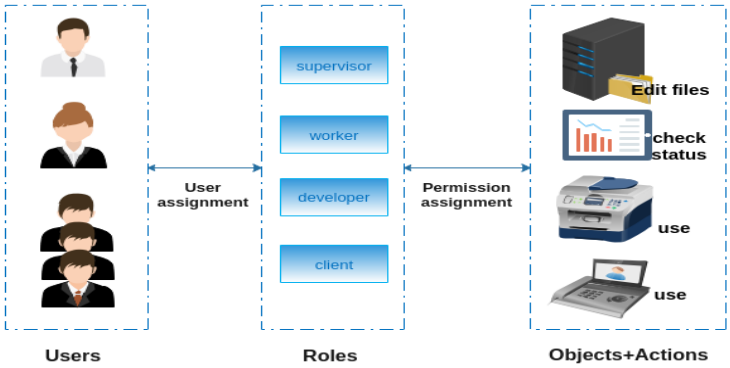

Figure 3.2 – Role Based Access Control

Attribute Based Access Control (ABAC)

ABAC [76] can be defined as a generalization of RBAC where security rules can be evaluated and access authorization can be determined based on various attributes presented by the subject, the object and the environment. Attributes may be considered characteristics of anything that may be defined within the access control model and to which a value may be assigned. Basically, ABAC relies upon the evaluation of attributes of the subject (that correspond to users information such as department, role, job title, address, etc.), attributes of the object (corresponding to resources characteristics such as the status, type, location, owner, etc.), the environment (representing the environmental conditions such as time, location, humidity, temperature, etc.), plus a formal relationship or access control rule defining the allowable operations for attributes combinations. Fig. 3.3 illustrates an ABAC access control scenario where a subject requests access to an object. This mechanism assembles policy, subject attributes, object attributes and environment conditions to determine and enforce a set of allowable operations by the subject upon the object.

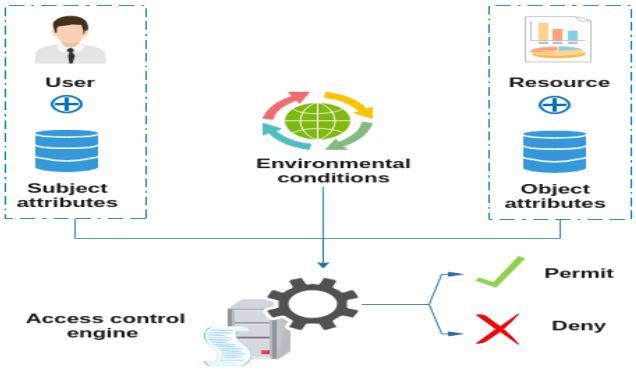

Figure 3.3 – Attribute Based Access Control

ABAC enables more precise access control by allowing for a higher number of discrete inputs into access decision and thereby providing a larger set of possible combinations of those variables to reflect a larger and more definitive set of possible rules to express policies, which are limited only by the computational language and the richness of the available attributes.

Organization Based Access Control (OrBAC)

OrBAC model [80] could be considered as one of the richest access control models in terms of components and applicability to many realistic situations. The main originality of OrBAC is the organization entity defined as a structured group of active entities, in which subjects play specific roles, in addition to the distinction it makes between the abstract and the concrete level as it is illustrated in Fig. 3.4. More specifically, instead of modeling the security policy by means of concrete concepts related to subjects, objects and actions, OrBAC relies on three abstract entities for expressing security rules namely roles, views, activities, where subjects are abstracted into roles to which the same security rule applies. Also an activity is a group of one or more actions, a view is a group of one or more objects. In the decision making process, OrBAC takes into consideration various context information representing a specific situation conditioning the validity of the policy rule and that could be temporal, spatial or declared by the subject. Moreover, in OrBAC, specification of security policies is no more restricted to permissions. Instead, a security policy may include four different access types namely: permission, prohibition, obligation and dispensation. Intuitively, a prohibition is a negative permission implying that one must not perform some action. An obligation is associated with an action someone must perform and is usually triggered when some conditions are satisfied. Finally, a dispensation is a negative obligation implying that one may not perform some action.

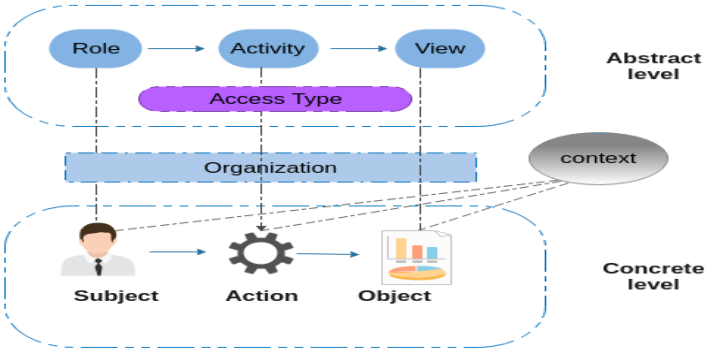

Figure 3.4 – Organization Based Access Control

However, even if OrBAC has several advantages for expressing security policies, it is unfortunately only adapted to centralized structures and does not cover the distribution, the collaboration and the interoperability needs when considering collaborative systems and multi organizations based environments as it is the case of smart factories and Industry 4.0.

Table 3.1 summarizes the already presented access control systems regarding their dynamicity, scalability, context awareness and rules expressivity.

Access control model	Scalability	Dynamicity	Context Awareness	Rules expressivity	Pros
RBAC	Policies cannot evolve easily because the creation of new roles can lead to rebuilding the entire model.	Access permissions defined in a static and fixed manner thus could not change in correspondance to network dynamics	Access rules statically defined without taking the context into consideration	Access rights based on roles	
ABAC	Scalable	Access control decisions defined to accommodate dynamically changing attributes	Additive information related to subject/object and environment considered	Access right based on attributes related to resource, subject and environment.	Fine grained More scalable More flexible More interoperable Support of delegation
OrBAC	Scalable	Dynamic security policies with the use of contexts	Takes the context (e.g., specific situations, time and location constraints) into account.	Access rights based on roles, views, activities and context. Various types of access such as permission, prohibition, obligation and recommendation.	Structured and an abstracted expression of the policy Fine grained More scalable More flexible More access right types

Table 3.1 – Classical Access control models

3.2.2 Discussion

In the context of smart factories and considering the need for dynamic security rules definition and parameterization, context awareness, security rules abstraction, scalability of resources, actions, subjects and situations, plus expressivity and fine-granularity,

we can consider OrBAC as a good candidate for providing an adequate access control model for such environments.
Reminding that within a multi factories based environment, entities belonging to different factories interact with each other in order to realize a common goal where the concept of multi organizational environment characterized by large scale and independent structures with decentralized systems, where each factory defines its own model, assigns its own roles and specifies its own access control policies.
The question to be asked here how to verify the role attribution procedure and how can we provide flexibility to entities to fully control their roles as well as access requests related to their resources? Another issue is the integrity and the confidentiality where entities access requests and operations history are stored within a local database or a cloud infrastructure. Hence, there is a risk that these databases can be subject to unauthorized access and modification, without neglecting the fact that they remain a bottleneck and a single point of failure that can disrupt the entire system. Consequently, traceability, notarization and auditability of access records will be challenging. One more issue is privacy breaches and lack of transparency in such environments where an undebatable lack of a trust is arising especially towards third part getting access to data collected by billions of entities creating information. Hence the need for a security through transparency approach allowing users, machines and collaborating parties to retain their anonymity and to preserve their privacy. A last but not least issue is the difficulty of managing security policies according to the context dynamicity especially with the colossal number of entities supposed to be managed and that could change their behavior over time.
That's why providing an adequate solution to distributively govern the system and to dynamically, securely and contextually manage the access while keeping a living document trace about the flow of resources data being shared become crucial and no more important than ever.

3.2.3 The proposed solution: Decentralizing smart factories environments in a traceable and secure way through the blockchain

A distribute, secure, transparent, consensually and publicly verifiable solution to smart factories based environments would solve many of the raised issues described above. In this direction and given the noted features of blockchain technology, this last applied to such systems provides promising possibilities and solutions to issues they encounter. Actually, the decentralized, autonomous, and trustless inherent capabilities of the blockchain technology make it an ideal component to become a fundamental element for addressing security issues in collaborative and resource sharing based smart factories systems.

However, adopting the blockchain technology is not straightforward and will require addressing the following research questions:
1- How to ensure a distributed system governance where entities can join, leave the system and get involved in the consensus mechanism without worrying neither about the well conduct nor about the security of common processes and shared resources?
2- How to provide a lightweight, dynamic and trustworthy access control framework that reconfigures access rules in response to time, events and more importantly to entities' changing behavior and attitudes?
3-How to guarantee the integrity and the consistency of access requests and operations history and how to trace the interception and the modification made over them?
4-How to enable a strong privacy guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies?
In this direction, few proposals of blockchain related access control systems have been presented in the current literature.
Recently, The blockchain technology was used as a storage structure for access control policies in [106]. Therefore, its computing feature was examined in [129] where it plays the role of a decentralized access control manager. In this work, FairAccess was proposed to offer a fully decentralized pseudonymous and privacy-preserving authorization management framework for IoT devices. The proposed framework used OrBAC access control model to enable users to own and control their data whose policies were stored in a private blockchain solution. Subsequently, the idea of using smart contracts for achieving access control has been adopted in [51], [52], [95] for different access control systems.
In [51], authors proposed RBAC-SC, a Role Based Access Control system using Smart Contracts. In this model, Ethereum's Smart Contract technology was used to realize a trans-organizational utilization of an organization's roles.
In [95], a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0 was presented. The proposed framework leverages the underpinning characteristics of blockchain as well as several cryptographic materials to realize a decentralized, privacy preserving solution. However an implementation of the proposal is missing as well as the evaluation and the proof of its performance.
In [191], a Transaction based access control platform was proposed where the ABAC model was integrated within the blockchain system to manage subject registration, object escrowing and publication plus access request and grant.

3.2.4 Discussion

As seen, the blockchain technology has been used for several purposes and within various models of access control. However, just few studies have focused on integrating the

blockchain technology within access control systems in the context of smart factories environments. Our focus in this work is not only to ensure fine grained, flexible and secure resource access authorization in the context of smart factories environments but also to support distributed and dynamic governance and management of the overall system where all relevant parts can make a comprehensive decision of joining entities registration and consensus management for entities requesting mining permissions. Another focus of this framework is the integration of trust management with the access control model in order to determine whether subjects are trusted and well behaved enough so that they could access resource data.

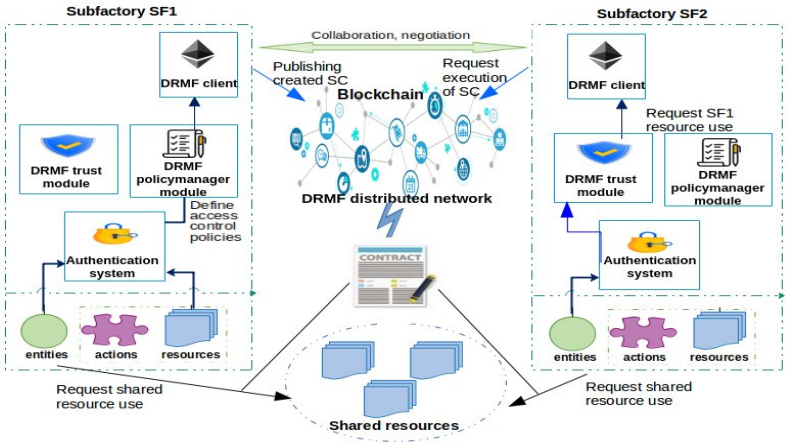

Figure 3.5 – System architecture

3.3 Proposed Approach

3.3.1 Overview

In this chapter we present DRMF, a Distributed Resource Management Framework based on the blockchain technology for Industry 4.0 deployments. The main objectives of this work are as follow:

- Information notarization: The use of blockchain to keep a living document trace about the flow of data and resources being shared by collaborating entities guarantees an extra level of transparency, control and notarization during collaboration where a proof of existence, of ownership, of access and modification is essential for decision making process.

- Distributed system governance: For each entity willing either to partake in the consensus mechanism or to join the blockchain network and sharing common resources, a verification of its behavior by most participants in the system is made in order to ensure its legitimacy absolutely necessary to confirm its joining request. The registration of new entities as well as the management and the elimination of

joined ones require an agreement to be reached between the pool of entities taking in charge the consensus mechanism.

- Dynamic access management: this framework achieves distributed, dynamic, contextual and trustworthy access authorization through the integration of the OrBAC access control model within a distributed ledger where transactions serve as verifiable and traceable medium of policies definition and parametrization, access request procedures as well as related operations.

Fig. 3.5 shows the overall structure of the proposed system. As illustrated, each domain (e.g. SF1, SF2) holds its own entities with different roles, for example a human worker in the manufacturing subfactory SF1, a supervisor agent in the quality control subfactory SF3, these last could perform several actions on each shared resource such as using the shipping trucks, sharing their trust records, querying for available ones, etc. Let us assume that a human worker within SF1 wants to have access over the shared resource Truck 3 to perform a shipping operation. To do so, the process would work as follows: the agent needs to get authenticated first along with an authentication system, which is responsible for making authorization decisions, verifying devices identities and generating authorization tokens. Furthermore and instead of statically evaluating the access request, a DRMF trust-module is involved to assess the trustworthiness degree of the requesting entity and to judge its behavior taking into account different trust aspects and parameters (that will be discussed in Section. 3.3.2). The result of this evaluation will be incorporated within the context structure to be sent to the DRMF policy-manager-module which is in charge of formulating the corresponding transaction to be broadcast via the DRMF client to the DRMF distributed network where the corresponding smart contract will be executed. Reminding that within a smart factory environment, IoT devices are used to collect and analyze data coming from smart products, other smart devices and related smart services, that's why in the case of IoT devices with tight resource constraints, DRMF modules are assumed to be deployed in more powerful network components that will be connected directly to each device, otherwise it is deployed within the device/entity itself.

3.3.2 System composition

In the following we will detail the different software modules composing our system, the specific role and the operation of each smart contract as well as the workflow of the overall architecture. As shown in Fig. 3.5, our framework consists of the following components:

DRMF distributed network

This component is the main core of our system. It defines: (1) a set of smart contracts ensuring the notarization, the distributed governance and the dynamic access management, (2) a distributed network composed of a set of peers responsible mainly for receiving the established transactions, verifying and validating its state and executing the functions contained within.
In this work, three Ethereum based smart contracts are considered namely: Access Contract (AC) designed to manage access authorization made over shared resources, Governance Contract (GC) used to ensure a distributed governance of the overall system, and Lookup Contract (LC) acting as a register to map between the required services and the contracts ensuring their management. More details about each smart contract will be provided in the following subsections.

DRMF client

This component implements the full functionality required to join and to participate in the DRMF distributed network. This handles a broad set of tasks, such as connecting to the peer-to-peer network, encoding and sending transactions, keeping and exploring blocks copies and deploying and interacting with smart contracts.

DRMF-Trust module

This component is in charge of assessing the trustworthiness degree of shared resources as well as requesting entities in order to capture their ability and willingness to behave as expected. The realized assessment will be linked with access control decisions through the context structure defined within security rules that will no longer be statically defined and parameterized, but instead will depend on entities behavior in addition to the environment dynamicity and context change. We remind here that our proposal is based on the OrBAC access control model where the integration of trust management to enhance the security level of the corresponding system has been studied by the research area during the last years. There are multiple ways to define trust and to evaluate it. This last could be based on a set of paramaters where the most relevant ones defined in the literature are experience, reputation and knowledge [167], the experience parameter corresponds to the interpretation of the previous interactions established with immediate neighbors. These evaluations will be propagated as trust recommendations to other network nodes to constitute the reputation parameter, that once kept, will be considered after as the knowledge part of trust. How to evaluate trust is not the topic of this chapter, however we will provide an example of metrics that matching with the proposed use case, could be considered as effective ones within the trust evaluation process.
We remind here that according to our use case, the evaluation of trust will focus on

both access requesting entities and requested resources (which are the shipping trucks) as target nodes addressed by the trust model. The trustworthiness assessment of the shipping resources could be based on their speed, position, direction of motion, engine coolant temperature, battery voltage, real-time diagnostics and performance evaluation results of vehicle status, the trustworthiness degree of their drivers, etc, [189].

DRMF-PolicyManager module

This component serves in the one side as the defining part of access rules to be encapsulated into transactions and reloaded to the blockchain after validation, on the other side this module serves as the acquisition source of attribute values required for policy evaluation that once received and intercepted will be sent to the corresponding smart contract.

Authentication system

This component is mainly responsible for verifying the validity of entities' identities as well as the legitimacy of demands and requests sent to the blockchain network. Participating entities are authenticated based on the provided credentials. In our framework, we rely on the openID Connect [144] (OIDC) which is an identity layer on top of the OAuth 2.0 protocol [73]. We have chosen OIDC since it is free, open and decentralized (no central authority approves or registers relying parties or service providers). Its integration does not require complicated update in the deployed application. Indeed, it follows a restful approach which makes it easy to use and to interoperate.
Remembering here that at the beginning of the authentication process, participating entities are authenticated based on the provided credentials. These last can be represented by using different mechanisms, such as login/password, digital certificates, authentication keys, etc. In case of a successful authentication process, this module generates an access token which is delivered in order to avoid subsequent authentication procedures.

Smart contracts

In our work, we proposed three smart contracts: LC, AC and GC.
Lookup contract (LC): The main role of this smart contract is to map between the required services and contracts ensuring their management. To do so, it maintains a lookup structure that registers the required information to find and execute the methods in question. This structure contains the name of the smart contract in which the method is developed, its address, the address of its creator and the name of the requested method. The operation of this smart contract will be mainly based on the following methods:

- lookup(): This method receives in input the name of the requested method to return the address of the corresponding contract (i.e., the access contract AC, the governance contract GC).
- addFunction(): This method receives in input the information details of a new function to add to the lookup structure, obviously only the creator of the corresponding smart contract can add new methods.
- deleteFunction(): This method receives the name of an existing function to delete from the lookup structure.

Access contract (AC): This smart contract is mainly designed to achieve distributed, interoperable, contextual, trustworthy and secure access control for multi organization systems where participating parties interacting and collaborating all together share common resources for which access rules should be maintained and parameterized by the collaborating parties. This smart contract is based on the integration of the OrBAC access control model within a distributed ledger to express access control policies. We opt here for such integration firstly to guarantee the fact that access policies are available at any time and evaluated in distributed environments where there is no central authority to define roles and to generate security rules what would overcome the problem of having a single point of failure. Secondly to ensure verifiable and transparent role assignments where any entity can verify if another one owns really the role it pretends to have and that this last is managed and issued by its factory or belonging organization. Third to enable the dynamic reconfiguration of access rules in response to time, events and more importantly to entities' changing behavior and attitudes.
To ensure such features, this smart contract maintains a set of security rules. Table. 3.2 illustrates a simple example of rules where each row corresponds to the policy defined on a certain tuple. Basic fields of each row are:

- View: it represents the resource for which the access is requested. According to our use case scenario, this last could be a record related to production data, manufacturing entities, trust records, etc.
- Activity: it represents the action to be performed on the resource such as check, update, use, etc.
- Role: that represents the entity requesting the access for a certain resource.
- Context: it is used to express different types of extra conditions or constraints that control activation of rules expressed in the access control policy.
- Access type: it defines the access type defined on the view, according to the OrBAC model, this last could be permission, prohibition, obligation and recommendation.

View	Activity	Role	Context	Access type
Truck related file	update	Supervisor	Sup.Trust-score > T-Th1 AND Current-time IS IN Working-hours	Permission
Genesis-block-addr	mine	BC-node	BC-node.Trust-score > T-Th2 AND Node-registered	Permission
Truck 3	use	human worker	Worker.Trust-score > T-Th3 AND Current-time IS IN Working-hours AND Department IS IN Shipping	Permission

Table 3.2 – Security rules list example

Returning to the use case presented in Section. 3.3.1, to perform the shipping operation using the Truck 3, an evaluation of the agent's access request is needed first to allow or not the demanded access. To do so a verification of the well conduct and functioning of both the requesting entity and the requested resource is essential, hence an authorization over the access control smart contract is required to decide whether the access is permitted or prohibited. The access authorization and according to the OrBAC access control model depends on a set of contextual conditions whose activation will activate the corresponding rule. In this example, to have the requested access accepted, the human agent should have a trust score above the defined threshold for the corresponding role, he should belong to the shipping department and request to use the Truck during his working hours. We can notice here that the fact of using trust management within the access control would enable the dynamic reconfiguration of security rules that will change in response to involved parties' changing behavior and attitudes. To do so, we have added a novel type of context which is related to trust management. The role of this latest is to check if the trust levels of both the access requesting and the requested entity respect well the threshold defined.
The operation of the Access contract will be based on the following methods:

- addpolicy(factory id, role id, string activity, view id, struct context, string permName): this function launched by resource owners aiming to define and to add a new access control policy for a newly shared/stored resource. It takes as input the owner's belonging factory public key, the role id of the subject, the view id of

the object, the context in which the access is demanded, the permission to be attributed. As a result of this function, a new policy item will be added to the policies list.

- Updatepolicy(factory id, role id, string activity, view id, struct context, string newPerm): this function launched by resource owners in order to update an already added access control policy.
- deletepolicy(factory id, role id, string activity, view id, struct context): this function receives the main identification information of a policy to be deleted from the policy list.
- accesscontrol(factory id, entity id, string activity, view id, struct context): this function executed by a subject requesting entity in order to authorize its access request upon a certain resource identified with its view id within a certain context. To do so a verification of the request is made to check the validity of the subject role, the existence of the policy within the defined ones and the behavior of the requesting entity to detect a potential doubtful/suspect access demand. As a result an access result will be returned to both the requesting and the requested entities and the access process will be executed.

Governance contract (GC): This contract is mainly designed to govern consensus and to manage who can partake in the consensus mechanism within the blockchain network. More specifically it is responsible for determining the consensus algorithm to be executed for the mining procedures, as well as for registering and managing miner entities. To do so, this contract is supposed to store blockchain addresses related to entities having either transaction validation, mining or voting permissions. For registering miner entities, the consensus contract is used to validate entities requesting mining permissions in order to be allowed either to validate, create and add new blocks to the ledger. Here we note that once the system is deployed, this contract will contain initial validators representing the collaborating parties. For overwriting miner entities, a request to delete the entity in question is submitted by the one who has noticed its malicious behavior or its breakdown. Thus once the rest of entities have reached a majority, the miner will be removed from the consensus contract.
We remind here that in order to be registered, or to send overwriting instruction, the requesting entity needs first to get authenticated and authorized over the access control smart contract, then the request will be transmitted to the pool of entities taking in charge the consensus mechanism, once confirmed that it does not pose a threat to the system, its address will be added to the governance contract.

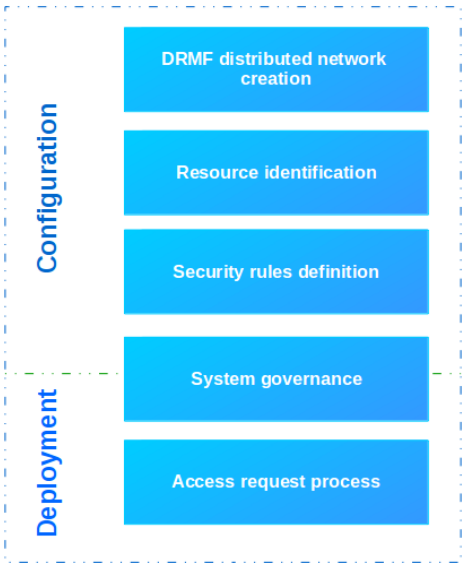

Figure 3.6 – Proposed framework phases

3.3.3 Prototype workflow

A description of the different phases and operations of the proposed framework is presented in Fig. 3.6. This last involves a succession of operations wherein:
(i) DRMF distributed network is created and a consortium is defined in order to reach the desired agreement among collaborating parties that once identified get registered and involved in the consensus procedure.
(ii) Given the existence of resources to be shared, these last are identified in order to be accessed and used in a notarized manner.
(iii) For such purpose, first security rules should be defined in order to manage the access among the collaborating parties.
(iv) Entities requesting to join the system or to participate in the consensus mechanism, are registered obviously after evaluation of their behavior, corresponding roles are identified and attributed as well.
(v) When willing to perform an action over an existing resource, the requesting entity sends an access request that further to which a decision is made.
These five steps our proposed framework is made up of are detailed in the next paragraphs.

DRMF distributed network creation and entities registration

Once the DRMF distributed network is created and becomes operational, the process of adding new entities with different roles and classifications can be launched. Here, it should be assumed that new entites have been already identified within the authentication system and have a public identifier that is unique to their organization. It should also be assumed that they have received an Ethereum address required to partake in

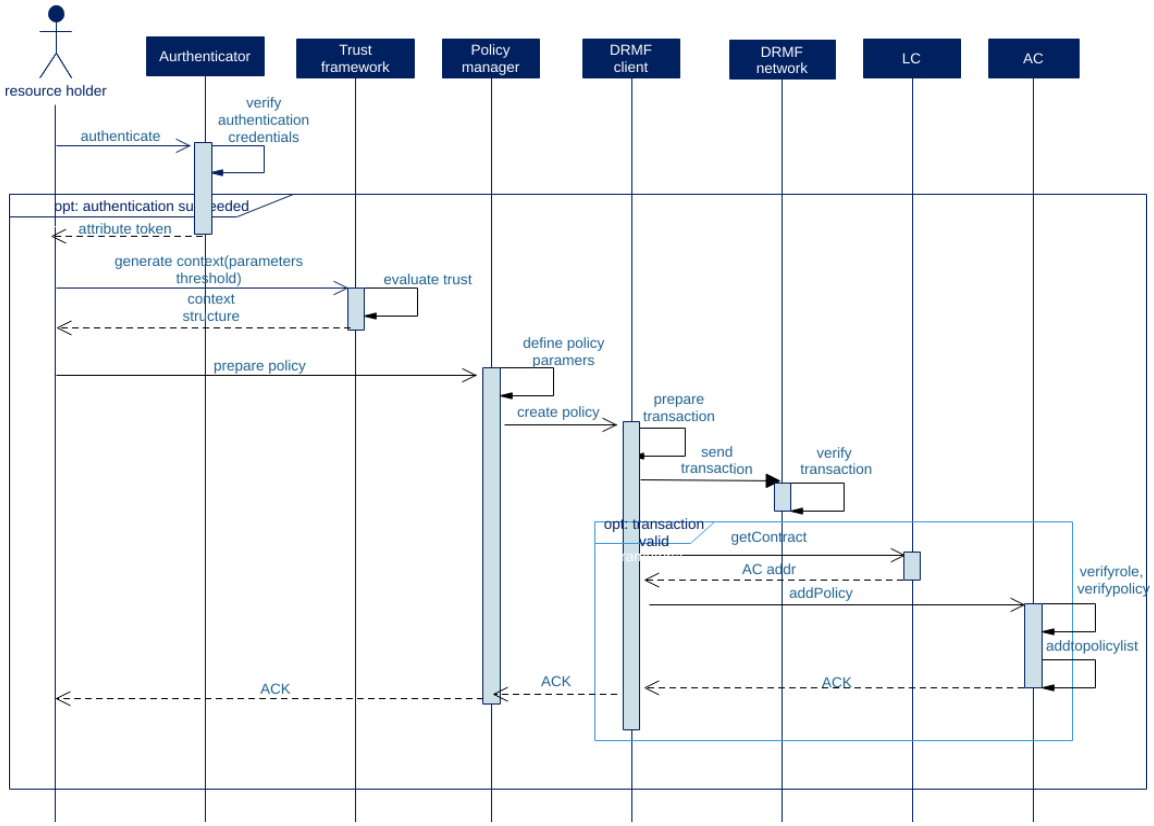

Figure 3.7 – Policy creation transaction workflow

the Ethereum network. Therefore, the process of adding a new entity begins by having the pool of consensus entities validate that the public identifier corresponds to the pretended role and suits the requested classification.

Security rules definition

When collaborating sub factories agree on adding an access control policy for a newly deployed resource to be shared among them during a certain time. For example collaborating sub factories SF1, SF2, and SF3 agree on adding a new truck identified with a new attributed address R_addr1. To manage the access over the shared resource, an access policy needs to be added to the AC smart contract and shared within the DRMF distributed network. The proposed framework works as illustrated in the UML sequence diagram in Fig. 3.7. The resource holder needs to get authenticated first along with the authenticator, in case it is already authenticated, the attributed token is used to have access to the system. Therefore, the DRMF-Trust module evaluates the resource trust value, defines the trust threshold to be satisfied by the access requesting entity and generates the context structure to be sent to the DRMF-PolicyManager module. This last defines the possible roles, the activity to be performed and encapsulates the defined access control policy in form of a scripting language in order to be added to the policies list within the AC and sends it to the DRMF client in order to be broadcast to the

DRMF network. The peer to peer nodes verify the transaction, and record it within the distributed ledger in case of success validation. At this stage, a new AC is created and deployed on the blockchain, obviously a new entry is added to the LC in order to register the required information of the newly created AC via the addFunction() method.

Access request transaction workflow

In case of access request, the proposed framework works as follows: A subject (e.g., a human worker within the supervision service of the performance and quality control subfactory SF3, identified with his ethereum address E_Addr) wants to perform an action (e.g., execute) on a protected resource (e.g., a controlled robotic arm responsible for auto body panels spot welding operations within the assembly subfactory SF2, identified as well with its ethereum address R_addr). The subject E_addr after being authenticated within the authentication system for a first authentication case or after validating the access token it uses, will submit its access request to perform the execute action on the resource R_addr. The DRMF-Trust module at this stage receives the access request, assesses the entity trustworthiness, derives its trust value, prepares context related attributes and generates the context structure to be sent to the DRMF-Policymanager module acting as a Policy Enforcement Point (PEP). This last formulates the access request to an access transaction and broadcasts it to the peer to peer network via the DRMF client in order to run the AC. We remind here that before sending the access demand transaction, the DRMF client needs to have the address of the AC, to do so it calls the getContract method of the LC to retrieve the AC address and concerned method. Once received, the access demand transaction is sent to the AC acting as a Policy Decision Point (PDP) that evaluates the access demand by verifying the validity of the subject role, the existence of the access policy within the defined policies list and especially checking the subject's behavior, as a result it determines whether the request should be permitted or denied. Finally, if it is permitted the transaction is valid and it will be recorded in the blockchain else the transaction will be rejected and a notification will be sent to the requester.
The described workflow is illustrated in Fig. 3.8.

3.4 Implementation and evaluation

In this section, we will introduce first the different tools used for the implementation of our framework therefore we will show experiment results demonstrating its feasibility.

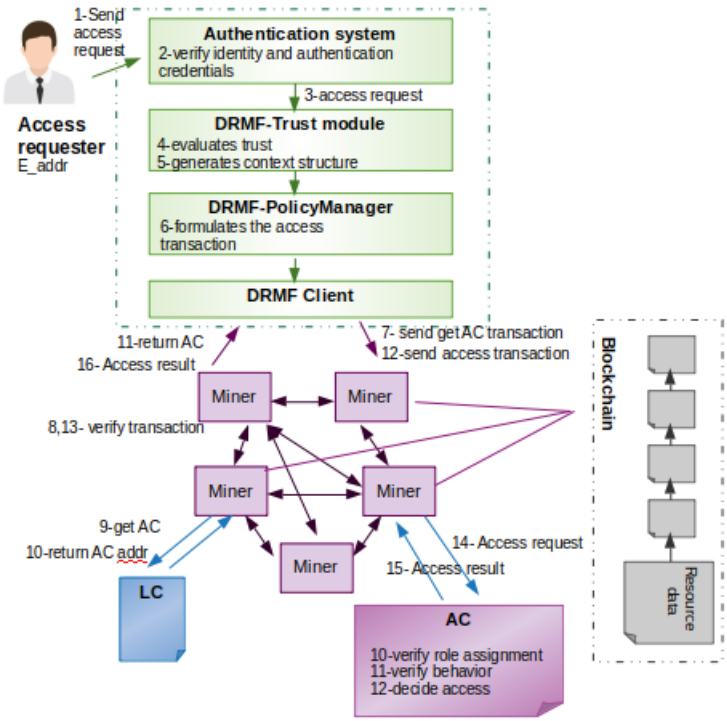

Figure 3.8 – Access request transaction workflow

3.4.1 Work environment

For the implementation of our proposal, we have set up a testbed as illustrated in Fig.3.9 featuring several hardware and software components as listed below:

- a Dell Precision M6800 machine with 4th Generation Intel Core i7 processor and 8Gb of RAM in which we have configured a private Ethereum blockchain network consisting of three nodes having the functionalities of Ethereum miners.
- an Intel Core i5-3210M laptop with 6 Gb of RAM and 2.40 GHz of CPU frequency in which an Ethereum node was set up.
- two Raspberry Pi 3 Model B configured to act as shipping resources shared among the collaborating factories.

For the implementation of our proposal, we chose Ethereum, which is currently the most common blockchain platform for developing smart contracts. According to the use case study we introduced in Sec. 3.1, our scenario consists of 3 subfactories SF1, SF2 and SF3, where the administration service is represented by each Ethereum node within the Dell machine. These factories collaborate altogether alongside the production process and sharing as common resources two trucks responsible for the trucking and shipping operations. These last are represented by the two single boards (RPi3) that are connected to the blockchain network where related information according to their real-time diagnostics and performance evaluation in addition to their trustworthiness degree are collected, sent and stored within the distributed ledger. Moreover a human

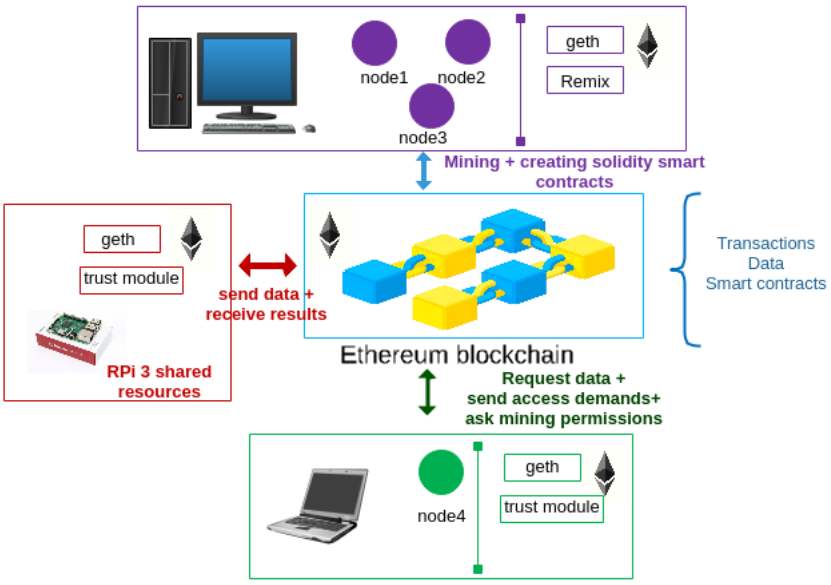

Figure 3.9 – Work environment

worker within the shipping service willing to perform a shipping operation using a truck resource is represented by the laptop machine.
For our prototype implementation we use geth client [66] which is a command line interface for running a full Ethereum node implemented in Go language As illustrated in Fig.3.9, a geth client is configured on each entity so that it could act as an Ethereum node. For each entity we have created an account and set it to form the DRMF network. Mining tasks are ensured by the Dell machine insofar that it has a relatively large computing software and storage capability where the Proof of Authority (PoA) consensus mechanism is supported by each node. This entity took in charge also the creation and the deployment of solidity smart contracts. In what concerns the trust module integrated within either the laptop machine or the RPi share resources, we have used the model presented and implemented in the chapter 6.

3.4.2 Proof of concept

As it was presented in Sec.3.3.2, each smart contract is based on a set of methods developed under specific algorithms according to the defined use case scenario. Reminding that our DRMF framework is based on the following main functionalities: (1) Registering a new resource with a corresponding address. (2) Definition of security rules. (3) Access request. (4) Behavior evaluation. (5) System governance.
In order to show the feasibility of our proposal, we conducted some experiments related to the access control and the consensus governance procedures. For the validation of the access contract, we added a new role 'shipping worker', a new context with the trust threshold, the affiliation and the working hours, a corresponding policy then is added to the policies list that according to the OrBAC model, is specified as follow:

permission(org: SF1, shipping worker, use, truck3, shipping worker.trust-score > trust-th3 AND current-time IS IN working-hours AND shipping worker.department IS IN shipping)
Therefore, we defined a malicious behavior where an on-off attack is launched to cause a low trust score that was calculated by the DRMF-trust module. As a penalty for the malicious behavior, access requests from the subject will be blocked for a certain period of time (until the trust score is above the trust threshold fixed to 0.4). Fig.3.10 and Fig.3.11 show access results displayed after a legitimate behavior and a malicious one respectively.

For the validation of the consensus contract, we need here to prove the well achieve-

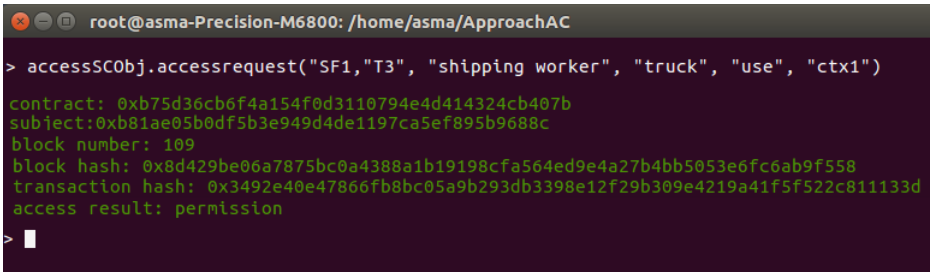

Figure 3.10 – permission of access

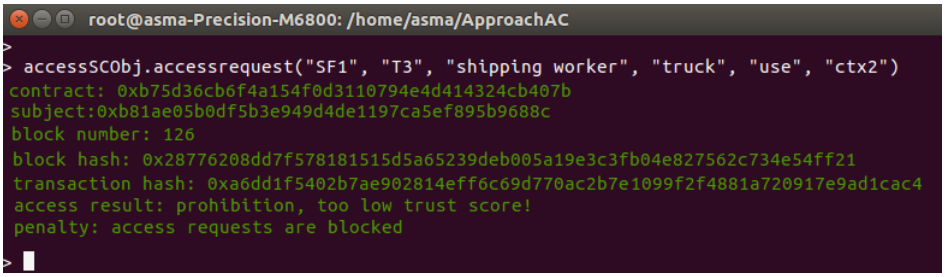

Figure 3.11 – prohibition of access

ment of consensus while dynamically adding and removing nodes to and from the network. We will consider the case of a new node willing to partake in the consensus mechanism within the blockchain network. This last needs to get authenticated and authorized first over the access control smart contract, then the request will be transmitted to the pool of entities taking in charge the consensus mechanism, once confirmed that it does not pose a threat to the system, its address will be added to the consensus contract. We added a new role 'supervisor', a new context with the trust threshold, the affiliation and the working hours. We specify thereafter the policy to be added as follow:
Obligation(org: SF1, supervisor, mine, genesis-block-addr, registered AND supervisor.trust-score > trust-th2
AND current-time IS IN working-hours AND supervisor.department IS IN administration)
Fig.3.12 shows corresponding access results.

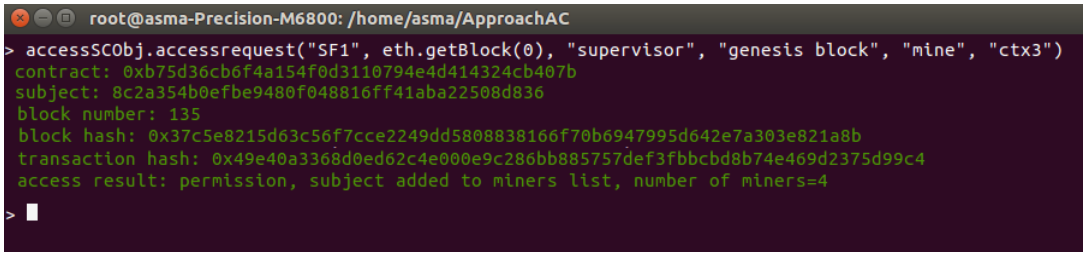

Figure 3.12 – registration of a new entity within the consensus mechanism

3.4.3 Discussion

In this paragraph, we analyze the different features provided by DRMF that utilizes blockchain to keep a living document trace about the flow of resources being distributed and shared among collaborating parties while implementing distributed, dynamic and secure resource access authorization.

Transparency

The DRMF framewwork achieves the transparency property since all functions executed within smart contracts are reflected on the corresponding log of both the respective smart contract and the DRMF distributed network. By this way, an entity could not perform any transaction without other entities' knowing and validation, and also it cannot deny any transaction it has committed.

Verification

The DRMF framewwork effectively achieves verifiable access control through the verification of role assignments, access rules existence and non violation in addition to the legitimacy of access requests implemented within the AC.

Flexibility

Our proposal gives participating entities the flexibility to join or to leave the system or even to partake in the consensus mechanism easily. Functions implemented within the GC ensure the registration of new joining entities, the overwriting of existing ones and the management of those taking in charge the consensus mechanism.

Dynamicity

Dynamic reconfiguration of access rules in response to entities' changing behavior and attitudes is ensured within the AC that does not only verify and validate access authorization statically by checking access rules' defined conditions are met, but also dynamically by checking the behavior of the access requesting entity and judging its legitimacy or maliciousness according to the trust score derived by the DRMF-Trust module.

Conclusion

We have presented in this chapter the design and the implementation of a distributed resource management framework based on the Blockchain technology in order to (1) notarize the flow of data and resources being shared by collaborating parties, (2) achieve a secure, trustworthy, fine-grained, and traceable access control and (3) dynamically and distributively manage new joining entities as well as those willing to partake in the consensus mechanism. Combining the resource management system with the blockchain technology, we enable a more reliable resource data confidentiality and integrity verification during sharing and we make a time-stamped log of both entities' access transactions and behavior. However, adopting the blockchain technology to handle shared resources management and controlling the access made over them is not straightforward and additional critical issues emerge that are:

1. The public and transparent aspect of the blockchain comes at odds with the private aspect of access control policies publically recorded in the blockchain. Often times, the policies for determining who can access the resources are sensitive also and need protection as well.

2. Traceability issue where the verifiable and traceable feature of blockchain may leak access requesters privacy and sensitive information, insofar that their access history related details once recorded in the blockchain, may conduct to learning their authorization functionality pattern, as well as to tracing their access activities.

3. Correctness and safety issue of smart contracts that could be exposed to a variety of security threats and attacks leading to significant malicious scenarios resulting in terrible losses.

To deal with such issues, we will present in the next two chapters how to adapt the proposed framework to tackle these challenges and to achieve the required security level.

Chapter 4

Privacy Aware Distributed Access Management Framework

Contents

INTRODUCTION

We have discussed in the previous chapter how the blockchain technology could be an efficient solution for access control schemes to address challenges they encounter regarding data confidentiality, immutability in addition to traceability and notarization of access demands and authorizations. We have seen also how using the blockchain technology within such systems has raised significant additional fears and concerns about the privacy of participating entities regarding both access history and shared policies in the blockchain. On the basis of these considerations, our main focus alongside this chapter is to ensure strong privacy guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies. The proposed scheme is then integrated within our DRMF framework to preserve the anonymity of both the access requester entities as well as the collaborating parties, by this way the transparency feature of our framework will be maintained while guaranteeing and preserving the privacy of its users.
Alongside this chapter, the first Section presents the motivation behind this work illustrated by an example scenario inspired from the case study presented in the previous chapter. We provide in Section 2 then some insights into privacy issues associated with the blockchain and discuss different techniques considered by existing works. Thereafter, in Section 3 an overview of the proposed scheme is presented and a detailed description of our system composition and functioning principles is provided. Finally Section 4 delves into the implementation of the proposed scheme, a set of experimental results validating our approach are shown.

4.1 Research motivation

In a large scale smart factories environments where interacting parties collaborate all together along the production processes while sharing common resources for which access is controlled and managed via a distributed and a consensus based framework. These parties in certain cases are from different sites and sometimes even have diversified competing and conflicting interests with each other. Within such scenario, an agent for example or an entity being part of such an organization might not want other involved parties to associate his identity to the data he requests, or the time he accesses the network. However, he may wish to keep confidential whether he accessed such resources, what data types he was interested in, or from which part he obtained the data he requested. Also and in case of malicious competitors existence, although these last cannot read shared data files and resources related information, they can read stored access control policies, and thus they can deduce information about an organization current activity details, they can even try to satisfy the access condition in order to get authorized to the corresponding resource.

To address such issues, a new decentralized access control framework is required to preserve in a strong way the privacy and the anonymity of entities alongside the access procedure.

4.2 Privacy issues in Blockchain and considered strategies

Generally speaking, to protect privacy, the blockchain system needs to satisfy the following requirements:

- Identity Privacy: which means inlinkability between established transactions (with random addresses and pseudonyms) and the real identities of their partakers as well as the transactional relationships between interacting parties.
- Transaction Privacy: which means that the transaction contents (e.g., amount or transacting patterns) can only be accessed by specified users, and kept unknown to the public blockchain network.

In this direction, various privacy preservation strategies have been proposed in the literature either in the context of public blockchains or in the context of permissioned blockchains [60, 75]. These last could be divided according to [60] on the basis of privacy requirements into two main categories as illustrated in Fig. 4.1.

4.2.1 Identity based privacy preservation techniques

While users use random addresses and pseudonyms when acting in the blockchain, their real identity and personal information could be easily revealed using various strategies such as behavioral analysis tools, traceability techniques, etc. In order to achieve a higher level of identity privacy and security, anonymization, unlinkability and untraceability features should be guaranteed. Currently, to achieve such goal there are four frequently-used mechanisms for protecting anonymity in blockchain [60, 75]. These last include: mixing services, ring signature, anonymization, and non-interactive zero-knowledge proof as presented in Fig. 4.1.
In the following we will give a brief description of most used techniques and we will present how related proposals have used them for preserving identities privacy in the blockchain.

1. Mixing services: This technique was mainly introduced to guarantee anonymity in blockchain based financial applications transactions by transferring payments from an input set of addresses to an output set in such a way that it is hard to trace which input address paid which output one which successfully confuse the trail back to the funds' original source.

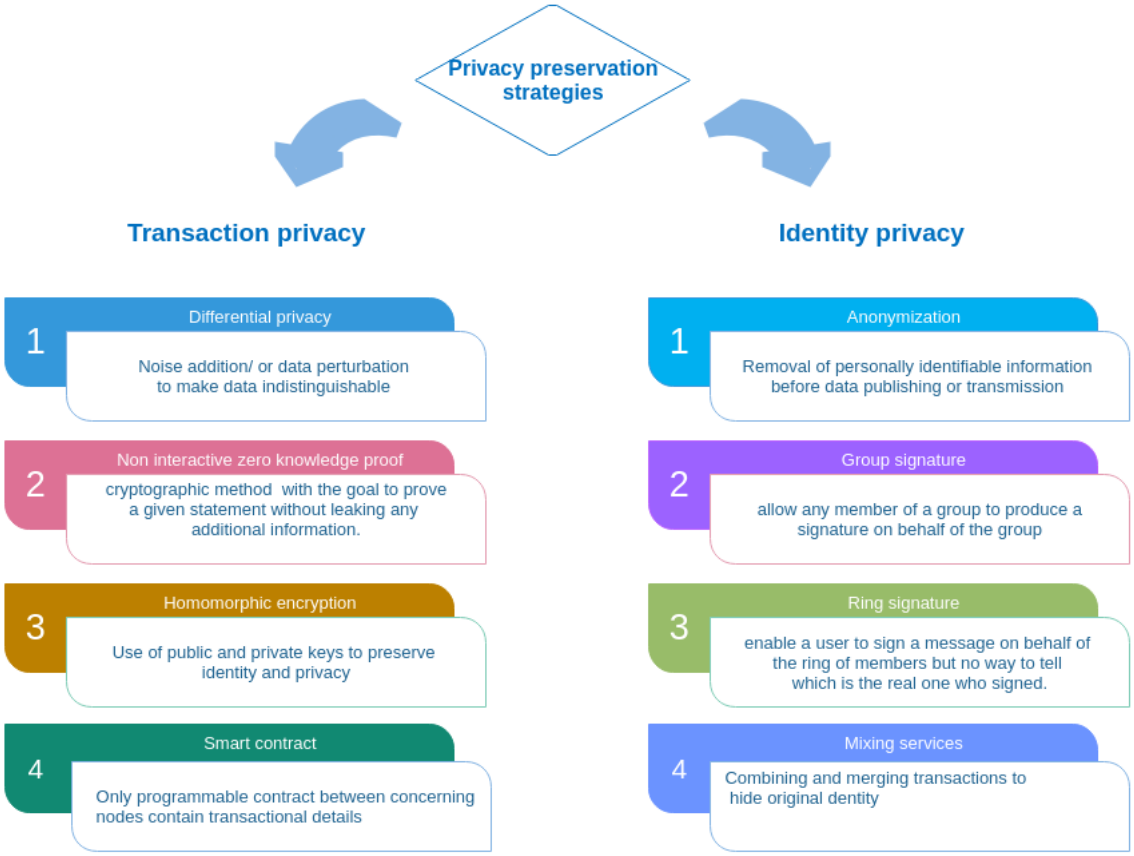

Figure 4.1 – Considered strategies for privacy preservation in blockchain

As illustrated in Fig. 4.2, a trusted third service called mixer is involved to intermix input transactions in order to hide the value and identities from adversaries in such a way that associating input and output transactions from these last' point of view is impossible.

In this context, most existing mixing services adopt a completely centralized architecture which may cause some problems insofar that the mixing server may keep records of mixing information and store the mapping from inputs to outputs transactions which may be leaked after mixing. Hence distributed mixing services have emerged to remedy centralized systems' limitations.

As a first example, CoinJoin was introduced in [112] to make a joint transaction mixing the link between inputs and outputs so that the exact direction of data flow will be kept unknown to the other peers. This protocol provides anonymity for users and ensures that funds will be transferred to their addresses, as they will check and validate the mixing transaction before signing on it.

As an amelioration to CoinJoin, CoinShuffle was proposed by [141] to achieve the internal unlinkability. It utilizes an anonymous group communication protocol to hide the participants' identities from each other. This method achieves the internal unlinkanility by the simple trick of layered encryption, and the cost of high communication and computation overhead.

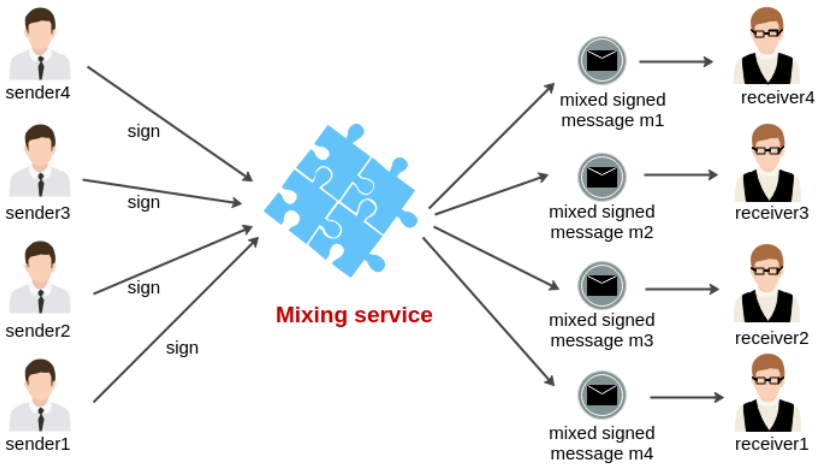

Figure 4.2 – mixing services scheme

2. Ring signature: This technique was initially designed by [139] as a digital signature that, by generating an anonymous signature, conceals the actual identity of a signer among a group of signers forming a ring which makes it a special form of group signature that excludes the group manager entity and where no particular group setup procedure is necessary. A ring signature is generated by one of ring members and it can be verified by anyone who owns all ring members' public keys. Nobody has the ability to identify who actually signed it. Unless the signer exposes himself, there is no mechanism for others to find out which one in the ring is the actual signer. All information a verifier can confirm is that someone included in the ring has generated the signature. For any verifiers, the identity of the signer is absolutely anonymous. It is very useful in some particular situations where identity information can not be revealed while secrets must be authenticated with signatures.
   As illustrated in Fig. 4.3, when willing to sign a transaction, the signer user aggregates a group of public keys (Pk1, Pk2,..,PkN) from other participants to generate a ring. All participants corresponding to these public keys are regarded as ring members. Therefore, a signature is generated using the aggregated keys PKi plus the signer's own private key SK. This last then has to announce all public keys of ring members for verifiers to verify this ring signature. In particular, other ring members may not be aware of a ring signature has been generated utilizing their public keys.
   As described, the ring signature scheme can be divided into 3 parts: key generation, ring signature generation and signature verification.
   In the current literature several works have used this scheme for preserving identity privacy and anonymity in blockchain transactions within several applications such as financial applications, e-voting systems and healthcare services.

In [58], authors presented a lightweight privacy-preserving scheme that guarantees anonymity and security of users during the sharing and the management of their medical data. The proposed scheme uses lightweight Ring structure along with digital signatures in order to allow participating patients to sign their transactions anonymously.
In [105], authors proposed a smart contract based decentralized trustless e-voting system where participating voters have cryptographic assurance ensuring the protection of their privacy. To meet the basic security needs of electronic voting systems and to hide the real identity of each voter while avoiding multiple votings, the proposed framework uses linkable ring signature for each voter to group a signing ring. Authors opted for such technique in order to guarantee double voting avoided insofar that public checkers could verify that whether two signatures on different messages are generated by the same signer.
In the context of financial applications, Ring confidential transactions was proposed in [126] as the combination result of Maxwell's approach of Confidential Transactions [111] with ring signatures in order to hide both sender and receiver addresses as well as the transaction amount information. The proposed solution can provide identity privacy and transaction privacy simultaneously. The proposed approach is implemented in Monero cryptocurrency system.

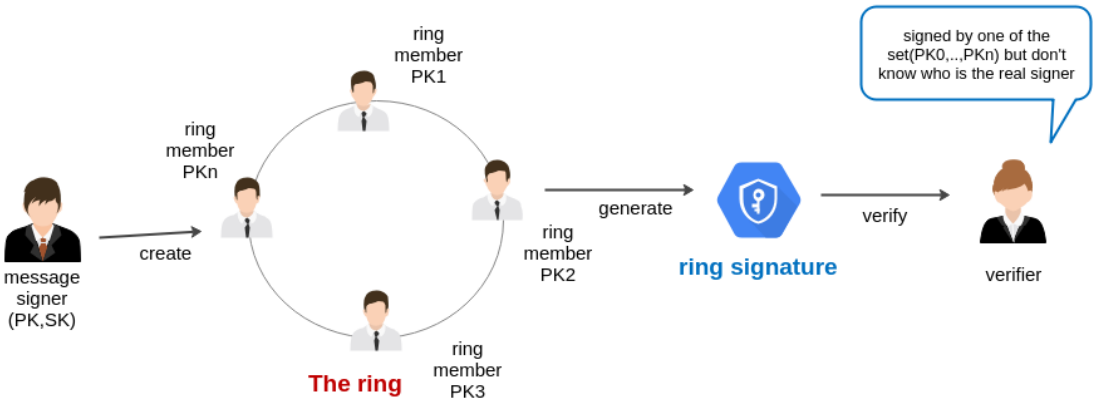

Figure 4.3 – Ring signature scheme

3. Group signature: This technique is mainly conceived to allow any member of a group to produce a signature on behalf of the group, enabling users to sign with the authority of the group, without revealing the specific signer's identity. The concept of a group signature, as illustrated in Fig. 4.4, is that a trusted group master or manager is responsible for setting up a group of users who can sign messages on behalf of the whole group, without revealing their individual identity. The group master holds a master key with the ability to reveal the signer of any

signature generated by a group member in the past. As a result of this, group signatures offer the participants anonymity only under the condition that the group master does not choose to reveal the signer's identity.

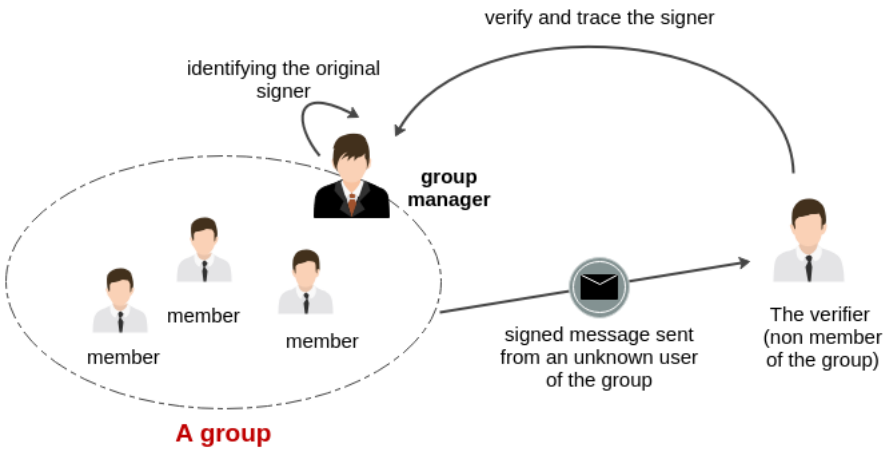

Figure 4.4 – Group signature scheme

4.2.2 Transaction based privacy preservation techniques

In the current blockchain environments, the issue is not only that data is permanently stored on a ledger, never to be erased or altered, but that by nature it exists on a blockchain which is irreversibly shared with the entire network what makes it easily accessible by each participating user in case of public blockchain or by authorized ones in case of private blockchain solutions which increases privacy concerns regarding exchanged transactions content. In order to achieve a higher level of transaction privacy in blockchain, three main approaches could be considered for securing transactions carried data and information. These last include: differential privacy, non interactive zero knowledge proof, homomorphic encryption and smart contracts as it is illustrated in Fig. 4.1. In the following we will give a brief description of most used techniques and we will present how related proposals have used them for preserving transaction privacy in the blockchain.

1. Non Interactive Zero Knowledge (NIZK) proof: Zero-knowledge proof (ZKP) is a cryptographic method by which one party called the prover can prove a given statement to another party known as the verifier without leaking any additional information. The essence of a ZKP method is that it is trivial to prove that someone possesses knowledge of certain information by simply revealing it. Hence the challenge of how to justify such possession without revealing the information itself or any additional information. To do so, ZKP must satisfy three main parameters called completeness, soundness and zero knowledge.

Non-interactive zero-knowledge proof (NIZK) is a variant of ZKP where the interaction between the prover and the verifier is missing to consist of just a single action in which the prover generates a proof chosen randomly from a set of questions to which it has always response and send it to the verifier in charge of checking if the prover knows the secret information to answer the question using the generated proof and another special function called check a proof, as it is illustrated in Fig 4.5.

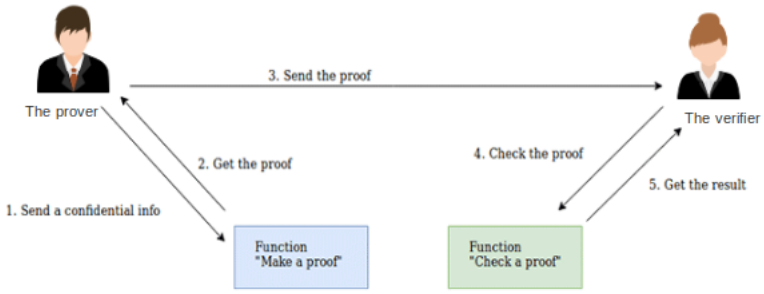

Figure 4.5 – Non Interactive Zero Knowledge proof scenario

The loss of interaction between the prover and the verifier in addition to its ability to prove the correctness of an assertion independently without leaking additional information makes NIZK proof well suitable in blockchain to verify transactions anonymously and in a distributed way and thus to create privacy preserving protocols.

We present hereafter some examples of its usage in preserving privacy of blockchain.

In [87], Hawk is presented as the first work to simultaneously provide transactional privacy and programmability in the blockchain. This method is based on the idea of Zerocash and the smart contract system, users send encrypted and committed information to the smart contract, and rely on the NIZK proofs to enforce the correctness of contract execution and funds' transfer. While the result of smart contract can be publicly verifiable, the entire sequence of transaction actions taken in the contract are kept confidential from the public.

In [162], authors proposed a language, zkay, that supports expressive privacy specifications allowing developers to specify data ownership by annotating variables as private to particular accounts. To enable running the proposed zkay contract on public blockchains, this last is transformed to a contract where values are encrypted for their owner and correctness is enforced using NIZK proofs, guaranteeing that transformed contracts preserve privacy and functionality.

In [148], a new cryptocurrency Zerocash protocol is issued to achieve anonymity and transaction privacy by making use of zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) proof and a commitment scheme to hide a

payment's original address. Furthermore, the coin value is added in the commitment and zero-knowledge proof is used so that the value is arbitrary and publicly verifiable.

2. Homomorphic encryption: this technique is a cryptographic encryption methodology that satisfies homomorphism so as to preserve arithmetic operations carried out on ciphertexts. Performing as a black box, it allows any party to perform operations on the ciphertexts while preserving the privacy of digital original data. This attractive feature makes homomorphic cryptography well suited for hiding and performing timely update of the amount and other metadata of a transaction. Typical homomorphic cryptographic schemes which could be used to protect privacy of blockchain include the Pedersen commitment scheme [133] and Paillier cryptosystem [131].
In this context, authors in [175] designed a framework where the Paillier cryptosystem is used to hide the real amount of each transaction, and Commitment Proof is used for checking the validity of the encrypted amount (i.e., ensures that the amount is positive and verifies the trade-off between inputs and outputs). These encrypted transactions are like sealed asset envelopes which can be merged, separated, or used while keeping the amount invisible.

4.2.3 Discussion and Problem statement

Table 4.1 summarizes the already presented works regarding the target they address, the category to which they belong, the focus they are interested at, the tool they use and the application scenario in which they are applied.
As seen, we note that there have been many efforts for protecting privacy in blockchain. These efforts focused mainly on the following aspects:

- Obfuscating transactions relationships to resist linking or tracing analysis.
- Hiding real identities of both the sender and the receiver via complicated cryptographic primitives.
- Blinding the transaction content whilst retaining the verifiability and computability.

Adopted strategies for preserving identities and transactions privacy can be adapted to different requirements of a range of application areas such as financial applications, e-voting systems and healthcare services. However, just few studies have focused on using privacy preservation approaches to meet the requirements of blockchain based access control systems in the context of smart factories environments. In this context, just one work [157] proposed a privacy aware decentralized access control framework based on the Tangle technology. The proposed work provides privacy of the policy

stored within the Tangle by leveraging Masked Authenticated Messaging (MAM) data communication protocol that allows users to transmit policies in the encrypted format for storage.
In this direction, two main limitations need to be addressed alongside this study:

- Pseudonymity: here blockchain anonymity is mainly guaranteed by allocating some addresses generated from a one-time public key (e.g. Ethereum address) to participating devices, and clearly it will be challenging to identify a specific one in a large real-world infrastructure such as a smart factory environment. However when considering some techniques such as transaction graph analysis and quantitative analysis, il will be possible to find the connection between an address and it's concrete entity. Once an adversary can link the allocated address to a specific entity, all of its access requests and procedures related records will be disclosed; hence, compromising the device's identity and personal privacy.

- Traceability: while blockchain transactions serve as verifiable and traceable medium of access request procedures, it may leak access requesters privacy and sensitive information, insofar that their access history related details once recorded in the blockchain, may conduct to learning their authorization functionality pattern, as well as to tracing their access activities.

- Policy public visibility: here Blockchain based solutions generally require the submission of access control policies directly into the blockchain to ensure verifiable-consistency, immutability and notarization. Unfortunately, this will also reveal all the access control policies to the public, meaning that anyone can learn the required policies to access resources even when they are not authorized. This will further leak entities sensitive information beyond the inferred metadata from accessible data.

Work	Addressed target	Category	Focus	Considered technique	Application scenario
[112]	Bitcoin-like blockchain	Identity privacy	combining multiple Bitcoin payments from different senders into a single transaction so that data flow exact trx is kept unknown	Mixing services	Financial transactions
[58]	Ethereum smart contracts	Identity and Transaction privacy	allowing participating patients to sign their transactions anonymously while sharing their encrypted data.	Ring signature Double encryption using ARX ciphers and public encryption schemes.	Healthcare
[105]	Ethereum smart contracts	Identity and Transaction privacy	using cryptographic assurance to ensure the protection of voters real identity while avoiding multiple votings.	Linkable ring signature	E-voting
[126]	Monero cryptocurrency	Identity and Transaction privacy	preserving anonymity and preventing double spending by combining Confidential Transactions with ring signatures.	Ring signature	Financial applications
[87]	A developed Smart contract	Transaction privacy	applying smart contract to store the encrypted committed coins generated by users then NIZK proofs are used to enforce the correctness of contract execution.	zkSNARK library Libsnark	Financial transactions
[162]	Zkay smart contracts	Transaction privacy	allowing developers to specify data ownership by annotating variables as private to particular accounts	ZoKrates library	
[175]	Bitcoin Smart contracts	Transaction privacy	Hiding the real amount exchanged within transactions through homomorphic cryptogtraphy	Paillier cryptosystem Commitment Proof	Financial transactions

Table 4.1 – Security rules list example

4.3 Proposed Approach

4.3.1 Main objectives

In blockchain based access control schemes, it is necessary to protect information security and preserve attributes privacy of the participating role during the access procedure. In this direction our main objectives in this work are as follows:

1. Untraceability with the goal to protect the access requester related information in the form of the address (public key). A participating entity here and on the basis of access records and history, cannot be able to trace who has demanded the access and within which condition this last is attributed.
2. Unlinkability where transactions identities should be anonymous in the sense that it reveals nothing about the real identity of their issuers.
3. Confidential policies where sensitive access control policies should be revealed just to authorized parties.

4.3.2 Overview

To achieve such objectives, we introduce along this work an anonymous privacy-preserving distributed access management framework called PDAMF, that we propose to integrate within our DRMF framework in order to ensure strong privacy guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies as it is illustrated in Fig. 4.6.

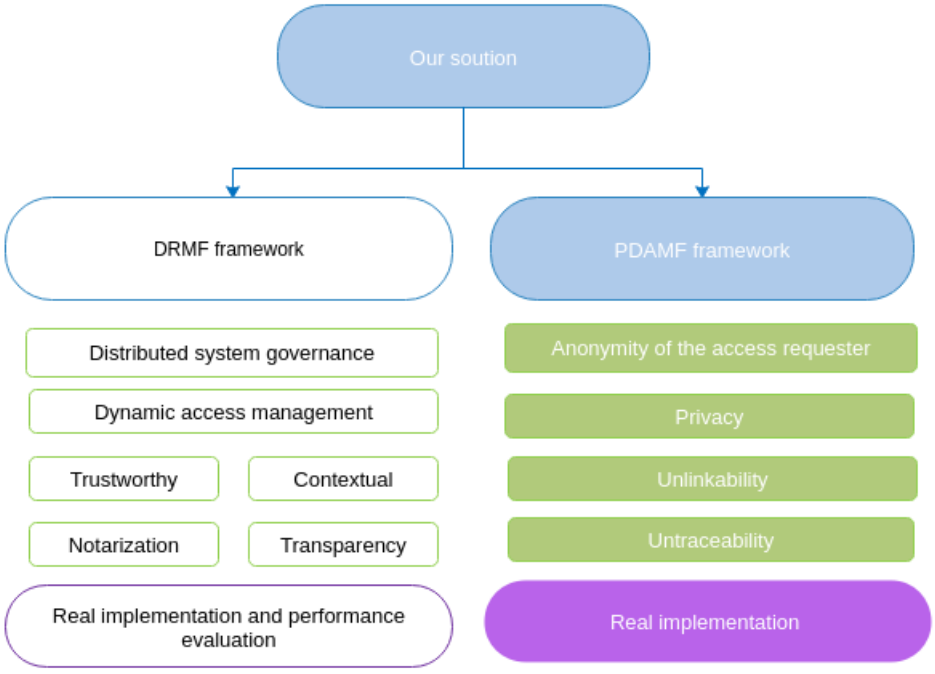

Figure 4.6 – Architectural view of PDAMF

To tackle this issues, we propose to integrate anonymous signature schemes within blockchain smart contracts while keeping the verification process transparent and notarized.
Our main focus in this work lies on ring signatures to preserve access requesters privacy and enhance their anonymity regarding the unlinkability of their real identites to their authorization tokens and on the other side the intraceability of their access history on the basis of their authorization tokens. By relying on such scheme, our framework could provide privacy and verifiability at the same time.
One benefit of using ring signatures over other anonymizing techniques presented in 4.2.1 especially group signature and mixing services mainly refer to the fact that these last need some sort of centralized trusted entity for combining transactions (in case of mixing services) or managing group members (including revealing their real identities in case of group signatures). Reminding that in case the trusted part is compromised, the anonymity of the transaction will also be compromised. When it comes to zero knowledge proof, this last incurs high computation overheads specially in the proof generation phase which may be unfeasible for our use case.

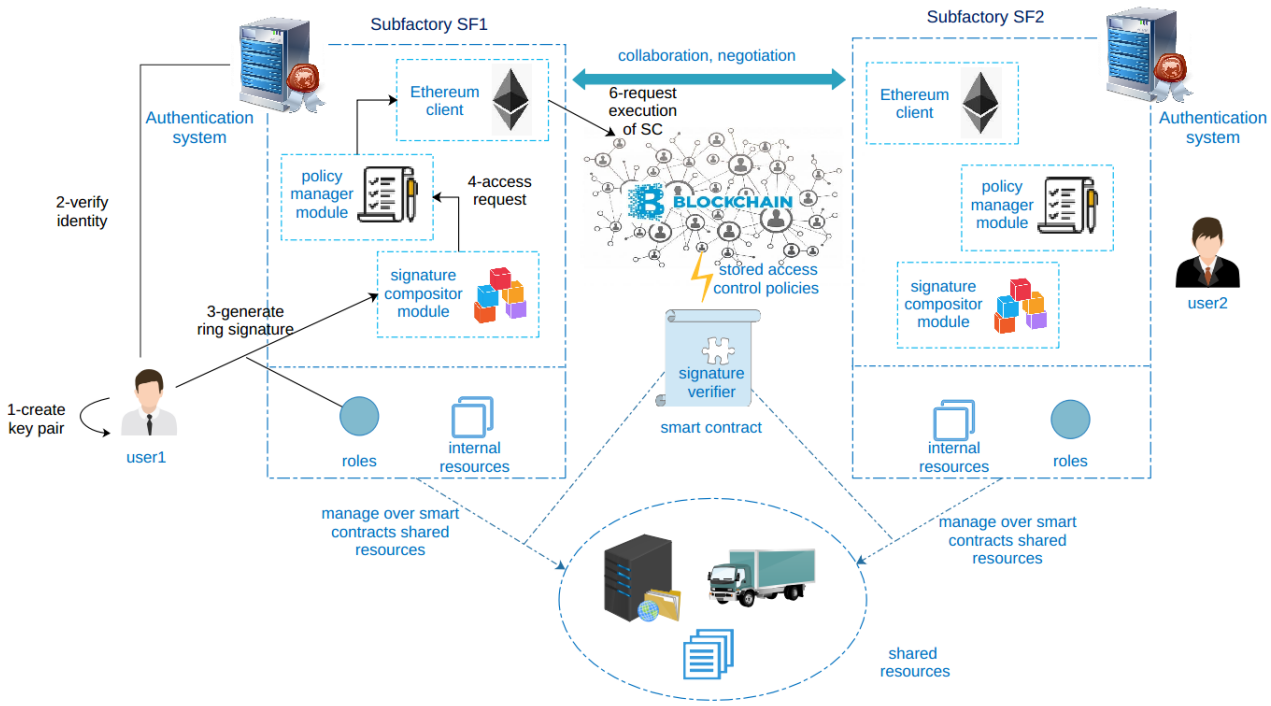

Figure 4.7 – Overview of the proposed scheme

Fig. 4.7 shows the overall structure of the proposed system according to the use case presented in the previous chapter. The vision of our privacy-preserving distributed access management framework is a system of autonomous organizations and domains (i.e. subfactories, customers, suppliers, transportation partners) collaborating with each

other while exchanging products and sharing resources in an attempt to realize a specific goal. The access management over shared resources plus the governance of the overall system is based on specific rules defined through smart contracts created and deployed within the blockchain network acting hereof as a policy retrieval point where all security rules are stored in form of transactions. Alongside, each domain holds its own entities with different roles. These last when wishing to have access over the shared resources, are asked to pass through smart contracts defining the set of rules and conditions that need to be fulfilled in order to get access authorization. We remind here that managing the access via smart contracts deployed within a publicly visible and transparent ledger will reveal requester entities sensitive attributes what could violate their privacy and anonymity. Hence our idea to use ring signature scheme for preserving identity privacy. That is, nodes within the P2P network can decide whether Adam, a shipping agent, having sending an access request to use the shipping truck resource, satisfy the security rule, without learning any other information about his personal attributes or without being able to trace his action throughout the whole process. Instead, by using ring signature, they will be able to trace Adam's corresponding role and no more his real identity. In this way the anonymity and untraceability of the token he uses is achieved.

4.3.3 System composition

The proposed framework and as it is illustrated in Fig. 4.7 consists of the following entities:

- Participating users (Ui) demanding access authorization over shared resources (SR) using attributed roles (Ri) and by means of a ring signature RS generated by use of a randomly selected subset of public keys P1, P2,.., Pr of the r ring members, together with the secret key Ss of the s-th member representing the user who has signed the access demand transaction.

- Authentication system: This component is mainly responsible for verifying the validity of entities' identities as well as the legitimacy of demands and requests sent to the blockchain network. Participating entities are authenticated based on the provided credentials. These last are used to prove the authority and legitimacy of public keys. In exceptional situations, as for example in case of a malicious behavior launched by a certain user after having access granted causing consequently resource damage, confidential data stealing, or even resources and devices tracking, the real identity of the corresponding signature real originator could be revealed.

- Peer to peer network: This component is the main core of our system. It defines: (1) a set of smart contracts ensuring the access management and the verification

of generated ring signatures, (2) a distributed network composed by a set of peers responsible mainly for receiving the established transactions, verifying and validating its state and executing the functions contained within.

- Signature compositor module: This component is mainly responsible for generating a ring signature from the ring of participants without revealing the identity of the signature's producer. The generated signature is then used for signing access authorization transactions.

- Policy manager module: This component serves in the one side as the defining part of access rules to be encapsulated into transactions and reloaded to the blockchain after validation, on the other side this module serves as the acquisition source of attribute values required for policy evaluation that once received and intercepted will be sent to the corresponding smart contract.

- Ethereum client: This component implements the full functionality required to join and to participate in the distributed network. This handles a broad set of tasks, such as connecting to the peer-to-peer network, encoding and sending transactions, keeping and exploring blocks copies and deploying and interacting with smart contracts.

Let us assume that a human worker denoted as user1 within the subfactory SF1 wants to have access over the shared resource Truck 3 to perform a shipping operation. To do so, the process would work as follows: the agent needs to get authenticated first along with the authentication system, which is responsible for making authorization decisions, verifying devices identities and generating authorization tokens. Furthermore, the access requester will generate his address corresponding to a pair of key (Pk_{user1}, Sk_{user1}). The public key Pk_{user1} is published in order to enable other participating entities to interact with user1. The secret key Sk_{user1} is used to spend the authorization token to have access allowed. Therefore, the signature compositor module will query the set of all other registered users in the contract, retrieved public keys are used for forming the ring and generating the ring signature. Created signature is therefore used to sign the access request transaction formulated by the policymanager module and broadcast via the Ethereum client to the peer to peer network where the corresponding smart contract shall be successfully executed to have access allowed and get an authorization token. To do so, the transaction signature is verified by the smart contract, if this last is valid, and the access request transaction matches the defined access control policy, this last is accepted by the ledger and the access is granted.

4.3.4 System operational blocks

The proposed framework divides the interaction between involved parties into five operational blocks as follows: a set up phase, a security rules definition phase, a key generation phase, an access request phase, and an access control phase. A description of these different phases and operations is presented in Fig. 4.8.
These five phases our proposed framework is made up of are detailed in the next paragraphs.

System set up

Once the distributed network is created and becomes operational, a consortium is defined in order to reach the desired agreement among collaborating parties that once identified get registered and involved in the consensus procedure. Therefore and given the existence of resources to be shared, these last are identified in order to be accessed and used in a notarized manner. Subsequently, the process of adding new entities with different roles and classifications can be launched. Here, it should be assumed that new entities have received an Ethereum address required to partake in the Ethereum network. Therefore, the process of adding a new entity begins by having the pool of consensus entities validate that the public identifier corresponds to the pretended role and suits the requested classification.

Security rules definition

When collaborating parties agree on adding an access control policy for a newly deployed resource to be shared among them and used as well by external users, an access policy needs to be added to the Access smart contract and shared within the distributed network. An access policy here is a set of rules and conditions based on a specific context and attributes that a requester entity has to fulfill in order to obtain access authorization over specific resource. These rules are expressed in OrBAC access control model and then encapsulated by the policy manager module in form of a scripting language in order to be added to the policies list within the Access smart contract. At this stage, the Ethereum client broadcasts the SrD transaction to the peer to peer network, this verifies, validates the transaction, and includes it into the blockchain if it was valid else it will be rejected and a notification will be sent.

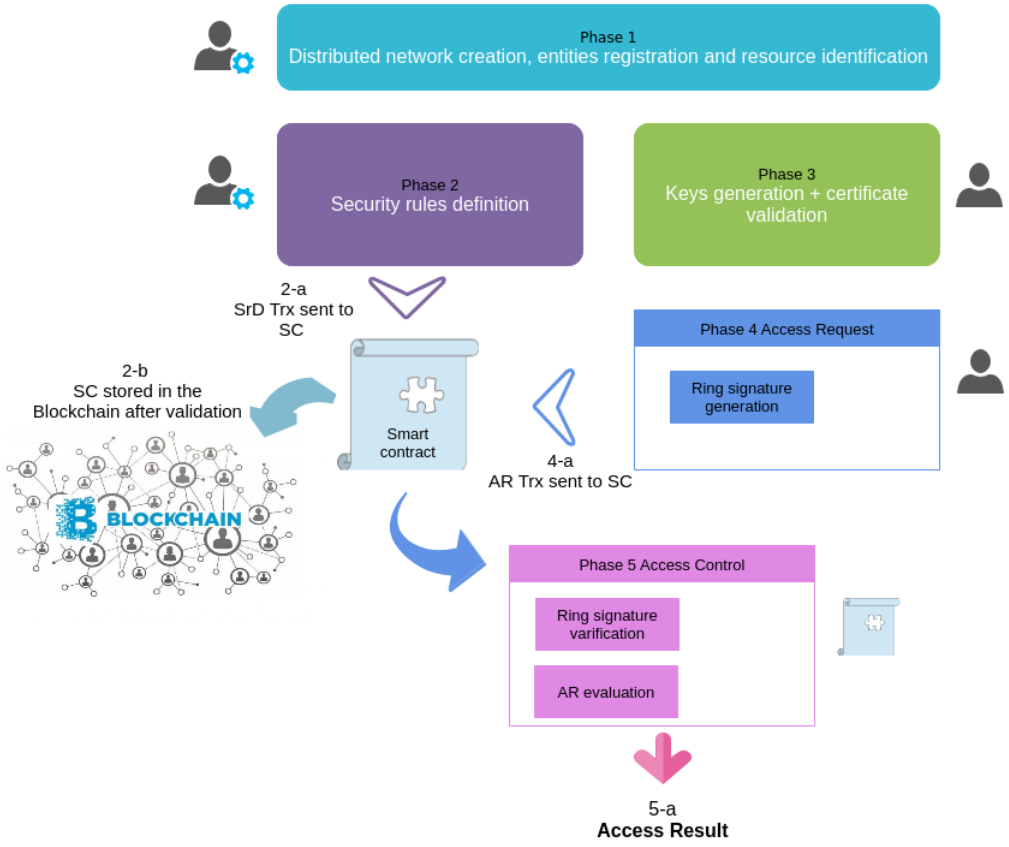

Figure 4.8 – System operational blocks

Key generation

The access requester engage alongside this stage in the keyGen algorithm that takes as input public parameter *pp* to output a signing key pair *(Pki,Ski)*, the generated public key in addition to identity descriptors are then sent to the authentication system that will generate a certificate based on the parameters passed, calculate a hash value over it and sign it using its private key. As a result, a digital certificate is created and sent back to the access requester. This last is mainly used to prove the authority and legitimacy of public keys. In exceptional situations, as for example in case of a malicious behavior launched by a certain user after having access granted causing consequently resource damage, confidential data stealing, or even resources and devices tracking, the real identity of the corresponding signature real originator could be revealed by the authentication system managing related keys.
The key generation algorithm is as follow:
We use hereafter the notation $\leftarrow_R$ to indicate choosing an element at random from a set, for example $t_j \leftarrow_R \mathbb{Z}_q$ shows t_j chosen at random from $\mathbb{Z}_q$.
1- For λ the security parameter, choose multiplicative group G with prime order q, and randomly chosen generator g of G.
2- Choose also two hash functions H and H_0 such that:

- $H : \{0,1\}_* \rightarrow G$
- $H_0 : \{0,1\}_* \rightarrow \mathbb{Z}_q$

3- Output public parameters pp = (λ, q, G, H, H_0).
4- run KeyGen algorithm $KeyGen(1^\lambda, pp)$ as follow:

- $x_i \leftarrow_R \mathbb{Z}_q$
- $y_i \leftarrow g^{x_i}$

5- Output the public key $Pki = (pp, y_i)$ and the secret key $Ski = (pp, x_i)$.

Access Request

When willing to perform an action over an existing shared resource, the requesting entity sends an access request transaction that further to which a decision is made after a successful execution of the Access Smart Contract.
To do so, the Access requester (ARq) runs in a first step the ring signature generation algorithm $RingGen(Sk_i, Ring, AR_Tx)$ as follow, where $Ring$ is the set of public keys Pk_i having participating in the ring creation, $Ring = (Pk_1, Pk_2, .., Pk_n)$:
If j in $Ring$ members and $j \neq i$ then:
compute $t_j, c_j \leftarrow_R \mathbb{Z}_q$,
compute $a_j \leftarrow g^{t_j} y_j^{c_j}$,
compute $b_j \leftarrow H(AR_Tx \parallel Ring)^{t_j} (H(AR_Trx \parallel Ring)_i^x)^{c_j}$,
endif
If $j = i$ then:
compute $r_i \leftarrow_R \mathbb{Z}_q$, compute $a_i \leftarrow g^{t_j} y_j^{c_j}$,
compute $b_i \leftarrow H(AR_Tx \parallel Ring)^{r_i}$,
endif
Calculate $c_i \leftarrow [H_0(AR_Tx, Ring, \{a_j, b_j\}_1^n) - \sum_{j \neq i} c_j] \; mod \; q$,
Calculate $t_i \leftarrow r_i - c_i x_i \; mod \; q$,
Return ($Ring$, AR_Tx, $H(AR_Tx \parallel Ring)^{x_i}$, c_i, t_i)
In a second step and after getting the address of the Smart Contract managing access to the shared resource and signing the Access request on behalf of the ring, the access requester triggers the Smart Contract with a Request Access transaction AR_Tx.

Access control

Once receiving the Request Access transaction AC_Tx, the Access smart contract (AC) will verify first the transaction signature as follows:
Parsing the output of $RingGen$ algorithm , and using the notation $H(AR_Tx \parallel Ring)^{x_i} = \sigma$ we compare:

$$\sum_1^n c_j = H_0(AR_Tx, Ring, \{g^{t_j} y_j^{c_j}, H(AR_Tx \parallel Ring)^{t_j} \sigma^{c_j}\}_1^n)$$

If the comparison is true, the signature will be accepted and validated, otherwise, it will be rejected.
As a second step of this phase, the Access contract will evaluate the access demand by verifying the validity of the subject role and the existence of the access policy within the defined policies list, as a result it determines whether the request should be permitted or denied.

4.4 Implementation and Analysis

In this section, we will introduce first the different tools used for the implementation of our framework therefore we will discuss expected properties and aspects and we will show experiment results demonstrating its feasibility.

4.4.1 Work environment

For the implementation of our proposal, we have used the testbed presented in Chapter 3 and illustrated in Fig. 4.9 featuring hardware and software components listed below:

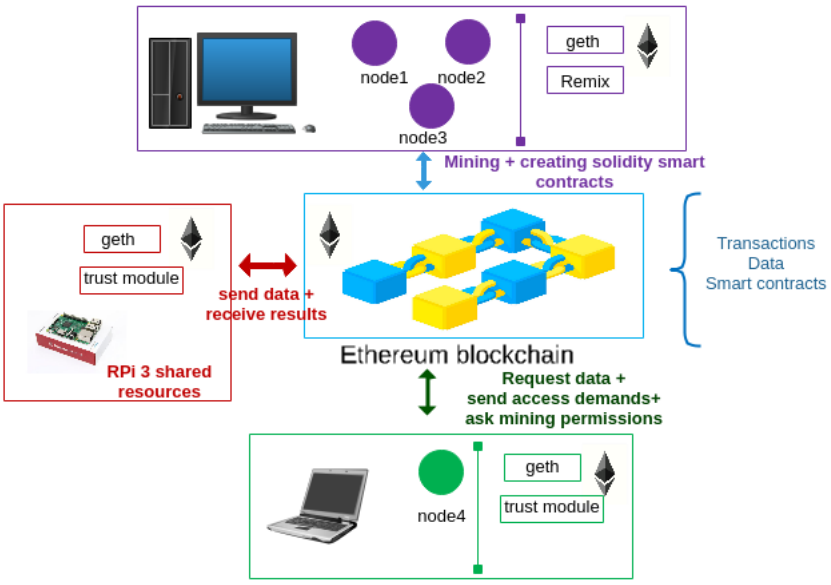

Figure 4.9 – Work environment

- a Dell Precision M6800 machine with 4th Generation Intel Core i7 processor and 8Gb of RAM in which we have configured a private Ethereum blockchain network consisting of three nodes having the functionalities of Ethereum miners.

- an Intel Core i5-3210M laptop with 6 Gb of RAM and 2.40 GHz of CPU frequency in which an Ethereum node was set up.
- two Raspberry Pi 3 Model B configured to act as shipping resources shared among the collaborating factories.

For the implementation of our proposal, we chose Ethereum, which is currently the most common blockchain platform for developing smart contracts. Specifically we used geth client [66] configured on each entity so that it could act as an Ethereum node.
For the implementation of the ring signature scheme, we used the secp256k1 scheme for creating elliptic curve parameters and producing the signature where ECDSA is performed [152]. We chose such scheme for its particular structure that allows very fast performance when implementing elliptic curves' points addition and multiplication by a scalar.

4.4.2 Performance evaluation

Based on the software and the hardware we presented in the previous section, we carried out experiments to prove the feasibility of our proposed privacy aware access management framework. Such framework could protect the privacy of access requester entities and ensure the security and the anonymity of their identities in blockchain applications which is a long awaited feature that was missing in the current blockchain based access control related literature. In this direction, a ring signature primitive has been introduced to face the traceability problem of authorization tokens. Considering such primitive, a participating entity on the basis of access records and history, cannot be able to trace who has demanded the access and within which condition this last is attributed.
To demonstrate the feasibility of our proposal in achieving distributed and privacy aware access control for Industry 4.0 applications, we will evaluate in the following the performance time required for signing and verifying the generated ring signature while varying the number of participants involved in the ring construction process. This time is therefore compared to the performance time required for executing the entire access control related transaction.
In the case study we presented, corresponding results have been illustrated in Table 4.2. This last enumerates the performance time for ring signature related processes as well as the generated signature size.

Ring members	Signature generation time (s)	Signature verification time (s)	Signature size (Byte)
1	0.061	0.002	340
5	0.097	0.008	1405
10	0.102	0.01	3120
20	0.113	0.018	6113
50	0.125	0.022	15000

Table 4.2 – Ring signature performance

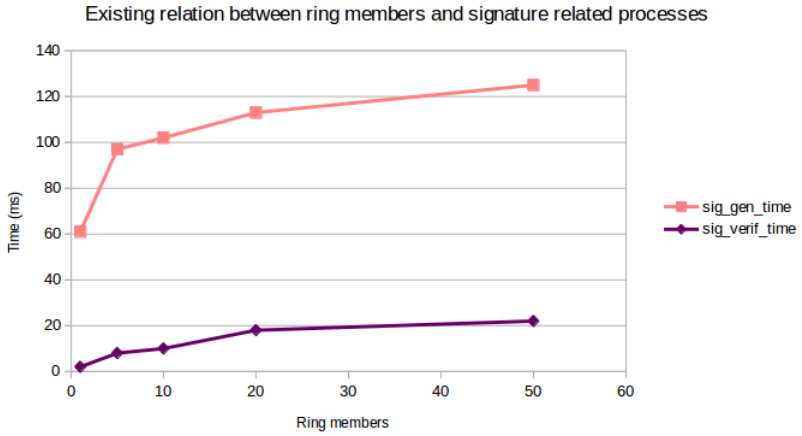

Figure 4.10 – Existing relation between ring members and signature generation and verification processes

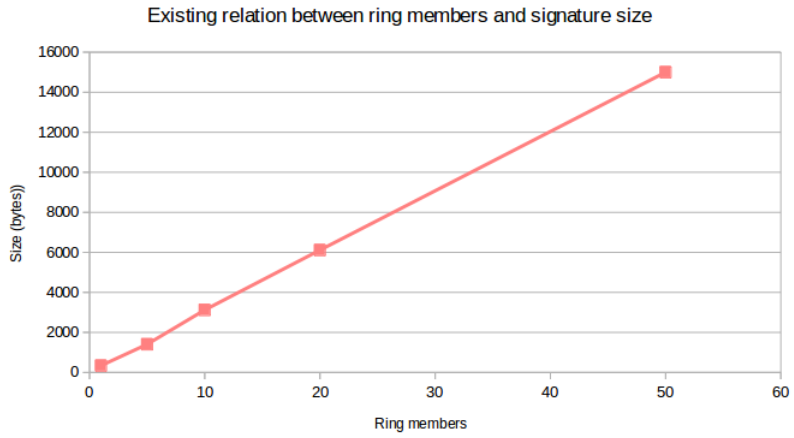

Figure 4.11 – Existing relation between ring members and signature size

Fig. 4.10 and fig. 4.11 present the existing relation between ring participants number and the time required for performing ring signature related processes plus the signature size. According to these figures we remark that there is a linear relationship between public keys number corresponding to the ring members and other parameters such as

the signature generation or verification time and the signature size. The more participants enroll in, the less the process is efficient in terms of performance time. When comparing this time to the performance time required for executing the Access smart contract for an entire access control transaction which is less than 25 seconds (24.8 s) for a 10 members ring size, we can demonstrate the possibility of our framework in achieving distributed privacy aware access control.
Notice that the time required for deploying and running smart contracts depends on various factors, like the system computing power, the system networking architecture. Thus, in the real-world public Ethereum system, the time cost may differ significantly from that in this case study.

Conclusion

In this chapter we have presented our approach to tackle privacy and security issues in decentralized access control mechanisms based on the blockchain technology. More specifically, our focus was to ensure strong privacy guarantees over the access control related procedures regarding the access requester personal information and sensitive attributes. To do so we have built a privacy aware distributed access management framework based on the ring signature on the elliptic curve, and used the complete anonymity of the ring signature to ensure the security and the anonymity of access requesters identities in blockchain applications. By such a way, the transparency feature of our framework is maintained while guaranteeing and preserving the privacy of its users.
Our evaluation regarding the performance time needed for enabling such scheme shows that our proposal is feasible, deployable and suited for our use case environment. For future work, we plan to generalize our privacy aware framework to consider the privacy of shared access control policies within the blockchain distributed ledger where they would be revealed just to authorized parties.

Chapter 5

An Event-B based Approach for Formal Modeling and Verification of Smart Contracts

Contents

5.1 INTRODUCTION

While smart contracts are becoming widely recognized as the most successful application of the blockchain technology that could be applied into various industries and for different purposes such as e-commerce, energy tradings, assets management, and healthcare services, their implementation has posed several challenges insofar that they could handle large amount of money and digital assets in addition to their ability to manipulate critical data and transactions related information which makes them attractive targets of security threats and attacks that could lead to significant problems like money losses, privacy leakage and data breach. To better deal with such issues, reasoning about the correctness, the safety and the functional accuracy of smart contracts before their deployment on the blockchain network is critical and no important than ever. In this context model checking tools are well adopted for the formal verification of smart contracts in order to assure their execution as parties' willingness as well as their reliable and secure interaction with users. In this direction, this chapter will focus on behavior based formal verification of smart contracts in order to verify their compliance with the specification for given behaviors. The verification is conducted using Event-B formal verification method in addition to a model checking tool along which expected safety properties are formalized, validated and judged to be satisfied or unsatisfied. The described approach of this proposal is applied to the use case application presented in Chapter 3 where implemented smart contracts are formally modeled and verified using the model checking tool.
The rest of this chapter is organized as follows. Section 2 recalls the basic preliminaries and background needed for the understanding of our proposal described in the remainder of this chapter. Section 3 discusses then smart contracts security issues and vulnerabilities and presents related proposals carried out in the area of smart contracts formal verification. Thereafter Section 4 describes the proposed approach from a theoretical and formal point of view and details the proposed Event-B formalization followed by the verification process of smart contract's behavior in Section 5. Finally in Section 6, the chapter ends up with some conclusions and an outlook of our future work to study in this area.

5.2 Background and problem statement

In this section, we will introduce first the main preliminaries and background needed for understanding of our proposal specifically formal verification methods and Event-B formal method. We will point out then potential security issues that smart contracts may have and we will review research proposals related to their security verification in recent years.

5.2.1 Event-B formal method

Event-B [4] is both a language and a method for formal specification and verification of secure systems. It has been proposed by J-R Abrial as the successor of the classic B Method [5]. Event-B has preserved the advantage and the simplicity of the B method while making improvements in several aspects, including the specification of reactive systems. Following the B Method, Event-B uses basic mathematical notations, first order logic and set theory. It supports a large part of the development life cycle, from the specification/design phase to the implementation phase. This allows the early errors detection which prevents from execution errors and facilitates maintenance.
The complexity of a system is mastered thanks to the refinement concept allowing to gradually introduce the different parts that constitute the system starting from an abstract specification to a more concrete one. The abstract specification describes the fundamental properties of the system. Requirements and details are added incrementally through the refinement process.

Machines and Contexts

An Event-B specification is made of two main elements: contexts and machines. The machine is a fundamental component for the formal construction of a system in Event-B. It specifies its dynamic part and includes several elements such as variables V , invariants Inv, and events E that establish the system state change. The variables define the state of the system to be specified. The possible values that the variables hold are restricted using invariants written using first-order predicates. Invariants should remain valid in each state of the system. Thus, they should be valid in the initial state and after the execution of each event.
Machines often need static elements of the system such as constants C, sets S, and axioms A that specify their properties. These elements are included in a context that describes the static part of an Event-B specification. To have access to its elements, a context is seen by a machine (i.e. SEES Context). An event can be executed if it is enabled, i.e. all the conditions G, named guards, prior to its execution hold. Among all enabled events, only one is executed. In this case, substitutions Act, called actions, are applied over variables. In this thesis, we restrict ourselves to the becomes equal substitution, denoted by (x := e).

Refinement in Event-B

Refinement is a process of enriching or modifying a model in order to augment the functionality being modeled, or/and explain how some purposes are achieved. Both Event-B elements context and machine can be refined. A context can be extended by defining new sets Sr and/or constants Cr together with new axioms Ar. A machine is refined by adding new variables and/or replacing existing variables by new ones Vr

that are typed with an additional invariant Invr. New events can also be introduced to implicitly refine a skip event (i.e. It does not modify the already existing variables).

Verification and Validation of Event-B Models

Event-B is supported by the Atelier-B platform [46] and can be used in conjunction with the Pro-B animator/model-checker [92, 91] in order to animate and validate a formal development.
ProB is an animator and explicit automatic model checker, originally developed for the verification and validation of software development based on the B language. This tool implements an automatic model checking technique to check LTL (linear temporal logic) [135] and CTL (Computational Tree Logic) [44] properties against a B specification. The core of ProB is written in a logical programming language called Prolog. Its purpose is to be a comprehensive tool in the area of formal verification methods. Its main functionalities can be summarized up as follows:

1. ProB can find a sequence of operations that, starting from a valid initial state of the machine, moves the machine into a state that violates its invariant,

2. Giving a valid state, ProB can exhibit the operation that make the invariant violated,

3. ProB allows the animation of the B/EventB specification to permit the user to play different scenarios from a given starting state that satisfies the invariant. Through a graphical user interface implemented in Tcl/Tk, the animator provides the user with: (i) the current state, (ii) the history of the operation executions that has led to the current state and (iii) a list of all enabled operations, along with proper argument instantiations. By this way, the user does not have to guess the right values for the operation arguments.

4. ProB supports the model checking of the LTL and CTL assertions.

5.2.2 Problem statement

Potential security issues and assurance of smart contracts

Besides their correct execution, it is also crucial that the design and the implementation of smart contracts are secure against vulnerabilities aiming at tampering and stealing assets they handle. Indeed, several security vulnerabilities in Ethereum smart contracts have been discovered. A recent analysis reveals that among 19336 smart contracts deployed on the public Ethereum blockchain, 8333 contracts suffer from at least one security issue [104]. An example of attack was in June 2016, the DAO (the world's largest

crowdfunding project deployed on the Ethereum) was attacked by hackers, causing more than 3 million ETH separated from the DAO resources pool which is worth around $60 million.

A survey of possible attacks on Ethereum contracts was presented in [16] where security vulnerabilities of smart contracts were grouped into three classes according to the level in which they are introduced namely Solidity, EVM bytecode, and blockchain.

Another issue that smart contracts may have is their dependency on external calls especially when they execute external contracts' codes within their functions, call other contracts' function and wait for its returned value, or even call another contract that may change its global state. If there is an exception raised (e.g., not enough gas, exceeding call stack limit) in the called contract, the calling contract terminates, reverts its state and returns false. However, depending on how the call is made, the exception in the calling contract may or may not get propagated. In these cases, the contract's control flow should not be influenced by an adversary contract.

To deal with such issues, reasoning about the correctness of smart contracts before their deployment on the blockchain network is critical and no important than ever. How to write reliable smart contracts was presented in [98], where two aspects were considered for the correctness verification and security insurance of smart contracts including programming correctness and formal verification.

Formal verification provides a powerful technology for the correctness verification of the established specification of smart contracts. In the following subsection, we will review research proposals carried out in this field.

Proposal	Target	Category	Focus	Tool
[28]	Solidity SC and EVM bytecode	Program based formal verification	Translated SC into F* to check the correctness.	F*
[13]	EVM bytecode	Program based formal verification	Analyzed EVM bytecode of contracts statically.	Isabelle/HOL
[1]	Solidity SC	Behavior based formal verification	Considered the interaction between users, programs and the environment.	Statistical Model Checking
[29]	DSCP BITHALO	Behavior based formal verification	Used probabilistic formal models for verification.	Game theory Markov Decision Process Prism tool
[123]	Solidity SC	Behavior based formal verification	Proposed a generic modeling method then applied the model checking.	NUSMV Model checking
[81]	LLVM bitcode supports different high level languages	Program based formal verification	Proposed a source code translator to convert the smart contract embedded with policy assertions to LLVM bitcode and a verifier to determine assertion violations.	Zeus (the prototype implemented)

Table 5.1 – Smart contracts based formal verification approaches

Related works

Multiple efforts have been carried out in the current literature for the correctness verification of the established specification of smart contracts through formal verification approaches.
In [28], authors proposed a verification method based on programming language. They translated Solidity written smart contracts into an F* language (functional programming language aimed at program verification) in order to check the safety, and the functional accuracy of implemented contracts. The translation is made both at the source level

(functional correction specifications) and bytecode level (low-level properties).
In [13], the Isabelle proof assistant was used to verify the binary Ethereum code whose corresponding sequences were organized into linear code blocks. A logic program was then created where each block is processed as a set of instructions. Each part of the verification is validated in a single trusted logical framework from the perspective of bytecode.
In [1], a new verification method is proposed to verify a smart contract's behavior in its execution environment. The proposed approach considered the interaction between both users and programs, plus programs and the environment. The Behavior Interaction Priorities (BIP) framework was used for components modeling, therefore the Statistical Model Checking (SMC) tool was used for verification.
In [29], game theory approach was combined with formal methods to address the challenging aspect of smart contracts, the proposed approach focused on DSCP (a Decentralized Smart Contract Protocol inspired by BITHALO), therefore a probabilistic formal model was proposed to verify smart contracts based on PRISM tool.
In [123], authors proposed a generic modeling method of smart contracts based Ethereum applications, the model checking approach was then considered to verify the implementation's compliance with the specification. The proposed model was written in NuSMV language with properties formalized into temporal logic CTL to be subsequently applied to a case study based on smart contracts and coming from the energy market place.
In [81], a framework called ZEUS was proposed to automatically verify the correctness of implemented smart contracts using both abstract interpretation and symbolic model checking and to validate their fairness. The proposed framework consists of three components including a policy builder against which the smart contract must be verified, a source code translator for the conversion of this last to LLVM bitcode, and third a verifier in charge of determining and reporting policy violations.
Table 5.1 summarizes the already presented works regarding the target they address, the category to which they belong, the focus they are interested at and the tool they use.

5.3 Proposed Approach

5.3.1 Overview

The main objective of this work is to formally model an Ethereum blockchain application based on smart contracts formal verification methods. We rely alongside this work on Event-B which is a state based model-oriented formal method intended for system development. Its strength lies in a well-defined modelling and analysis process, which allows one to specify a system from an abstract specification to a concrete one.
In order to verify and validate their safety, correctness and functional accuracy, smart

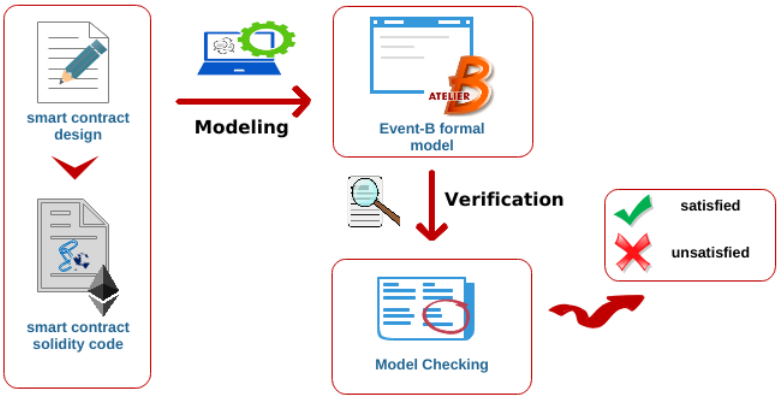

Figure 5.1 – Overview

contracts, as illustrated in Fig. 5.1, are translated from solidity (which is a high level programming language designed for implementing smart contracts running on the Ethereum blockchain), into Event-B formal language. Therefore, and in order to check whether implemented smart contracts behave as they are supposed to do, a verification of the formal model is conducted using a model checking tool along which expected safety properties are formalized, validated and judged to be satisfied or unsatisfied.

5.3.2 AC Event-B formal model

In Listing 1, the set of enumerated types, the state variables, the constants and the properties of defined attributes constituting the machine ACCESS CONTRACT are introduced. For instance, AccessType_AS, Agent_AS, Activity_AS are specified as sets containing predefined constants an AccessType (respectively an Agent, an Activity) variable could have. Table 5.2 illustrates a brief description of each variable specified within the ACCESS CONTRACT machine.

VARIABLE	Description
AccessType	the access type to be attributed after each access demand
Agent	the entity asking access authorization
Activity	the action to be performed once the access is authorized
Resource	the resource over which the access is requested
Role	the role demanding the access
Context	the situation that must be respected so that the access is granted
SecurityRule	the defined access control policy
SecurityRuleList	the list of security rules
RoleList	the list of roles
ContextList	the list of contexts

Table 5.2 – Security rules list example

Reminding here that a role structure is characterized by its description, the Organization by which it is issued (e.g. the subfactory SF1, SF2, etc.), the department to which it does belongs (e.g. logistics), and the trust threshold it has. On the other hand, the context structure is identified by its description (that could be for example shipping, renting, etc.), minInterval (which is the minimum allowable time interval between two successive access requests), nb_req_threshold (that corresponds to the maximum number of access requests in the minimum interval), working hours (corresponds to the working hours interval).

MACHINE *ACCESS_CONTRACT*
SETS
$AccessType\_AS = \{permission, prohibition\}$;
$Agent\_AS = \{agent1, agent2, agent3\}$;
$Activity\_AS = \{use, check, add, update\}$;
$Resource\_AS = \{truck1, truck2, truck3, truck4, record, trustfile\}$;
$Context\_AS = \{ctx1, ctx2, ctx3\}$;
$ContextDesc\_AS = \{ctxdesc1, ctxdesc2, ctxdesc3\}$;
$SecurityRule\_AS = \{SR1, SR2, SR3\}$;
$Role\_AS = \{role1, role2, role3\}$;
$RoleDesc\_AS = \{shippingworker, productionworker, supervisor\}$;
$Organisation\_AS = \{SF1, SF2, SF3\}$;
$Departement\_AS = \{logistics, manufacturing, administration\}$;
$State = \{free, occupied\}$

ABSTRACT_VARIABLES
$AccessType, Agent, Activity,$
$Resource, Context, SecurityRule,$
$SecurityRuleList, Role, RoleDesc,$
$Organisation, Departement, TrustThres,$
$RoleList, WorkingHours, ContextList,$
$ContextDesc, minInterval, nbRequests,$
$nbReqThreshold, resultAccessControl,$
$Requests, TimeLastAccess, AccessTime,$
$LastRequestId, currentTime, TrustValue, ResourceState$

INVARIANT
\\Typing invariants
$AccessType \subseteq AccessType\_AS \ \wedge$
$Agent \subseteq Agent\_AS \ \wedge$
$Activity \subseteq Activity\_AS \ \wedge$
$Resource \subseteq Resource\_AS \ \wedge$
$Context \subseteq Context\_AS \ \wedge$
$SecurityRule \subseteq SecurityRule\_AS \ \wedge$
$Role \subseteq Role\_AS \ \wedge$
$RoleDesc \subseteq RoleDesc\_AS \ \wedge$
$Organisation \subseteq Organisation\_AS \ \wedge$
$Departement \subseteq Departement\_AS \ \wedge$
$ContextDesc \subseteq ContextDesc\_AS \ \wedge$
$AccessTime \in \mathbb{N} \ \wedge$
$LastRequestId \in \mathbb{N} \ \wedge$
$currentTime \in \mathbb{N} \ \wedge$
$TrustValue \in 0..10$
$nbReqThreshold = 2$
END

Listing 1. The description of SETS, ABSTRACT VARIABLES and INVARIANT clauses.

The definition of sets and abstract variables are succeeded by the introduction of typing invariant properties in order to complete the model construction. Listing 1. includes the typing invariants of added variables For example, we define the variable Resource to store shared resources over which the access is demanded. Obviously *Resource* is a subset of *Resource_AS* (Inv4, Listing 1.).
Listing 2. defines typing invariants related to role, context, resource, request and security rules structures. As an example, the *ResourceState* invariant (invariant 8) is used to constrain changes in the state of a resource between free and occupied states.

MACHINE *ACCESS_CONTRACT*
INVARIANT
\\Invariant 1
$RoleList \in Role \leftrightarrow (RoleDesc * Organisation * Departement) \quad \wedge$
\\Invariant 2
$TrustThres \in Role \rightarrow 0..10 \quad \wedge$
\\Invariant 3
$minInterval \in Context \rightarrow (0..10) \quad \wedge$
\\Invariant 4
$nbRequests \in Context \rightarrow (0..2) \quad \wedge$
\\Invariant 5
$ContextList \in Context \leftrightarrow (ContextDesc * Resource) \quad \wedge$
\\Invariant 6
$WorkingHours \in Context \rightarrow \mathbb{N} * \mathbb{N} \quad \wedge$
\\Invariant 7
$TimeLastAccess \in ContextDesc \leftrightarrow \mathbb{N} \quad \wedge$
\\Invariant 8
$ResourceState \in Resource \rightarrow State \quad \wedge$
\\Invariant 9
$Requests \in \mathbb{N} \leftrightarrow (Context * \mathbb{N} * Role) \quad \wedge$
\\Invariant 10
$resultAccessControl \in Requests \leftrightarrow BOOL$
\\Invariant 11
$SecurityRuleList \in SecurityRule \leftrightarrow$
$(AccessType * Activity * Resource * Role * Context)$
END

Listing 2. Role, Context, Resource, Request and Security Rule typing invariants.

As a next step, this model is enriched by safety invariant properties plus the definition of events. The guard and action of these last must be specified in such a way that it establishes invariants preservation.

MACHINE *ACCESS_CONTRACT*
INVARIANT
\\Invariant 12
$card(SecurityRuleList) \leq card(SecurityRule\_AS) \quad \wedge$
\\Invariant 13
$\forall(sr1, sr2, accessT1, act1, rsr1, role1, context1, accessT2, act2, rsr2, role2, context2).$
$(accessT1 \in AccessType \quad \wedge \quad act1 \in Activity \quad \wedge \quad rsr1 \in Resource \quad \wedge$
$context1 \in Context \quad \wedge \quad role1 \in Role \quad \wedge \quad accessT2 \in AccessType \quad \wedge$
$act2 \in Activity \quad \wedge \quad rsr2 \in Resource \quad \wedge \quad context2 \in Context \quad \wedge \quad role2 \in Role \quad \wedge$
$sr1 \in SecurityRule \quad \wedge \quad sr2 \in SecurityRule \quad \wedge$
$(sr1 \mapsto (accessT1 \mapsto act1 \mapsto rsr1 \mapsto role1 \mapsto context1)) \neq$
$(sr2 \mapsto (accessT1 \mapsto act1 \mapsto rsr1 \mapsto role1 \mapsto context1)) \Rightarrow sr1 \neq sr2)$
\\Invariant 14
$\forall(rsr, req).$
$(rsr \in Resource \quad \wedge \quad req \in Requests \quad \wedge \quad ResourceState(rsr) = occupied$
$\Rightarrow resultAccessControl(req) = FALSE)$
END

Listing 3. Safety invariant properties.

Safety invariants:
In order to ensure consistent, correct and safe functioning process of the considered framework, we define a set of constraints as it is illustrated in Listing 3. These last, and that must be preserved by events specification, are formalized as follow:

1. Property 1: Once the smart contract is called upon a request is made, an access result shall always be returned.

2. Property 2: Ensure that there is only one instance for each access rule defined and shared within the Access smart contract.

3. Property 3: Once a shared resource is used by a certain entity, the access over it must be blocked and could never be attributed to another entity.

Events Specification

In the following Listings, we use events to describe the behavior of our *ACCESS_CONTRACT* Event-B machine.

MACHINE *ACCESS_CONTRACT*
EVENTS
$addPolicy = ANY\ \ role, context, rsr, act, accessT, sr$
WHERE $role \in Role\ \wedge$
$context \in Context\ \wedge$
$rsr \in Resource\ \wedge$
$act \in Activity\ \wedge$
$accessT \in AccessType\ \wedge$
$sr \in SecurityRule\ \wedge$
$sr \notin dom(SecurityRuleList)\ \wedge$
$(accessT \mapsto act \mapsto rsr \mapsto role \mapsto context) \notin ran(SecurityRuleList)$
THEN $SecurityRuleList := SecurityRuleList \cup \{(sr \mapsto (accessT \mapsto act \mapsto rsr \mapsto role \mapsto context))\}$
END;
END

Listing 4. *add Policy* event.

In Listing 4, we define the first event called addPolicy permitting to define and to add a new access control policy, based on the role demanding the access "*role*", the context defining the situation "*context*", the resource to be used "*rsr*", the action to be performed "*act*" and the access type to be attributed "*accessT*". This event requires the execution of addRole and addContext events before to be completed. As pre-condition, input parameters must be already defined and the security rule must not exist before within the "*SecurityRuleList*". The triggering of this event allows a new policy item to be added to the policies list whose size will be incremented.

MACHINE *ACCESS_CONTRACT*
EVENTS
$addRole = ANY\ \ rl, rldesc, org, dept, trust$
WHERE $rl \in Role\ \wedge$
$rldesc \in RoleDesc\ \wedge$
$org \in Organisation\ \wedge$
$dept \in Departement\ \wedge$
$trust \in TrustThres\ \wedge$
$rl \notin dom(RoleList)\ \wedge$
$(rldesc \mapsto org \mapsto dept) \notin ran(RoleList)\ \wedge$
$\forall(rr, oo, dd).(rr \in RoleDesc\ \wedge\ oo \in Organisation\ \wedge$
$dd \in Departement\ \wedge\ (rr \mapsto oo \mapsto dd) \in ran(RoleList)$
$\Rightarrow rr \neq rldesc\ \wedge\ (oo \neq org \vee dd \neq dept))$
THEN $RoleList := RoleList \cup \{(rl \mapsto (rldesc \mapsto org \mapsto dept))\}$
END;
END

Listing 5. *add Role* event.

MACHINE *ACCESS_CONTRACT*
EVENTS
$addContext = ANY\ \ ctx, ctxdesc, minIn, nbReq, rsr$
WHERE $ctx \in Context\ \wedge$
$ctxdesc \in ContextDesc\ \wedge$
$minIn \in 0..3\ \wedge$
$nbReq \in 0..3\ \wedge$
$rsr \in Resource\ \wedge$
$ctx \notin dom(ContextList)\ \wedge$
$(ctxdesc \mapsto rsr) \notin ran(ContextList)\ \wedge$
$\forall(cc, rr).(cc \in ContextDesc\ \wedge\ rr : Resource\ \wedge$
$(cc \mapsto rr) \in ran(ContextList) \Rightarrow cc \neq ctxdesc\ \wedge\ rr \neq rsr)$
THEN $ContextList := ContextList \cup \{(ctx \mapsto (ctxdesc \mapsto rsr))\}\ ||$
$minInterval := minInterval \Leftarrow \{ctx \mapsto minIn\}$
END;
END

Listing 6. *add Context* event.

Listing 5. and Listing 6. are responsible for defining and adding a new role, respectively a new context to their corresponding list, Rolelist and ContextList. For instance, addRole event uses in input the role name "*rl*", the corresponding description "*rldesc*", the organization to which the role belongs "*org*", the corresponding department "*dept*", and finally the trust threshold above which access demanders should have their associated trust scores "*trust*".

Listing 7. introduces the AccessControlAccept event. Through this last, an access request made by the role "*rl*", upon a certain resource "*rsr*" to perform the activity "*act*" within the specific context "*ctx*" is authorized. To do so, a set of guards are generated, these last represent the set of conditions that should be respected, they include the typing guards related to input parameters, the belonging guards over which the existence of role "*rl*" and context "*ctx*" within the Rolelist and respectively the Contextlist will be checked, finally the comparison guards defined to check the condition to be respected as well as the legitimate behavior of the access requesting agent. This last is evaluated through (i) the corresponding trust score "*TrustValue*" that should be above the trust threshold "*TrustThres*" of the associated role, (ii) the access requests number to detect a potential doubtful access demand, this last should not exceed the "*nbReqThreshold*", (iii) the last access request "*TimeLastAccess*" that should be launched within the minimum allowable time interval "*minInterval*", and (iv) finally the current access request that shall be launched during the working hours interval. As a result of this event, a list of variables will be modified through the event action clause specifically the "*Requests*" set, the "*LastRequestId*" and the "*TimeLastAccess*". Therefore the state of the resource will be changed to occupied and the access will be authorized.

MACHINE *ACCESS_CONTRACT*
EVENTS
$accessControlAccept = ANY \;\; rsr, rl, act, ctx$
WHERE **\\Typing guards**
$rsr \in Resource \;\wedge$
$rl \in Role \;\wedge$
$act \in Activity \;\wedge$
$ctx \in ContextDesc \;\wedge$
\\Belonging guards
$rl \in dom(RoleList) \;\wedge$
$(ctx \mapsto rsr) \in ran(ContextList) \;\wedge$
\\Comparison guards
$ResourceState(rsr) = free \;\wedge$
$TrustValue \geq TrustThres(rl) \;\wedge$
$nbRequests(ContextList \;\; (ctx \mapsto rsr)) < nbReqThreshold \;\wedge$
$currentTime - TimeLastAccess(ctx) >$
$minInterval(ContextList \;\; (ctx \mapsto rsr)) \;\wedge$
$(\forall(xx, yy).(xx \in \mathbb{N} \;\wedge\; yy \in \mathbb{N} \;\wedge$
$(xx \mapsto yy) = WorkingHours(ContextList^{-1}(ctx \mapsto rsr)) \Rightarrow$
$currentTime > xx \;\wedge\; currentTime < yy))$
THEN $Requests := Requests \cup \{LastRequestId + 1$
$\mapsto (ContextList^{-1}(ctx \mapsto rsr) \mapsto currentTime \mapsto rl)\} \;\;||$
$LastRequestId := LastRequestId + 1 \;\;||$
$TimeLastAccess(ctx) := currentTime \;\;||$
$resultAccessControl(LastRequestId + 1 \mapsto$
$(ContextList{\sim}(ctx \mapsto rsr) \mapsto currentTime \mapsto rl)) := TRUE \;\;||$
$ResourceState(rsr) := occupied$
END;
END

Listing 6. *Access Control Accept* event.

MACHINE *ACCESS_CONTRACT*
EVENTS
$accessControlRelease = ANY \quad rsr, rl, act, ctx$
WHERE **\\Typing guards**
$rsr \in Resource \quad \wedge$
$rl \in Role \quad \wedge$
$act \in Activity \quad \wedge$
$ctx \in ContextDesc \quad \wedge$
\\Belonging guards
$rl \in dom(RoleList) \quad \wedge$
$(ctx \mapsto rsr) \in ran(ContextList) \quad \wedge$
\\Comparison guards
$ResourceState(rsr) = occupied \quad \wedge$
$(currentTime - TimeLastAccess(ctx) \leq$
$minInterval(ContextList \quad (ctx \mapsto rsr)) \quad \vee$
$nbRequests(ContextList \quad (ctx \mapsto rsr)) \geq nbReqThreshold \quad \vee$
$TrustValue < TrustThres(rl) \quad \vee$
$\neg((\forall(xx, yy).(xx \in \mathbb{N} \quad \wedge \quad yy \in \mathbb{N} \quad \wedge$
$(xx \mapsto yy) = WorkingHours(ContextList^{-1}(ctx \mapsto rsr)) \Rightarrow$
$currentTime > xx \quad \wedge \quad currentTime < yy)))$
THEN $Requests := Requests \cup \{LastRequestId + 1$
$\mapsto (ContextList^{-1}(ctx \mapsto rsr) \mapsto currentTime \mapsto rl)\} \quad ||$
$LastRequestId := LastRequestId + 1 \quad ||$
$TimeLastAccess(ctx) := currentTime \quad ||$
$resultAccessControl(LastRequestId + 1 \mapsto$
$(ContextList{\sim}(ctx \mapsto rsr) \mapsto currentTime \mapsto rl)) := FALSE$
END
END

Listing 8. *Access Control Reject* event.

5.4 Verification of the smart contract behavior

In this section, our main objective is to demonstrate our approach by (i) validating the formal model introduced in Section. 5.3.2 and then (ii) checking that both the presented implementation and design of the smart contract verifies well some required typing and safety properties. More details about the proposed framework are provided in Chapter 3. As a first attempt to validate the Event-B models, we have applied these last to a simplified real example derived from the use case study illustrated as well in Chapter 3.

Indeed in this scenario, we added two new roles called 'shipping worker' and 'production worker' belonging respectively to the 'logistics' and 'manufacturing' departments within the subFactory SF1 and having as trust threshold the value of '0.5', correspondingly a new context is added to the contexts list, named 'shipping', it concerns the resource 'truck3' during the working interval '8to18' and where 2 access requests are permitted during the minimum time interval set up to '5 minutes'.

Afterwards, a corresponding security rule is defined and added to the security rules list after being specified as follow: SR1(access type = "permission", activity= "use", resource= "truck3", role= "shipping worker", context= "shipping"), in other words, this security rule authorizes the shipping worker to use the truck3 resource according to the context shipping, that is, his trust score is above the trust threshold, the current time is within the working interval and at most 2 access requests are launched during the last 5 minutes.

Subsequently, we defined 3 access demands corresponding respectively to 2 legitimate

behaviors and a malicious one. For the legitimate behaviors, two shipping workers from the logistics department with a trust score equal respectively to 0.7 and 0.65 demand for the first time to use the truck3 resource during their working hours interval. For the malicious behavior, we assume that a shipping agent launches an on-off attack resulting on a low trust score while demanding access over the truck3 resource.
In order to demonstrate that the formal specification of access contract models is correct, we aim to validate Event-B models using ProB model-checker and animator. This validation allows to verify that the invariants are preserved by all events. Since these models are deterministic and have finite state spaces, the model-checking and animation are sufficient to validate our model for a given initial state than theorem proving that requires considerable efforts.

Verification using model-checking :
Model-checking [45] is an automated approach for verifying that a system model conforms to its specifications. The system behavior is formally modeled and the specifications, expressing the expected properties of the system, are also formally expressed, in our case via formulas of the first-order logic. All experiments were conducted on a 64-bit PC, Ubuntu 16.04 operating system, an Intel Core i5, 2.3 GHz Processor with 4 cores and 8 GB RAM. Using the ProB model-checker and based on mixed breadth and depth search strategy, we have explored all states: 100% of checked states with 641 distinct states and 1613 transitions during 1987 milliseconds. No invariant violation was found, and all the operations were covered. This verification ensures that invariants are preserved by each event. Otherwise, a counter-example would be generated.

Verification using animation :
ProB can be used as a complementary of model-checker as an animator. Verification using animation is very important and can detect a series of problems, such as unexpected behavior of a model. ProB animator allows to visualize the dynamic behavior of a Event-B machine and we can systematically explore all the accessible states of a Event-B machine to check the studied properties. We have successfully applied the animation of ProB on the operational scenario that described at the beginning of this section. The animation of this scenario demonstrates the behavior of the our specification which implies that we have a verified and validated specification. However, if this is not the case, then we have to go back to the initial specification to correct the unacceptable behavior and re-apply the animation until the specification conforms to requirements.

Conclusion

In this chapter, we have applied the formal verification concept to smart contracts developed and deployed within our DRMF framework. The aim is mainly to verify the compliance of these last with their specification while validating their safety and functional accuracy under specific properties and for given behaviors. To do so, we adopted the Event-B formal method for translating solidity written smart contracts. Model checking technique has been exercised therefore on the resulted formal model to judge and validate some properties of interest. The presented method was applied to a simplified real example derived from a use case study describing shared resources management among collaborating organizations under Industry 4.0 environments.
As a future work, we will focus on enriching our model with other properties when considering malicious parties introducing specific scenarios of attacks.

Chapter 6

Blockchain Based Trust Management Mechanism for IoT

Contents

INTRODUCTION

We have seen alongside the previous chapters how integrating the blockchain technology could ensure the traceability and the confidentiality in addition to the immutability of shared information that once stored within the distributed ledger cannot be altered nor modified without being traced or revealed by validator peers. In this chapter, we argue that despite the blockchain is an effective technology for managing shared data traceability and notarization, it alone cannot support the reliability and the authenticity of such data regarding the trustworthiness of network participating entities. In such case, false data generated by malicious entities becomes immutable once recorded on the distributed blockchain. One approach to improve the trust and reliability of the data is to use a trust management mechanism to assess the trustworthiness of entities it is produced and collected by. Such entities within IoT environments are susceptible to faults and could be easily targeted by malicious threats and attacks, thus they cannot be blindly trusted.

In this direction, our objective in this chapter is to propose a secure and distributed trust management system based on the blockchain technology so that we can take advantages of security features it provides regarding reliability, traceability and information integrity. Blockchain based trust management can provide tamper proof data, enable a more reliable trust information integrity verification, and help to enhance its privacy and availability during sharing and storage. For this purpose, we design and implement a blockchain based trust architecture to collect trust evidences, to define a trust score for each device and to securely store and share them with other entities within the network by embedding them into blockchain transactions.

To do so, we propose first a generic trust mechanism for assessing the trustworthiness degree of each entity within the network and deriving a trust score that will be shared between involved entites and considered within the decision making process. In order to evaluate and experiment the proposed trust mechanism, we have chosen to use our solution within a specific use case which is the RPL routing protocol to ensure secure routing by protecting the network against misbehaving, selfish and malicious nodes regarding the routing procedure. We opted for such protocol mainly because RPL is considered as one of the emerging routing standards for ensuring multi hop communications in IoT networks.

In brief, the focus of this chapter is first how to share trust related information within the IoT network via blockchain and second how to assess trust within a multi-hop IoT network regarding the routing procedure.

For effective clarity and better understanding, this chapter will be divided into two parts where each one deals with one of the above mentioned issues. The rest of this chapter is organized as follows. Section 2 presents our blockchain based framework for trust management in IoT. In this section, and after discussing trust management systems related issues, an overview of the proposed scheme as well as its detailed design is given

followed by a set of experimental results validating its performance. Therefore and in order to validate our trust model regarding the RPL protocol, we present in Section 3 our trust framework for IoT regarding the routing procedure. In this part, we recall the basic concepts of routing protocols for IoT and presents related proposals regarding their security enhancements and trust aspects. Thereafter, a detailed design of the proposed trust model is given and a set of experimental results are shown. Finally in Section 4, the chapter ends up with some conclusions and an outlook of our future work in this area.

6.1 Toward a blockchain based framework for trust management in IoT

6.1.1 Related works and problem statement

Security presents a significant challenge for the implementation and the realization of IoT scenarios due to the different standards and communication stacks involved, the heterogeneous nature of IoT communications and respectively the imbalance in involved devices' resources capabilities. The presence of such issues makes it challenging to satisfy the security and privacy requirements of IoT [159].
Such requirements include data confidentiality and authentication, access control within the IoT network as well as privacy and trust among things and services.
When a number of things with dynamic behaviors communicate in an uncertain IoT environment, trust plays an important role in establishing secure communication, reliable data exchange and enhanced decision making.
Trust is a very general topic that may be applied to virtually any context. For this reason, it has been extensively studied in many areas such as sociology, economics, computer science, and so on [85], that's why we find its definition always depends on the environment, the objectives and the context in which it is applied. Its management has not been comprehensively investigated by the current research in the context of IoT [183].
We have reviewed in Chapter. 2 existing proposals carried out in the area of trust management for IoT with different design purposes. According to the carried review, these last generally consider a common small set of parameters to evaluate trustworthiness which are either related to QoS, social or reputation aspects without focusing on the security aspect that consists of verifying the confidentiality and integrity of trust evidences during their collection, propagation and communication between participating devices. Instead, they assume that collecting information from a large number of entities and executing aggregation operations on the exchanged trust related information will result in a relatively accurate assessment. Another issue that has not been extensively studied within proposed trust models is the sharing of trust information. In fact existing trust models do not explain how trust scores are represented and how they are

interpreted by involved and participating entities during the evaluation. In this context, some works [110, 34] made use of ontologies to represent trust and to annotate related trust information generated by either devices or resources within the network in question.

Another issue that trust management systems face is the ability to link an identity to a single entity and to prevent that a specific entity obtain more than one identity. Whereas identity management can play an important role in measuring the credibility of exchanged trust related information and resisting against Sybil attacks where a malicious entity can forge different identities to trick the system with multiple fake entities. As a partial solution to this issue, authors in [33] have introduced a framework to detect possible sybil attacks against trust management schemes within peer-to-peer (P2P) networks. This has been shown to almost entirely prevent a Sybil attack, although the cost to the network in terms of the resources required to verify each peer is high which makes the solution unsuitable for IoT networks.

Finally, a last but not least important issue is the storage of trust information, In fact and regarding IoT networks with tight resource constraints, some of the existing trust systems store trust information for devices with the highest trust values [39], others for those that have been recently encountered and interacted with which is not that good deal especially as trust computation depends on the past evaluations of all behaviors and interactions, while others keep trust related information stored within a central single entity [142] raising as a consequence thereof problems related to scalability.

Currently in the scientific research, numerous efforts have been emerged leading to significant advances in the fields of attacks resiliency, cryptography, identity management and decentralized computer networks resulting in the emergence of the blockchain technology, which has the potential to fundamentally overcome raised challenges and to solve almost of the above mentioned issues. This technology has been springing up, covering various fields such as data sharing, data storage, reputation systems and many other security services. In the context of data sharing, authors in [180] have proposed a blockchain-based data sharing framework that sufficiently addresses the access control challenges associated with sensitive data stored in the cloud. Similarly, in [94], authors have proposed ProvChain, a system that provides a data provenance system based on blockchain technology where occurring changes in the cloud storage system are tracked, recorded and events corresponding to the actions of the users are generated.

In the context of blockchain based reputation systems, authors in [55] have proposed a new reputation system that is practically applicable to multiple networks and with the objective is to store single dimension reputation value from the completed transactions. Another work was presented in [117] where a new security model based on the blockchain technlology was proposed 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.

Discussion As seen the blockchain technology has been used for several purposes and within various fields and application domains. However just few studies [55, 117] have focused on integrating this last within trust and reputation systems. On the other side, a secure and distributed based trust system is essential to ensure the network security and to guarantee trust information confidentiality, integrity and privacy during sharing and storage. Besides, trust information may contain sensitive information about trust providers and targets. Hence, there is a need to secure not only the network but also to ensure integrity and trustworthiness of exchanged information.
That's why providing an adequate solution to distributively and securely store and share trust information within the IoT network while guaranteeing their integrity, availability, authenticity, and authorization become crucial and no more important than ever. In this direction and given the noted features of blockchain technology, this last applied to trust management systems in IoT provide promising possibilities and solutions to issues they encounter as it was discussed previously.
In this part, we consider consortium blockchain to establish a secure blockchain based trust system. The use of consortium blockchain refers to the fact that this last uses permissions to control over who can join the network, and who can participate in the consensus process that doesn't have to spend an enormous amount of energy, computing power, time and money to be reached which makes it more suited to IoT environments compared with Bitcoin and public blockchain solutions where a puzzle need to be solved by miners to identify the value of nonce. In other words, we consider blockchain as a generic decentralized secured data storage structure were blocks are divided in two parts wherein: (i) a block header containing metadata where reference to the previous block hash is incorporated and (ii) data payloads representing the stored trust records.

6.1.2 Proposed approach

The main objective of this work is to propose a novel trust management system based on the blockchain technology. Our framework aims to define and evaluate a trust score for each device and to securely store and share these scores with the different devices within the IoT network guaranteeing their transparency, integrity, authenticity, and authorization. A description of the overall architecture of the proposed system is presented in Fig. 6.1.
In the following, we will describe the overall architecture of our proposed scheme. To efficiently compute trust values and to securely store and share them within the blockchain network, we first need to clearly understand the detailed composition of our system. Furthermore, a brief description of the required interactions to be established will be provided.

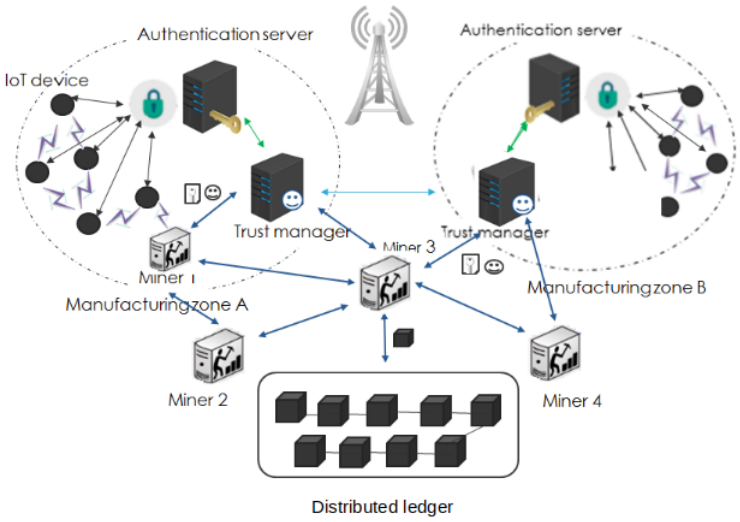

Figure 6.1 – Overall Architecture of the proposed model

System model

As seen in Fig.6.1, our system is composed of a number of manufacturing zones where each zone is made up of a set of physical resources (e.g. machines, ordinary sensors, IoT devices, etc.) along with an authentication manager, which is responsible for making authorization decisions, verifying devices identities and generating authorization tokens. Furthermore, each device is connected to a trust manager, which is in charge of assessing the trustworthiness degree as well as aggregating and computing a final trust score for each participating device within the manufacturing zone. In the case of IoT devices with tight resource constraints, this entity is assumed to be deployed in a more powerful network component that will be connected directly to each device, otherwise it is deployed within the device itself. The detailed design as well as the validation of the trust model will be presented in Section 6.2. Thereafter, a set of specific entities are deployed to receive trust data storage transactions, to create blocks out of it and to broadcast it into the blockchain network containing an ordered list of records linked together where each block references the previous one, known as the parent block, and storing trust data allowing as a consequence thereof the verification in a decentralized manner of trust data exchange among participating devices.
A detailed architecture of the proposed system is presented in Fig. 6.2. This last illustrates the different modules that make up each entity within the proposed system as well as interactions that could be established among them. More details are presented in the next subsection.
We remind here that in this work we consider just only one zone for our design that means that relationships and interactions established between devices belonging to different zones are not considered.

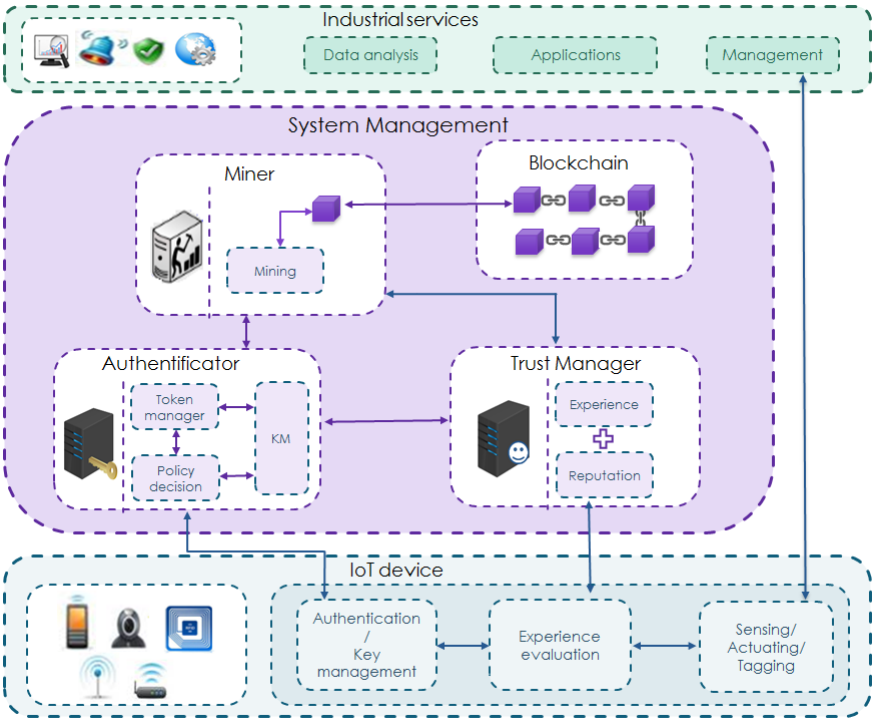

Figure 6.2 – Detailed architecture of the proposed model

System detailed design

As seen in Fig. 6.2, three main conceptual layers can be identified in our proposed model.

Device layer :
This layer consists of IoT devices that aim to collect and process information. These devices include sensors, actuators, RFID tags, machines, robots, etc. Their main tasks are to perform different functionalities such as querying and gathering data (e.g. location, temperature, humidity, etc.), executing tasks and other primary functions. Exceptionally and related to our proposed scheme, these devices will execute additional tasks related to trust management. By this way they will evaluate, send, access, and retrieve trust related information to and from the management system. In addition, they interact over the Internet with the industrial services querying and supervising their state as well as any information associated with them, taking into account their quantified trust scores analyzed, processed and stored within the System management. These scores can be used by devices to make decisions of sharing data with other devices according to their trust scores. Besides industrial applications and services can obtain data in a reliable way only from those that satisfy certain trust threshold.
To this end, each device after communicating with other devices, will assess their behavior during the interaction on the basis of certain metrics such as the packet delivery ratio, the community of interest and its honesty in attributing recommendations. These

metrics will be explained in the next Section. The assessment result will be periodically sent after authentication to the trust manager entity in order to compute the corresponding trust score and to forward it to the blockchain network.

System management layer :
This layer is composed of individual connected powerful entities responsible for the secure establishment, the efficient running and the reliability of the proposed scheme. More specifically this layer performs computation, verification and analysis on trust related information. It additionally reports actions on trust and reputation scores which are securely stored in a decentralized structure and indexed appropriately triggering later an action on it when needed or required. Results of every action completed are broadcast into an immutable network of consensus entities to guarantee verification, auditing and validation of the action in question. Finally, the layer has a responsibility of authenticating every device and managing their actions keys pertaining to trust data access from the entire system.
The different entities in the system management layer are:

- **Trust Manager:** This entity enables to establish a trusted and reliable environment where devices can interact with each other as well as with industrial IoT services without worrying about risks related neither to devices changing behaviors nor to transmitted information confidentiality and integrity. Thus they could make decisions and obtain data in a reliable way according to the assessed trust scores.
  We remind here that the trust framework is assumed to be deployed in the same target entity when it comes to non-constrained devices whereas it could be deployed outside in case of constrained ones.
  For the computation of trust, this last is computed on the basis of the direct observations and interactions referred as the direct trust and the recommendations exchanged between neighbors termed as the indirect trust.
  For trustworthiness assessment, each entity maintains multiple trust properties regarding neighbors it interacts with namely cooperativeness, competence and community of interest where (i) cooperativeness property reflects the cooperativeness level that entity e_j has as evaluated by entity e_i based on its behavior monitoring during the time interval $[0..t]$. It is calculated using the Packet Delivery Ratio (PDR) representing the number of packets successfully delivered to the number of those that have been sent out by the sender, (ii) the competence property provides the degree of the entity's ability to perform its intended tasks, assessed based on the energy and the computation capability it has in order to verify whether it is enough competent to perform its tasks or not, and last (iii) the community of interest property provides the degree of the common interest or similar tasks of e_j as evaluated by e_i computed as the ratio of the number of common community interests over the total number of their community interests.

The combination of these properties as well as with past trust evaluations during a specific time interval Δt will produce an overall trust value that can be efficiently and effectively used to ensure security improvements. The reason for which we consider past trust evaluations is mainly due to the fact that an entity behavior is not always constant but often changes in time.
A detailed description of the process to be executed for the trust management as well as the metrics considered for its validation will be given in the next part of this chapter.

- **Authenticator:** This entity is mainly responsible for verifying the validity of devices' identities as well as the legitimacy of demands and requests sent to both trust data storage and management systems. IoT devices and smart objects are authenticated based on the provided credentials. In our framework, we rely on the openID Connect [145] (OIDC) which is an identity layer on top of the OAuth 2.0 protocol [74]. It enables clients to verify the identity of the device based on the authentication performed by an OpenID Provider, and to obtain basic profile information about the device in an interoperable and REST-like manner [145]. We have chosen OIDC since it has different characteristics related to IoT environments. OIDC is free, open and decentralized (no central authority approves or registers relying parties or service providers). Its integration does not require complicate update in the deployed application. Indeed, it follows a restful approach which make it easy to use and to interoperate. Finally, it gives the possibility to use a JSON structure token that carries information about the device.
This entity could be composed of two subcomponents:

  (i) The Policy Decision Component (PDC): this module is in charge of making authorization decisions based on the policies defined to assign permissions that a device has related to trust storage and management systems. In case of a successful authentication process, this module generates an access token which is delivered in order to avoid subsequent authentication procedures. The PDP will be in charge of checking and validating this token. These access tokens are comprised of different fields, among them, we find: the requester authentication keys, the subject, the target resource, and the access rights (action, target resource and conditions). This approach allows subject devices to access target resources as many times as required without running all the authentication process till its beginning.

  (ii) The Key Management Component (KMC): this module generates authentication and transactions related keys that are used to authenticate devices' validity within the system and to digitally sign actions to be executed on the stored trust and reputation scores. These keys help to guarantee a level of security of the scheme. They include:

- Transaction private key: This key is generated for a each device willing to execute actions on trust scores, it is used to digitally sign requests for trust data access.
- Transaction public key: This key is generated for each device and sent to trust data storing entities' authenticators in order to be used as verification of requesters identity for data access. In addition, this key is also used to encrypt a package to be sent to each requester device. Consequently, for a request whose signature cannot match to the appropriate public key, such a request is considered as invalid.

- **Validator:** This entity is tasked to process and verify the authenticity, the validity and the integrity of trust records as well as details pertaining to each transaction. This last once received, will be broadcast into the network of validators where they will check its validity and then package it into block in order to be chained within the ledger. Obviously, all validator entities have a complete copy of the ledger kept up to date.
  In our framework, we consider Multichain [70] blockchain technology which is a consortium blockchain protocol that manages the access to the blocks using a list of registered participants. Only participants who have been previously registered have access to read and write blocks within the ledger. The consensus method that Multichain implements to approve transactions is the Round Robin (RR) algorithm. The reason for which we have chosen Multichain mainly refers to the fact that this last respects most the criteria of our requirements. Multichain is a permissioned blockchain solution based on the use of streams that act as an independent append-only collection of items which enforces more the confidentialy of shared data. Second it is characterized by its flexibility allowing permissions changes and delegations. Third it is based on the RR consensus mechanism where no complex computation resources regarding processing capabilities, computing power and time are required for the validation of blocks without forgetting currencies as it is the case of bitcoin based solutions where miners carry out computationally expensive calculations using their compute power in order to validate new blocks and add them to the blockchain.

Required interactions

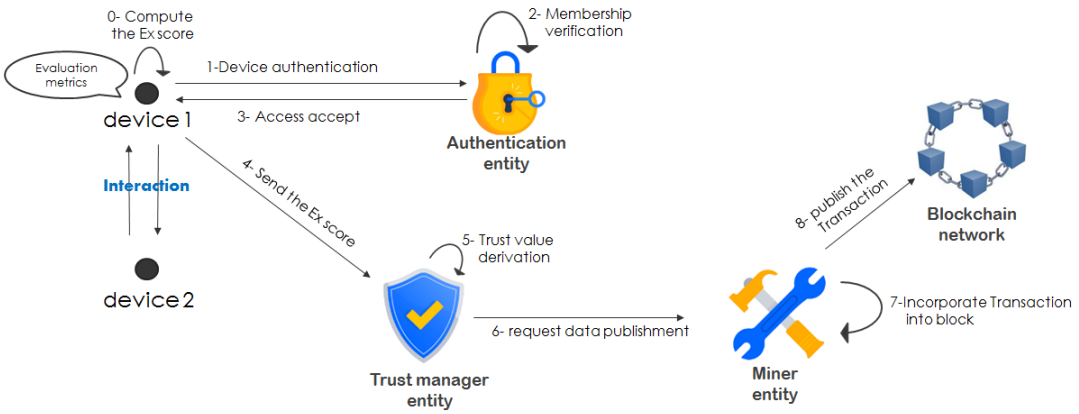

Figure 6.3 – Required interactions

The required message exchange and interactions among the different entities within our system have been split into three main stages. Fig. 6.3 shows the required interactions to be established for the realization of our system's proposed objectives.
Our scenario assumes an IoT device tries to store trust scores within the Blockchain network. During the first stage, the device needs a proper authorization credential to access the system. In case it does not have a valid capability token (e.g., it was expired, revoked or it's its first request), it contacts the authenticator entity using its membership key, indicating the specific resource and action to be performed. This last will make authorization decisions based on the policies defined within the PDC in order to assign permissions. In case of a successful authentication process, this component will deliver an authentication assertion and send it to the Token manager module in order to obtain authorization token. Obviously, before demanding access and while interacting and communicating with its direct neighbors, each device will monitor their behavior and assess their trustworthiness level based on the evaluation metrics, as a result, an experience score will be computed, signed with the Transaction private key and sent to the trust manager entity using the obtained authorization token. Reminding that in case of resource constrained IoT devices this entity is assumed to be employed in a separate and more powerful network entity, while in the case of a more powerful device, it is supposed to be part of it. The trust manager entity will take in charge the computation of the Reputation score and the derivation of an overall trust score based on the equations presented as well in Section 6.2. Thereafter, It sends the transaction signed with the device's private key to the validator, this last consists of the trust score, the Tr type (storage, access, update), the time stamp, the target ID and the established interaction. The validator entity and once receiving the transaction, forward it to the whole network where each entity will check the validity of the transaction, then assemble verified transactions into a block to be added to the existing ledger.

6.1.3 Implementation and Performance analysis

The implementation of our proposal is conducted using NS3 simulation tool for the simulation of the IoT environment. Our choice mainly depends on the fact that NS3 has emerged as the open simulation platform for networking research, it is a free and an open source discrete-event network simulator, targeted primarily for research and educational use.
In the other hand and for the implementation of the blockchain network, we have used Multichain as presented earlier. Table 6.1 lists parameter values used for this implementation. We consider an IoT environment with a number of IoT devices varying from 50 to 100 devices. These IoT devices are randomly scattered in the network. Devices belonging to the same community could share common interests or execute the same function and task. Each device is attributed a random number between 1 and 10 that reflects its belonging to one of the ten existing communities of interest within the IoT network. Besides, it could belong either to one community, two or three at the same time. For the compromised model, this last involves a number of malicious devices representing 20% of the total number of network devices. The behavior of such devices starts with the beginning of the network lifetime. In this model, three kinds of attacks were considered such as:

- Bad Mouthing attack: where malicious nodes aim to ruin the reputation of other well-behaved nodes by providing false recommendations against them what would decrease their trust score.
- Ballot Stuffing attack: contrarily to the previous one, malicious nodes here aim to promote other malicious nodes by providing good opinions about them what would increase their opportunity to be trusted.
- On off attack: As its name indicates, in this scenario, the malicious node behaves well and badly alternatively. By this way it could perform easily an attack before the trust system will be aware of.

For the Multichain network, in a Dell Precision M6800 machine with 4th Generation Intel Core i7 processor and 8Gb of RAM, we have used three virtual Linux machines (Ubuntu 16.04 version) in which we have deployed Multichain and set up the functionalities of validators within the blockchain network. Moreover and in order to more securely the storage and sharing processes, we have created three streams within the Multichain blockchain. These streams include:

1. A **Keys** stream: which used for the distribution of public keys.
2. A **Data** stream: which is used to store trust scores encrypted using the symmetric AES cryptography scheme.
3. An **Access** stream: which provides access for encrypted trust files.

The focus in this part is to evaluate our proposal's performance, reliability and resiliency against malicious attacks. Our simulation results have three parts. First, we evaluate the resiliency of our proposal against malicious behaviors. We then demonstrate the effectiveness of our storage and sharing solution based on the blockchain technology through the evaluation of the average response time as well as the number of established transactions and the computation power taken by each entity involved in the mining process.

Resiliency against attacks

In the following, we examine the resiliency of our proposal against malicious attacks launched by a set of devices within the IoT network.
First we evaluate the evolution of the average trust value of a well behaved node while varying the number of total bad nodes launching bad mouthing attacks.
The total number of network nodes is fixed to 50. In this part, we test the resiliency of our proposal against malicious behavior, to do so we fix the number of dishonest nodes, this last represents 20% of the total nodes number in the first case and 40% in the second one.

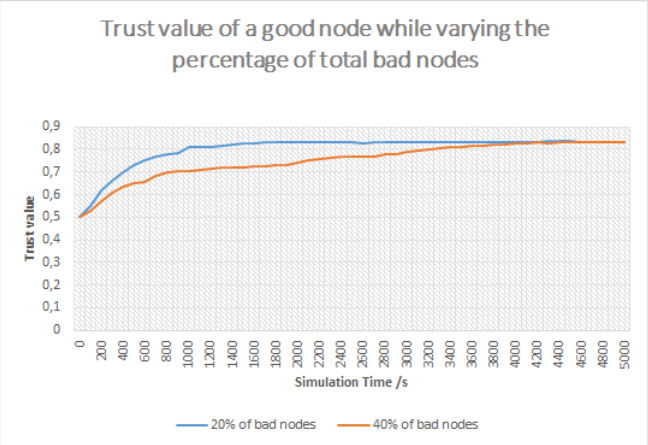

Figure 6.4 – Well behaved node trust evolution

Fig.6.4 shows average trust evaluation results towards a good node while varying the percentage of total bad nodes. We can see here that as the population of malicious nodes increases, both the convergence time and trust bias increase. However, the system is found to be resilient to malicious attacks for a percentage of bad nodes going till 40% of the total number of network nodes.

Simulation parameter	Value
Simulation tool	Ns3-3.13
Simulation run time	2 hours
Simulation coverage area	50m x 50m
Nodes distribution	Random
Total number of nodes	50..100
Number of malicious nodes	20%, 40%
trust update period	300s
Initial trust value	0.5
trust interval	0..1
Multichain mining diversity	0.3
Multichain Block size	8Mb
Multichain Transaction size	4Mb

Table 6.1 – Network related parameters used in simulation analysis

In Fig.6.5 we evaluate the evolution of the trust value of a malicious node while varying the percentage of total bad nodes launching ballot stuffing and on-off attacks. For on-off attack, the malicious node behaves well and badly alternatively regarding its packet delivery behavior. Specifically, the bad node deliver packets in the defined interval to gain high trust scores, once its score rises to 0,75, it starts to behave worse and when it senses its score drops below 0.4, it starts to act well again to rise its trust score.

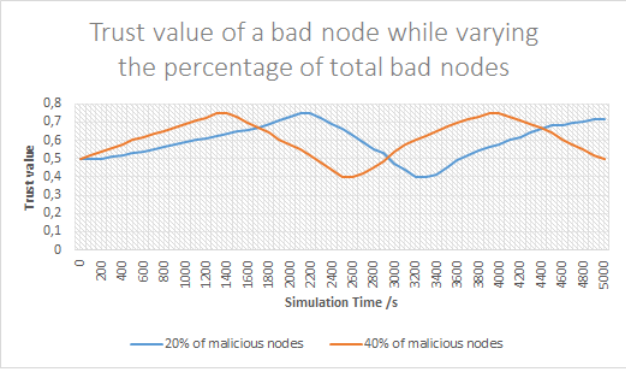

Figure 6.5 – Malicious node trust evolution

When it comes to the effect of the percentage of malicious nodes, we observe that the trust values fluctuation is considerably higher when this percentage is higher because more malicious nodes can collude together to promote the bad node and to quickly bring its trust level. The results presented in the previous two paragraphs are based

on the idea that each node uses the past and historical trust evaluations during the time interval $\Delta t = 1800s$ for the computation of a novel score. In this paragraph, we want to show how using the blockchain technology would enhance the accuracy and the resiliency of the network in question against on-off attacks. In fact and thanks to the traceability feature of the blockchain, trust information are time-stamped and permanently kept within the ledger for later use. For instance by tracing and reviewing historical scores, we can identify such alternative malicious behavior and as a consequence thereof we can penalize the malicious node and prevent it from gaining high trust scores again. As illustrated in Fig. 6.6, we can see clearly that once the bad node drops its trust score below 0,4 and starts to behave well again, this last, and thanks to the feature provided by the blockchain technology, is penalized and its score remains closed to 0,4.

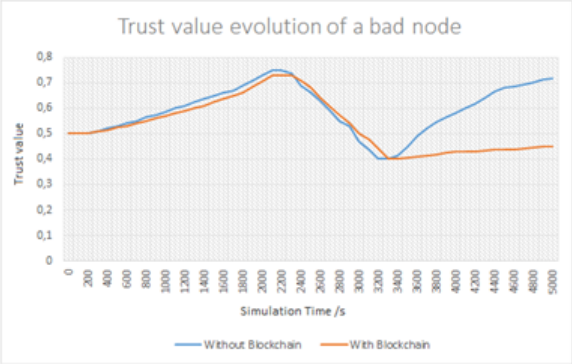

Figure 6.6 – Malicious node trust evolution considering blockchain

Performance evaluation

We evaluate here the response time while varying the size of the file holding trust informations and trust scores of each evaluated entity within the network, The file size ranges from 1KB to 2MB. As it is shown in Fig.6.7, the average response time of the blockchain network grows in parallel with the increasing size of the trust information file. This last reaches 1937ms for a 2Mb trust file size which is acceptable considering the security features provided by the blockchain technology. Moreover and in order

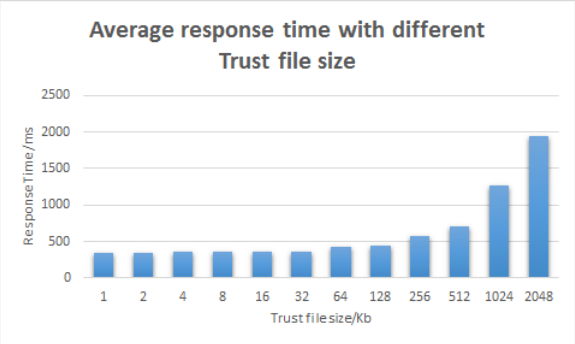

Figure 6.7 – Average Response Time

to evaluate more the performance of our proposal as well as its relevance and adequacy to IoT environments, we have evaluated in this step the computational resources needed to process incoming requests and transactions by validator entities. To do so, we have recorded the time at which the trust information file is sent for storage within the blockchain network as well as the time at which a confirmation for successful transaction is received. Results we got show that this process requires 750ms to be completed for a 512Kb trust file size, 35% of CPU and 0.016Gb of RAM consumption needed for the validation of transactions, the creation of blocks and the integration within the blockchain network which demonstrates and proves its deployability and feasibility to IoT environments. Fig. 6.8 shows the percentage of successful storage transactions processed in the Multichain network according to the size of the trust information file. To do so we have recorded the number of successful processed transactions over the total number of established ones and we have calculated the related percentage. As shown in Fig. 6.8, our proposal performs well in terms of successful storage transactions, the average throughput of the Multichain blockchain is 97.55%.

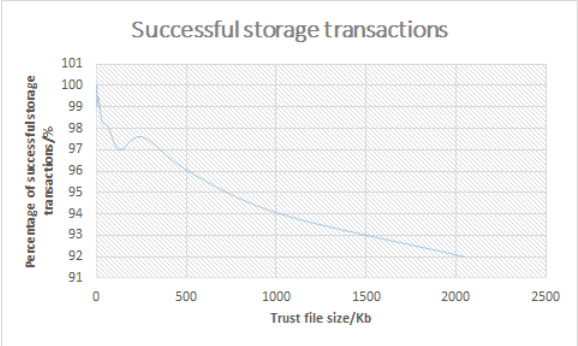

Figure 6.8 – Successful transactions

Discussion

In this section we have evaluated the performance of our proposal regarding its resiliency to launched attacks and its performance in terms of response time, computational resources and processed transactions. Our analysis of these metrics has shown that our proposal gives higher resiliency to attacks in comparison with the basic one without blockchain. This is mainly due to the traceability feature of the blockchain technology where not only established transactions but also the trust information file are kept permanently stored within the ledger which gives us the possibility to have an overall vision of entities' past behavior what could be both practical and useful for predicting the future behavior of malicious entities launching especially on-off attacks. Furthermore, and in order to prove the applicability of the blockchain within such a model while fully respecting the design goals we set at the beginning, we have evaluated the response time taken for the processing of storage coming transactions, we have shown that this last maintains low even with greater file sizes which supports the real time assessment objective.

6.2 Our trust framework for IoT regarding the routing procedure

We have presented in the previous section the design and the implementation of our trust management system based on the blockchain technology to collect trust evidences and to securely store and share them within and through the blockchain network. Using such technology, we enable a more reliable trust information confidentiality and integrity verification during storage and sharing and we make a time stamped log of both entities' transactions and behavior. In this part, we proposed a generic trust mechanism for assessing the trustworthiness degree of each entity within the IoT network. However the presented architecture does not address the problem of trust management and that's why we propose in the next section to evaluate and experiment the proposed trust mechanism. To do so we need a specific use case scenario in which the effectiveness and the resiliency of our mechanism are demonstrated. Our focus therefore was on ensuring multi hop communications in IoT networks. To do so we have chosen to use our solution within the RPL routing protocol to ensure secure routing by protecting the network against misbehaving, selfish and malicious nodes regarding the routing procedure. We opted for such protocol mainly because RPL is considered as one of the emerging routing standards for ensuring multi hop communications in IoT.

6.2.1 Background and Literature review

Routing Protocol for Low power and lossy networks

RPL, developed by the IETF working group, is an IPv6 routing protocol specifically designed for LLNs with very limited resources in terms of energy, computation and bandwidth. This protocol mainly targets collection based networks made up of nodes interconnected according to a specific topology called Destination Oriented Directed Acyclic Graphs (DODAG) as illustrated in Fig. 6.9, where sink nodes and gateways act as the roots of the Directed Acyclic Graphs (DAGs). Within each DODAG, each node is assigned a rank representing its position in the graph. Its computation depends on a set of specific routing metrics (e.g. delay, link quality, throughput, etc.). The translation of these metrics into ranks is based on an Objective Function (OF) responsible for rank computation and parent selection. The DODAG construction and maintenance phases are based on a set of control messages namely DODAG Information Object (DIO) delivered by the DODAG root to build routes, DODAG Information Solicitation (DIS) broadcast by nodes willing to join the network, Destination Advertisement Object (DAO) used to propagate reverse route information and Destination Advertisement Object Acknowledgement (DAO-ACK) messages sent as an acknowledgement of DAO messages.

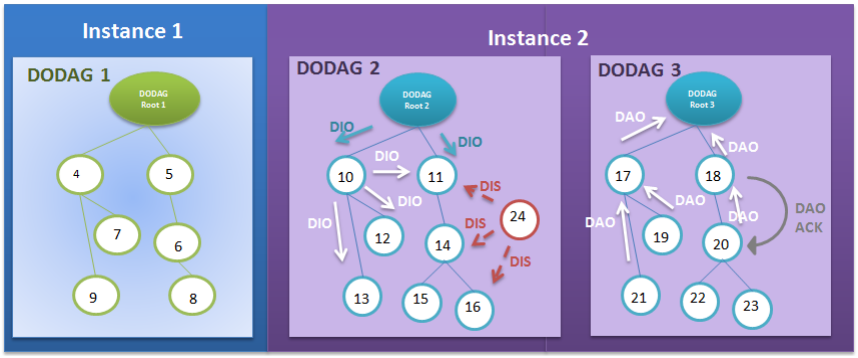

Figure 6.9 – RPL network topology and exchanged messages

Trust for the secure routing

Several trust management schemes have been developed to ensure secure routing by protecting the network against misbehaving, selfish and malicious nodes regarding the routing procedure. These nodes generally aim to attack the routing protocol by dropping, modifying and altering the transmitted routing related packets, as well as disrupting the routing processes. The integration of trust within the routing protocol could solve efficiently and effectively the problems to be faced when securing the routing scheme in IoT environments.

However the proposed schemes are particularly dependent on the environment they targeted and the routing protocol they are designed to be integrated within. Additionally, some of them have been tailored to wireless sensor networks, without considering the inherent requirements and features of IoT scenarios and more specifically those related to smart factories environments. In fact they do not keep in view wireless interference, QoS and energy constraints which make them little suitable to IoT devices used within smart factories.

When it comes to these networks, we may found several enhancements of the RPL protocol. Some of them have just focused on its evaluation [170, 6], others have tried to enhance its performance [65, 83], while just few ones have addressed its security and trust aspects [154, 114].

In [65], authors tried to overcome the limitations of the standardized RPL OFs providing thereof QoS guarantees for LLNs while considering several routing metrics. However the security aspect was not considered within the proposed approach keeping as a consequence thereof the routing protocol under threat of attacks.

This threat analysis was presented in [114] where authors have detailed and classified different attacks that could be initiated against the RPL protocol according to the attacker's goal as well as the network element to be impacted.

To secure the communication in an RPL based network, authors in [153] used the trusted Platform Module to establish trustworthiness of nodes before exchanging keying material. While in [56], authors have proposed to strengthen RPL by adding a new trustworthiness metric during the construction and the maintenance phases of its instances.

However, they have not proven the defense of the proposed scheme against attacks that could be launched.
For this reason, an amelioration was proposed in [57] which takes into account trust along the path using collaborative trustworthiness evaluation between the different nodes.
In [7], authors have presented Sectrust, a lightweight secure trust based routing framework for IoT nodes. The trust evaluation is based specifically on the successful interactions between IoT nodes and its calculation is based on some metrics such as the prospect of the positive interaction between the different nodes, their satisfaction and their energy level as well. However, although the proposed framework was designed to isolate common routing attacks, its effectiveness under these threats has not been proven nor evaluated.
Discussion: As seen, current IoT research has not yet fully and comprehensively investigated how to secure the routing processes in RPL based networks especially those related to the routing topology construction and maintenance phases, the communication establishment and progress and above all how to trust the participating entities and how to secure the network against the different threats and attacks this protocol is exposed to.
In this context, several issues need to be seriously considered and more investigated. On the one hand, the trust integration within the routing functions could affect the performance of the routing protocol as longer paths could be selected to avoid malicious nodes and thus it could cause a more important delay and energy. On the other hand, more research is required in terms of QoS consideration, and attacks resiliency issues within the routing protocols. In fact proposed schemes are generally designed to defend a specific class of attacks while trust could deal with various kind of threats and meeting both the energy and the QoS requirements.
For this purpose, an inspiration could be taken from other similar areas to IoT such as MANETs and WSNs where extensive research has been carried out and several approaches have been proposed regarding trust management for routing procedures.

6.2.2 Problem statement and design objectives

In this section, we will provide a brief representation of the important properties to be considered within the proposed scheme as well as the key constraints related to each one of them. Therefore, we will present the main objectives we attempt to accomplish in order to respect such considerations.

Problem statement

In the context of secure routing, a trusted route within our trust framework would satisfy the following properties:

- Trust: a route is trusted if only trusted nodes can participate in its establishment, design and maintenance. On the other side malicious and selfish nodes will be isolated and excluded from participating in the routing procedures.
  This property could be assessed accurately on the basis of the entities historical interactions and their observation of each other forwarding behavior to judge their trust degree.

- Delay awareness: a delay aware route should be able to provide low end-to-end delays. This property can be measured through the offered end-to-end delay from one source to a destination through a particular route.

- Energy Efficiency: a route is energy efficient if it is made up of nodes that have more energy than any other node. Therefore, an efficient topology construction and route selection for RPL must consider the remaining energy of the nodes to maximize the network lifetime. Obviously nodes with low battery levels should be avoided in the routing process as much as possible.

- Reliability: a route is reliable if it continuously provide available and high quality of the communication links along the path. This property can be evaluated through the link quality estimators, such as (i) the Packet Reception Ratio (PRR), (ii) the Packet Error Rate (PER) representing the number of incorrectly received data packets divided by the total number of received packets, (iii) the Expected Transmission count (ETX) estimating the predicted number of transmissions required to send packets over a link including re-transmissions.

Main objectives

Taking into account these different issues, a brief description of the main objectives to meet is provided hereafter.

1. **Real time Assessment:** Entities behavior and interactions are monitored in real-time to collect trust related information which will further support malicious behaviors detection.

2. **QoS optimization:** our goal here is to optimize the offered QoS taking into account the different QoS parameters such as the delay and the link quality estimators particularly the PRR, PER and ETX estimators. Hence we need a minimized delay, a minimized error rate, a minimized transmission count and a maximized reception rate.

3. **Network life time maximization:** a network life time could be defined as the time taken until the network partition due to battery failure and power outage. To maximize such parameter, the balancing of the consumed energy across the network may be an effective solution to enhance the network lifetime.

6.2.3 Proposed scheme

In the routing context, trust relies on the fact that entities within the routing process do not act maliciously or selfishly regarding the forwarding mechanism. To cope with such kind of behavior, trust could be considered as an efficient solution to secure the routing procedure.
In this paragraph, we will describe the overall architecture of our trust framework. To efficiently compute trust values and to effectively integrate them within the routing procedures, we first need to clearly understand the main meaning of trust as well as the detailed composition of our system. Furthermore, a brief description of the different blocks and the required interactions to be established will be provided.

Overview

The main objective of this work is to propose a novel and multidimensional trust management system for the RPL routing protocol. A new objective function based on our trust model is therefore integrated within the routing protocol and used for its topology construction and maintenance phases. More specifically, our framework aims to define and evaluate a trust score for each entity as well as for the link it is connected through. The evaluated score is then included in the DIO message and used in the rank computation process for the selection of the preferred parent within the DODAG structure.

Trust definition

A trust management system is often needed to produce reaction based on the real time evaluation of neighboring entities behaviors during established interactions in addition to feedbacks and recommendations gathered from indirect neighbors. These last aggregated together form an overall trust score that once shared and propagated over the network, participating entities could isolate malicious ones and consider secure and trustworthy routing paths for their communications.
In our proposal, trust is defined as a relationship between two entities, a trustor and a trustee. The trustor is the evaluating entity willing to join the DAG structure or to update its preferred parent in order to send its data packets. On the other side, the trustee is the evaluated entity which represents the candidate entity that would be chosen as the next hop to the root. This relationship is restricted to a time value, that is, the time in which the evaluation has been carried out. Moreover, this relationship is derived from direct observations and interactions referred as the direct trust and the recommendations exchanged between neighboring nodes termed as the indirect trust.

Proposed model detailed design

A description of the different phases of the proposed model is presented in Fig. 6.10. This model involves a cyclic succession of operations namely topology creation, authentication, information gathering, trust composition, trust storage, nodes filtering and trust application. It includes as well two main components: an authenticator entity plus a trust manager entity, and four dimensions specifically QoS dimension, Energy awareness dimension, Reputation dimension and Security dimension. These entities and dimensions will be detailed in the next two paragraphs.

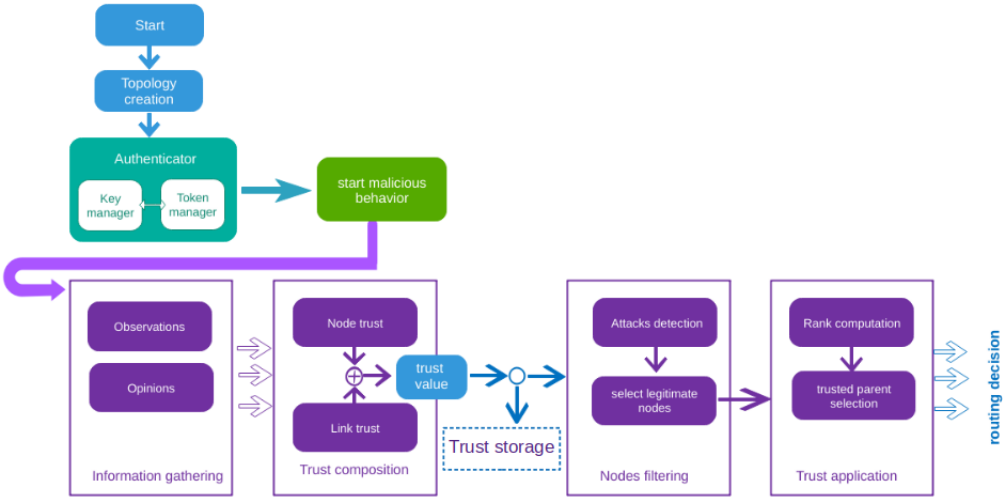

Figure 6.10 – Trust model operational blocks

This trust model and as illustrated in Fig. **??** is based on five main operational phases namely information gathering, trust composition, trust storage, nodes filtering and trust application.

(i) Information gathering: Before being able to produce trust related evidence, each entity has to gather enough information about its neighbors' behavior as well as links' quality indicators.

The trust structure to be sent to the trust manager is made up of the following information: node ID, neighbor ID, RE percentage, PFR value, ETX value, PRR value, PER value, the transmission delay as well as the entity time.

(ii) Trust composition: Upon receiving trust related information, the trust manager starts the trust composition process consisting of computing the trust score based on Node Trust (NT) and Link Trust (LT). This value is the weighted average of two parts as follow:

$$\widehat{T}_{ij}(t) = \begin{cases} 0 \;\; if(alert\ generated) \\ w1 * \widehat{NT}_{ij}(t) + w2 * \widehat{LT}_{ij}(t) else \end{cases}$$

$\widehat{T}_{ij}(t)$ represents the trust score an entity e_i has for e_j at time t. This score is limited to a continuous range from 0 to 1, where 0 denotes complete distrust whereas 1 represents

absolute trust.
$\widehat{NT}_{ij}(t)$ represents the NT level calculated based on node cooperativeness, node competence and its community of interest.
$\widehat{LT}_{ij}(t)$ denotes the LT which is assessed based on link quality and link performance.
The weight factors w1 and w2 are assigned to $\widehat{NT}_{ij}(t)$ and $\widehat{NT}_{ij}(t)$ respectively where $w1 + w2 = 1; 0 \leq w1 \leq 1$ and $0 \leq w2 \leq 1$.
Each computation is based on a set of properties where $\widehat{NT}_{ij}(t)$ represents the NT level calculated based on the trustor's direct observation of its one hop neighbors' behavior referred as the direct trust $\widehat{NT}^{d}_{ij}(t)$ and on the other hand, on the third parties' attributed recommendations called the indirect or the relative trust $\widehat{NT}^{r}_{ij}(t)$ as follow:
$\widehat{NT}_{ij}(t) = w_d * \widehat{NT}^{d}_{ij}(t) + w_r * \widehat{NT}^{r}_{ij}(t)$, w_d and w_r are the weights assigned to the direct and the indirect trust respectively.
The direct trust is calculated by considering both node cooperativeness (coop), node competence (comp) and community of interest (coi). At time t, it is defined as:
$\widehat{NT}^{d}_{ij}(t) = \widehat{NT}^{coop}_{ij}(t) * \widehat{NT}^{comp}_{ij}(t) * \widehat{NT}^{coi}_{ij}(t)$, where (a) $\widehat{NT}^{coop}_{ij}(t)$ reflects the cooperativeness level that entity e_j has as evaluated by entity e_i based on its behavior monitoring during the time interval [0..t]. It is calculated using the Packet Delivery Ratio (PDR) representing the number of packets successfully delivered to the number of those that have been sent out by the sender and the Packet Forwarding Ratio (PFR) representing the number of packets forwarded correctly to the number of those supposed transmitted and successfully received as well, (b) $\widehat{NT}^{comp}_{ij}(t)$ provides the degree of the entity's ability to perform its intended tasks within the routing process, it is assessed based on the energy and the computation capability it has in order to verify whether it is enough competent to perform its tasks or not, and last (c) $\widehat{NT}^{coi}_{ij}(t)$ provides the degree of the common interest or similar tasks of e_j as evaluated by e_i computed as the ratio of the number of common community interests over the total number of their community interests.
On the other side, the indirect trust $\widehat{NT}^{r}_{ij}(t)$ is set up upon recommendations of other entities within the neighborhood which reflects here the reputation dimension. In order to obtain trust recommendations, we first need to select trust recommenders with a trusted communication link, and thus get rid of the impact of malicious recommendations.
When it comes to the link trust $\widehat{LT}_{ij}(t)$, its evaluation aims mainly to reinforce the routing DAG establishment and maintenance processes by considering both the quality (qual) and the performance (perf) of the different links connecting the participating entities in order to successfully meet the QoS requirements, its value is defined as follow:
$\widehat{LT}_{ij}(t) = \widehat{LT}^{qual}_{ij}(t) * \widehat{LT}^{perf}_{ij}(t)$ where $\widehat{LT}^{qual}_{ij}(t)$ reflects the belief that the connecting link is efficient enough to respect the QoS guarantees. It is measured by ETX and PRR as indicators of the link quality between the entity and its neighbor. While $\widehat{LT}^{perf}_{ij}(t)$ characterize the performance of the link based on the PER and the transmission delay L.

The combination of these properties will produce an overall trust value that can be used efficiently and effectively to ensure security improvement for the RPL routing process.

(iii) Trust storage: Once the trust composition process has been completed, trust related evidence will be stored by the trust manager in a trust record table that contains apart the trust information that each entity has gathered for its candidate neighbors, the trust value computed according to the different properties as it has been already explained. To enforce the security aspect of the proposed scheme, a hash algorithm has been employed to encrypt the trust values when stored and retrieved from the trust database or when sent to the evaluating entities.
(iv) Nodes filtering: The detection and the isolation of insider attacks is the most important part within a trust framework insofar that malicious nodes are aware of every detail of the network process, they may tamper the content of transmitted packets, deny from sending messages to other legitimate nodes, they can even send fake routes to the legitimate nodes in order to get the packets or to disturb the operations. Thus a filtering phase is essential to classify network nodes and to isolate malicious ones. The filtering process is mainly based on the trust assessment where nodes whose trust score is above the trust threshold are classified as malicious, otherwise they will be considered as legitimate and thus selected to be candidates for DODAG construction and maintenance process.

(v) Trust application, a novel trust based routing metric for RPL routing protocol: The trust model previously described has been integrated into the DODAG construction and maintenance phases of RPL through the development of a new OF in order to rank nodes while calculating the most trusted path from each source to the root. To do so, each node sends to its neighbors the value of its rank which is included by default in the DIO message, once received the evaluating node e_{join} will check its record table for the most recent trust values of its $p \geq 1$ candidate parents $e_{cand1}..e_{candp}$, already sent by the trust manager. Afterwards, the rank $R(e_{join} \longmapsto e_{candq})$ will be computed according to the trust based OF and with respect to each candidate parent e_{candq}, according to the formula below:
$R(e_{join} \longmapsto e_{candq}) = R(e_{candq}) - T(e_{join} \longmapsto e_{candq})_t$
where $R(e_{candq})$ is the rank value of the candidate parent. Afterwards, the node with the minimum rank $R(e_{join} \longmapsto e_{candq})$ will be chosen as the preferred parent to reach the root.

6.2.4 Trust model validation

Simulation parameter	Value
Simulation tool	Contiki/Cooja 2.7
Mote type	Tmote Sky
Simulation run time	3600 seconds
Simulation coverage area	300m x 300m
Interference range	100m
Wireless communication range	50m
Total number of nodes	10..50
Number of malicious nodes	3..15
Radio environment	DGRM (Directed Graph Radio Medium)
Initial trust value	0.5
Trust interval	0..1

Table 6.2 – Network related parameters used in simulation analysis

Our experiments were performed using the InstantContiki 2.7 platform while integrating the proposed trust model into the RPL routing protocol. As we have noted, the InstantContiki was used as the development environment with the Cooja simulator to implement the proposed model. Let us remind that Cooja provides real environment to build IoT networks with different types of motes, and implemented code could be tested and uploaded to real motes without any modification.
The various simulation parameters are listed in Table 6.2. In this study, we have assumed that the attacking nodes behave as good nodes from inception and begin their malicious activities during time (when activated). In this model, two kinds of attacks were considered such as:

- Blackhole attack: when receiving routing packets, the malicious node will discard them instead of relaying as the protocol requires.
- Greyhole attack: The Greyhole attack presents the same attitude and behaviour as the blackhole attack. The difference is that the Greyhole malicious node does not drop whole packets, instead it selectively drops some part of them.

Performance evaluation results

In order to prove the performance of our proposal, we have performed several simulations in comparison with RPL and more specifically with RPL based on the
MRHOF, Minimum Rank with Hysteresis Objective
Function. This last uses hysteresis while selecting the path with the smallest ETX metric

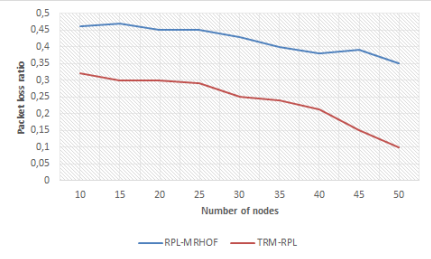

Figure 6.11 – Influence of the network size on the packet loss ratio

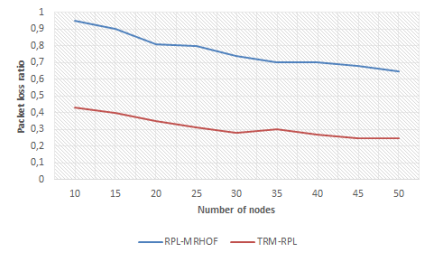

Figure 6.12 – PLR comparison between MRHOF and TRM-RPL under black hole attacks

value from the source node to the root. In addition, we have varied the number of network nodes, the percentage of malicious ones and we have analyzed the corresponding effect respectively on the Packet Loss Ratio (PLR), the Remaining Energy percentage (RE) and the resiliency to black hole attacks.

Packet loss ratio : We measure here the PLR while varying the number of legitimate and malicious nodes.
Fig. 6.11 shows the variation of the PLR with the increase of the network size while Fig. 6.12 shows its variation in the presence of a number of malicious nodes representing the quarter of the total number of network nodes.

The simulation result in Fig. 6.11 shows that the PLR decreases slightly with the increase in the network size. This is obvious as when there is a high number of neighbors, nodes are able to find alternate paths easily, which reduces packet loss. One can see that our trust framework called TRM-RPL experiences a more reduced packet loss than RPL. This is primarily due to the fact that TRM-RPL takes into consideration more link quality estimators in addition to the node attributes while evaluating the trustworthiness of each candidate parent. Minimizing the ETX, the PER and maximizing the PRR, the RE when selecting the next hop will imply a path with low PLR.
To prove the effectiveness of our trust model, we have evaluated its conduct towards black hole attacks where malicious entities dumps all packets they receive and that are supposed to be forwarded.
Fig. 6.12 displays a graphical representation of the percentage of packet loss under black-hole attacks. While TRM-RPL related PLR stayed below 0.4, the standard RPL (MRHOF) recorded a staggering one between 0.6 and 0.95.

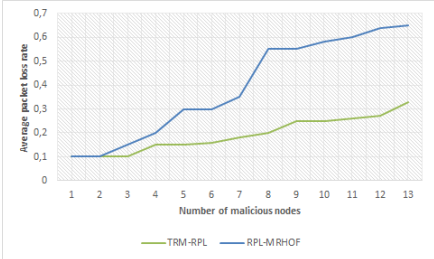

Figure 6.13 – Malicious nodes number impact on the packet loss ratio

Moreover, we have measured the impact of the number of malicious nodes launching black hole attacks on the packet loss. As shown in Fig. 6.13, we can obviously see that the PLR increases when the number of malicious nodes increases respectively for both TRM-RPL and RPL-MRHOF. However, an obvious observation of this simulation result is that TRM-RPL performs better in the presence of malicious nodes since packet loss rate under 0.2 is realized for up to 10 attacking nodes while to 6 attacking ones in RPL-MRHOF. The simulation result in Fig. 6.14 shows well how TRM-RPL detects the

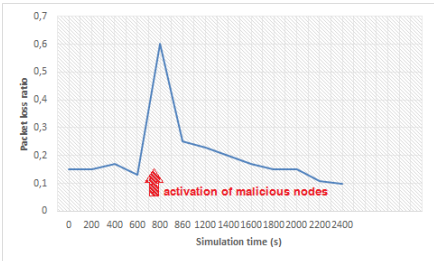

Figure 6.14 – PLR evolution of LT-RPL in a 50 nodes network size

malicious behavior of nodes and how it reacts to its presence. The network here is composed of 50 nodes where 15 among them start to act maliciously over a certain time after the network initialization. As shown in Fig. 6.14, once malicious nodes begin their malicious activities consisting of dropping their neighbors' packets, the PLR considerably increase in response to the occurring event. However the integration of the proposed scheme within RPL was shown and proven to be effective in reducing the impact of such behaviors insofar as the attacking nodes will not be chosen anymore as a next hop nodes during the network routing topology construction. As a result thereof, the PLR is reduced and maintained considerably stable over time.

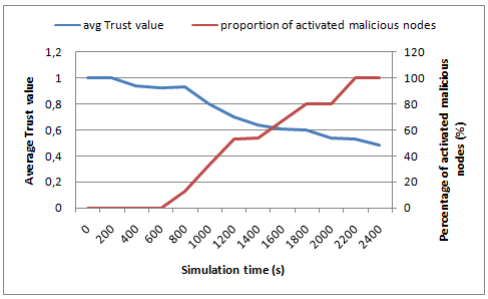

Figure 6.15 – Average trust value evolution

To clearly see the reaction of TRM-RPL in the presence of malicious nodes, we have measured the average trust value while increasing the percentage of activated malicious

nodes within a 50 nodes network size where 15 nodes are malicious as illustrated in Fig. 6.15. It is clear that since they start to behave abnormally regarding the packet forwarding behavior, malicious nodes are detected and their trust level significantly drops which justify the decrease of the average trust value of the whole network (the average value of both legitimate and malicious nodes).
Discussion: Our analysis of the packet loss has shown that TRM-RPL gives higher performance in comparison with MRHOF based RPL since it experiences a more reduced PLR even in the presence of malicious nodes launching black hole attacks. The detection, the reaction and the prevention of these behaviors have been proven as well.

Conclusion

We have presented in this chapter the design and the implementation of a secure trust management system based on the blockchain technology to collect trust evidences and to securely store and share them within and through the blockchain network. Using such technology, we enable a more reliable trust information confidentiality and integrity verification during storage and sharing and we make a time-stamped log of both entities' transactions and behavior. Our evaluation shows that our proposal is feasible, deployable and suited for IoT environments given its features of being decentralized, ensuring security and resiliency to a set of attacks and requiring a low overhead in addition to low resources. For the trust evaluation and composition process, we have proposed a novel trust scheme according to a specific use case scenario which is the RPL routing protocol. For such objective, we provided in a second part of this chapter a trust based routing metric for securing the routing procedure within the RPL protocol. Our trust model follows a multidimensional approach to enable an accurate trust assessment for IoT entities where four dimensions are taken into account namely Reputation, QoS, Energy and Security for the evaluation of the trustworthiness degree of network nodes as well as the links they are connected through. Our proposal was implemented, successfully tested and compared with the standard RPL protocol where its effectiveness and resilience to attacks has been proved to be better.

Chapter 7
Conclusion and Perspectives

Contents

This part concludes our thesis. Section 7.1 summarizes our research and reflects on the questions stated in the state of the art. Section 7.2 looks at improvements our contributions could benefit from as well as future research directions

7.1 Summary of contributions

The premise of the Internet of Things (IoT) is to interconnect not only sensors, mobile devices, and computers but also individuals, smart buildings and cities, electrical and water grids, and smart factories and industries, to mention a few. The integration of such technologies within the manufacturing environment and processes in combination with other technologies such as Cloud Computing, Cyber Physical Systems, Information and Communication Technologies as well as Enterprise Architecture, has introduced the fourth industrial revolution referred to also as Industry 4.0. However, realizing the extensive connectivity of IoT enabled industries while ensuring security and privacy still remains a challenge. In this thesis our research focused on four complementary issues, mainly (I) the dynamic and trust based management of access over shared resources within an Industry 4.0 based distributed and collaborative system, (ii) the establishement of a privacy preserving solution for related data in a decentralized architecture while eliminating the need to rely on additional third parties, (iii) the verification of the safety, the correctness and the functional accuracy of the designed framework and (iv) the evaluation of the trustworthiness degree of interacting parties in addition to the secure storage and sharing of computed trust scores among them in order to guarantee their confidentiality, integrity and privacy.
By focusing on such issues and taking into account the conventional characteristics of both IoT and IoT enabled industries environments, we proposed in this thesis a secure and distributed framework for resource management in Industry 4.0 environments. The proposed framework, enabled by the blockchain technology and driven by peer to peer networks, comprises three main pillars: (1) the decentralized, dynamic and trust based access management over resources shared among collaborating parties and industries, (2) Privacy preserving guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies and finally (3) the distributed and secure sharing and storage of trust information within the IoT network via blockchain.
We briefly summarize our contributions as follows:

- First Contribution: we proposed a Distributed Resource Management Framework for Industry 4.0 Environments. The proposed framework based on the blockchain technology aim to keep a living document trace about the flow of resources being distributed and shared among collaborating parties while using the OrBAC access control model to implement distributed, fine grained, flexible and secure resource

access authorization through the design and the implementation of Ethereum based smart contracts.

Moreover, and in order to better support the security requirement, this framework adds the notion of trust management to the access control model where a trust framework is integrated to evaluate access requester entities' behavior. Finally our proposal is conceived to support distributed and dynamic governance of the system through the registration of new entities requesting mining permissions.

- Second Contribution: We ensured strong privacy guarantees over the access control related procedures regarding the access requester sensitive attributes as well as the shared access control policies. The proposed scheme is integrated within our DRMF framework to preserve the anonymity of both the access requester entities as well as the collaborating parties, by this way the transparency feature of our framework will be maintained while guaranteeing and preserving the privacy of its users.

- Third Contribution: As a third contribution, we reasoned mainly about the correctness, the safety and the functional accuracy of smart contracts before their deployment on the blockchain network. For this contribution, we used Event-B formal verification method to formally model written smart contracts in order to verify and validate their safety, correctness and functional accuracy in addition to their compliance with their specification for given behaviors. To illustrate the proposed approach, its application to a realistic industrial use case was described.

- Fourth Contribution: We proposed a secure trust management system based on the blockchain technology to provide tamper proof data, enable a more reliable trust information integrity verification, and help to enhance its privacy and availability during sharing and storage. For this purpose, we designed a blockchain based trust architecture to collect trust evidences, to define a trust score for each device and to securely store and share them with other devices within the network by embedding them into blockchain transactions. For assessing the trustworthiness degree of each entity within the network and deriving a trust score to be shared between involved entites and considered within the decision making process, a generic trust mechanism was defined. For the evaluation of this last, we have chosen to use our solution within a specific use case which is the RPL routing protocol to ensure secure routing by protecting the network against misbehaving, selfish and malicious nodes regarding the routing procedure.

7.2 Future research directions

However, there are still many aspects that need to be further explored to improve the entire infrastructure for both IoT and IIoT environments. In this section, we identify and analyze some future research directions.

7.2.1 Access rules privacy

Although collaborating parties could have full and decentralized control over the system and resources shared among them while preserving in a strong way their anonymity and sensitive attributes during the access control procedure, the policy public visibility and the privacy of shared access rules in the blockchain is still an issue. Here blockchain based solutions generally require the submission of access control policies directly into the blockchain to ensure verifiable-consistency, immutability and notarization. Unfortunately, this will also reveal all the access control policies to the public, meaning that anyone can learn the required policies to access resources even when they are not authorized. This will further leak entities sensitive information beyond the inferred metadata from accessible data. Hence confidential policies should be enabled where sensitive access control policies should be revealed just to authorized parties.

7.2.2 Interoperability

Even though the blockchain technology enables decentralized applications and platforms with interesting properties of transparency, auditability or censorship resistance, the technology faces many challenges, specifically regarding the interoperability. This last may concern several levels: between projects on a same blockchain, between different blockchains, or between blockchains and existing systems. This is due to the fact each blockchain uses its own standard and economic model. As a future work direction, we hope to improve the interoperability of our decentralized blockchain framework. It involves developing models, methods and formal tools for designing decentralized applications. In particular, the following aspects could be considered: Semantic representation of entities and resources, Storage of these objects, Exchange protocols of these objects between users, Validation of their integrity, Their integration in the application, ensuring interoperability and scalability of the application.

7.2.3 Consensus algorithms benchmarking

Alongside this thesis we have been interested in using the blockchain as an enabling technology for supporting a secure, distributed, resilient, fault tolerant, and censorship resistant approach to both IoT and IIoT networking. For consortium blockchain based systems several consensus algorithms could be considered. However the evaluation of

these algorithms is actually challenging in adversarial environment. Indeed to be secure, a protocol should comes out with clear security model and assumptions, under which the protocol proceeds as expected guaranteeing safety and liveness properties. Actually most of the modern consensus mechanisms comes out without these information, hence it is difficult to evaluate their employability. In this context, as a future direction for this work and in order to evaluate blockchain systems, we highlight the need of a general purpose benchmark who measure performances of these last under specific security and safety related assumptions. Hence we could define a framework for benchmarking blockchain consenus mechanisms and evaluating them in a more formal approach.

Bibliography

[1] Tesnim Abdellatif and Kei-Léo Brousmiche. "Formal verification of smart contracts based on users and blockchain behaviors models". In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. 2018, pp. 1–5.

[2] Oumaima Ben Abderrahim, Mohamed Houcine Elhdhili, and Leila Saidane. "TMCoI-SIOT: A trust management system based on communities of interest for the social Internet of Things". In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2017, pp. 747–752.

[3] Saveen A Abeyratne and Radmehr P Monfared. "Blockchain ready manufacturing supply chain using distributed ledger". In: *International Journal of Research in Engineering and Technology* 5.9 (2016), pp. 1–10.

[4] Jean-Raymond Abrial. *Modeling in Event-B: system and software engineering*. Cambridge University Press, 2010.

[5] Jean-Raymond Abrial and Jean-Raymond Abrial. *The B-book: assigning programs to meanings*. Cambridge University Press, 2005.

[6] Nicola Accettura et al. "Performance analysis of the RPL routing protocol". In: *2011 IEEE International Conference on Mechatronics*. IEEE. 2011, pp. 767–772.

[7] David Airehrour, Jairo Gutierrez, and Sayan Kumar Ray. "A lightweight trust design for IoT routing". In: *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE. 2016, pp. 552–557.

[8] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications". In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[9] Fadele Ayotunde Alaba et al. "Internet of Things security: A survey". In: *Journal of Network and Computer Applications* 88 (2017), pp. 10–28.

[10] Muhammad Salek Ali et al. "Applications of blockchains in the Internet of Things: A comprehensive survey". In: *IEEE Communications Surveys & Tutorials* 21.2 (2018), pp. 1676–1717.

[11] I.F. All. "The 5 Worst Examples of Iot Hacking and Vulnerabilities in Recorded History". In: *URl: https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities/* (2017).

[12] Allseen Alliance. "Alljoyn framework". In: *Linux Foundation Collaborative Projects. URl: https://allseenalliance. org/framework (visited on 09/19* (2016).

[13] Sidney Amani et al. "Towards verifying ethereum smart contract bytecode in Isabelle/HOL". In: *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*. ACM. 2018, pp. 66–77.

[14] Donovan Artz and Yolanda Gil. "A survey of trust in computer science and the semantic web". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007), pp. 58–71.

[15] Shehzad A Ashraf et al. "Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G". In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2016, pp. 1–8.

[16] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. "A survey of attacks on ethereum smart contracts (sok)". In: *International conference on principles of security and trust*. Springer. 2017, pp. 164–186.

[17] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "Siot: Giving a social structure to the internet of things". In: *IEEE communications letters* 15.11 (2011), pp. 1193–1195.

[18] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[19] Rita Azzi, Rima Kilany Chamoun, and Maria Sokhn. "The power of a blockchain-based supply chain". In: *Computers & industrial engineering* 135 (2019), pp. 582–592.

[20] Arati Baliga. "Understanding blockchain consensus models". In: *Persistent* 2017.4 (2017), pp. 1–14.

[21] Debasis Bandyopadhyay and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization". In: *Wireless personal communications* 58.1 (2011), pp. 49–69.

[22] Fenye Bao and Ing-Ray Chen. "Dynamic trust management for internet of things applications". In: *Proceedings of the 2012 international workshop on Self-aware internet of things*. ACM. 2012, pp. 1–6.

[23] Fenye Bao and Ing-Ray Chen. "Dynamic trust management for internet of things applications". In: *Proceedings of the 2012 international workshop on Self-aware internet of things*. ACM. 2012, pp. 1–6.

[24] Fenye Bao, Ray Chen, and Jia Guo. "Scalable, adaptive and survivable trust management for community of interest based internet of things systems". In: *2013 IEEE eleventh international symposium on autonomous decentralized systems (ISADS)*. Citeseer. 2013, pp. 1–7.

[25] Ali Vatankhah Barenji, Zhi Li, and Wai Ming Wang. "Blockchain cloud manufacturing: Shop floor and machine level". In: *Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies*. VDE. 2018, pp. 1–6.

[26] Ali Vatankhah Barenji et al. "Blockchain-Based Cloud Manufacturing: Decentralization". In: *arXiv preprint arXiv:1901.10403* (2019).

[27] Arpita Bhargava, Shekhar Verma, and Brijesh Kumar Chaurasia. "Kalman filter for trust estimation in VANETs". In: *2015 IEEE UP Section Conference on Electrical Computer and Electronics (UPCON)*. IEEE. 2015, pp. 1–6.

[28] Karthikeyan Bhargavan et al. "Formal verification of smart contracts: Short paper". In: *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*. ACM. 2016, pp. 91–96.

[29] Giancarlo Bigi et al. "Validation of decentralised smart contracts through game theory and formal methods". In: *Programming Languages with Applications to Biology and Security*. Springer, 2015, pp. 142–161.

[30] IBM Blockchain. "IBM Food Trust-A new era for the world's food supply". In: *URL: https://www.ibm.com/downloads/cas/8QABQBDR* 17 (2019).

[31] Carsten Bormann, Angelo P Castellani, and Zach Shelby. "Coap: An application protocol for billions of tiny internet nodes". In: *IEEE Internet Computing* 2 (2012), pp. 62–67.

[32] Benno Bunse, Henning Kagermann, and Wolfgand Wahlster. "Industrie 4.0-smart manufacturing for the future". In: *GTIA-Ger. Trade Invest* 40 (2013), pp. 12–23.

[33] Lin Cai and Roberto Rojas-Cessa. "Containing sybil attacks on trust management schemes for peer-to-peer networks". In: *2014 IEEE International Conference on Communications (ICC)*. IEEE. 2014, pp. 841–846.

[34] Quyet H Cao et al. "A trust model for data sharing in smart cities". In: *2016 IEEE International Conference on Communications (ICC)*. IEEE. 2016, pp. 1–7.

[35] Licia Capra and Mirco Musolesi. "Autonomic trust prediction for pervasive systems". In: *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)*. Vol. 2. IEEE. 2006, 5–pp.

[36] Dong Chen et al. "TRM-IoT: A trust management model based on fuzzy reputation for internet of things." In: *Comput. Sci. Inf. Syst.* 8.4 (2011), pp. 1207–1228.

[37] Dong Chen et al. "TRM-IoT: A trust management model based on fuzzy reputation for internet of things." In: *Comput. Sci. Inf. Syst.* 8.4 (2011), pp. 1207–1228.

[38] Ray Chen, Fenye Bao, and Jia Guo. "Trust-based service management for social internet of things systems". In: *IEEE transactions on dependable and secure computing* 13.6 (2015), pp. 684–696.

[39] Ray Chen, Jia Guo, and Fenye Bao. "Trust management for SOA-based IoT and its application to service composition". In: *IEEE Transactions on Services Computing* 9.3 (2014), pp. 482–495.

[40] Ray Chen, Jia Guo, and Fenye Bao. "Trust management for SOA-based IoT and its application to service composition". In: *IEEE Transactions on Services Computing* 9.3 (2014), pp. 482–495.

[41] Zhikui Chen et al. "A scheme of access service recommendation for the Social Internet of Things". In: *International Journal of Communication Systems* 29.4 (2016), pp. 694–706.

[42] Jin-Hee Cho, Ananthram Swami, and Ray Chen. "A survey on trust management for mobile ad hoc networks". In: *IEEE Communications Surveys & Tutorials* 13.4 (2010), pp. 562–583.

[43] Konstantinos Christidis and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things". In: *Ieee Access* 4 (2016), pp. 2292–2303.

[44] Edmund M Clarke and E Allen Emerson. "Design and synthesis of synchronization skeletons using branching time temporal logic". In: *Workshop on Logic of Programs*. Springer. 1981, pp. 52–71.

[45] Edmund M Clarke Jr et al. *Model checking*. MIT press, 2018.

[46] "ClearSy: Atelier B: B System". In: ClearSy, 2019.

[47] Pamel Cobb. "German steel mill meltdown: Rising stakes in the internet of things". In: *Security Intelligence* (2015).

[48] Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. "Blockchain for the Internet of Things: A systematic literature review". In: *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE. 2016, pp. 1–6.

[49] Corero. "The Mirai Botnet: All About the Latest Malware DDoS Attack Type". In: *URl: https://www.corero.com/resources/ddos-attacktypes/mirai-botnet-ddos-attack.html* (2016).

[50] Michael Crosby et al. "Blockchain technology: Beyond bitcoin". In: *Applied Innovation* 2.6-10 (2016), p. 71.

[51] Jason Paul Cruz, Yuichi Kaji, and Naoto Yanai. "RBAC-SC: Role-based access control using smart contract". In: *IEEE Access* 6 (2018), pp. 12240–12251.

[52] Gaby G Dagher et al. "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology". In: *Sustainable Cities and Society* 39 (2018), pp. 283–297.

[53] Chris Dannen. *Introducing Ethereum and Solidity*. Vol. 1. Springer, 2017.

[54] Dean. "51% attack". In: *URL: http://cryptorials.io/glossary/51-attack/* (2015).

[55] Richard Dennis and Gareth Owen. "Rep on the block: A next generation reputation system based on the blockchain". In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE. 2015, pp. 131–138.

[56] Nabil Djedjig, Djamel Tandjaoui, and Faiza Medjek. "Trust-based RPL for the Internet of Things". In: *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE. 2015, pp. 962–967.

[57] Nabil Djedjig et al. "New trust metric for the RPL routing protocol". In: *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2017, pp. 328–335.

[58] Ashutosh Dhar Dwivedi et al. "A decentralized privacy-preserving healthcare blockchain for iot". In: *Sensors* 19.2 (2019), p. 326.

[59] Dave Evans. *The Internet of Things, How the Next Evolution of the Internet Is Changing Everything*. Cisco IBSG, 2011.

[60] Qi Feng et al. "A survey on privacy protection in blockchain system". In: *Journal of Network and Computer Applications* (2018).

[61] David Ferraiolo, Janet Cugini, and D Richard Kuhn. "Role-based access control (RBAC): Features and motivations". In: *Proceedings of 11th annual computer security application conference*. 1995, pp. 241–48.

[62] Audit Tax Consulting Corporate Finance. "Industry 4.0 Challenges and solutions for the digital transformation and use of exponential technologies". In: *Finance, Audit Tax Consulting Corporate: Zurich, Swiss* (2015).

[63] Peter Friess. *Digitising the industry-internet of things connecting the physical, digital and virtual worlds*. River Publishers, 2016.

[64] Christian Fuhrhop, John Lyle, and Shamal Faily. "The webinos project". In: *Proceedings of the 21st international conference on World Wide Web*. 2012, pp. 259–262.

[65] Olfa Gaddour, Anis Koubâa, and Mohamed Abid. "Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL". In: *Ad Hoc Networks* 33 (2015), pp. 233–256.

[66] "Geth Client for Building Private Blockchain Networks". In: *URL: https://github.com/ethereum/go-ethereum/wiki/geth* (2019).

[67] Kannan Govindan and Prasant Mohapatra. "Trust computations and trust dynamics in mobile adhoc networks: A survey". In: *IEEE Communications Surveys & Tutorials* 14.2 (2011), pp. 279–298.

[68] Kannan Govindan and Prasant Mohapatra. "Trust computations and trust dynamics in mobile adhoc networks: A survey". In: *IEEE Communications Surveys & Tutorials* 14.2 (2011), pp. 279–298.

[69] Tyrone Grandison and Morris Sloman. "A survey of trust in internet applications". In: *IEEE Communications Surveys & Tutorials* 3.4 (2000), pp. 2–16.

[70] Gideon Greenspan. "Multichain private blockchain-white paper". In: *URl: http://www.multichain. com/download/MultiChain-White-Paper. pdf* (2015).

[71] Jia Guo, Ray Chen, and Jeffrey JP Tsai. "A survey of trust computation models for service management in internet of things systems". In: *Computer Communications* 97 (2017), pp. 1–14.

[72] Sergio Gusmeroli, Salvatore Piccione, and Domenico Rotondi. "A capability-based security approach to manage access control in the internet of things". In: *Mathematical and Computer Modelling* 58.5-6 (2013), pp. 1189–1205.

[73] Dick Hardt. "The OAuth 2.0 authorization framework". In: (2012).

[74] Dick Hardt et al. *The OAuth 2.0 authorization framework*. Tech. rep. RFC 6749, October, 2012.

[75] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions". In: *Future Generation Computer Systems* 97 (2019), pp. 512–529.

[76] Vincent C Hu et al. "Attribute-based access control". In: *Computer* 48.2 (2015), pp. 85–88.

[77] Junqin Huang et al. "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism". In: *IEEE Transactions on Industrial Informatics* 15.6 (2019), pp. 3680–3689.

[78] HS Jennath, S Adarsh, and VS Anoop. "Distributed IoT and Applications: A Survey". In: *Integrated Intelligent Computing, Communication and Security*. Springer, 2019, pp. 333–341.

[79] Audun Jøsang, Roslan Ismail, and Colin Boyd. "A survey of trust and reputation systems for online service provision". In: *Decision support systems* 43.2 (2007), pp. 618–644.

[80] Anas Abou El Kalam et al. "Organization based access control". In: *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*. IEEE. 2003, pp. 120–131.

[81] Sukrit Kalra et al. "ZEUS: Analyzing Safety of Smart Contracts." In: *NDSS*. 2018.

[82] Ghassan O Karame et al. "Misbehavior in bitcoin: A study of double-spending and accountability". In: *ACM Transactions on Information and System Security (TISSEC)* 18.1 (2015), pp. 1–32.

[83] Panagiotis Karkazis et al. "Design of primary and composite routing metrics for RPL-compliant wireless sensor networks". In: *2012 International Conference on Telecommunications and Multimedia (TEMU)*. IEEE. 2012, pp. 13–18.

[84] Rafiullah Khan et al. "Future internet: the internet of things architecture, possible applications and key challenges". In: *2012 10th international conference on frontiers of information technology*. IEEE. 2012, pp. 257–260.

[85] DH Knight and NL Chervany. "The meaning of trust". In: *Tech. Rep.* (1996).

[86] Nikolay Korovaiko and Alex Thomo. "Trust prediction from user-item ratings". In: *Social Network Analysis and Mining* 3.3 (2013), pp. 749–759.

[87] Ahmed Kosba et al. "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts". In: *2016 IEEE symposium on security and privacy (SP)*. IEEE. 2016, pp. 839–858.

[88] Sathish Alampalayam Kumar, Tyler Vealey, and Harshit Srivastava. "Security in internet of things: Challenges, solutions and future directions". In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE. 2016, pp. 5772–5781.

[89] Nandakishore Kushalnagar, Gabriel Montenegro, Christian Schumacher, et al. "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals". In: (2007).

[90] Jun-Ya Lee, Wei-Cheng Lin, and Yu-Hung Huang. "A lightweight authentication protocol for internet of things". In: *2014 International Symposium on Next-Generation Electronics (ISNE)*. IEEE. 2014, pp. 1–2.

[91] Michael Leuschel and Michael Butler. "ProB: A model checker for B". In: *International Symposium of Formal Methods Europe*. Springer. 2003, pp. 855–874.

[92] Michael Leuschel and Michael Butler. "ProB: an automated analysis toolset for the B method". In: *International Journal on Software Tools for Technology Transfer* 10.2 (2008), pp. 185–203.

[93] Xiaomin Li et al. "A review of industrial wireless networks in the context of industry 4.0". In: *Wireless networks* 23.1 (2017), pp. 23–41.

[94] Xueping Liang et al. "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability". In: *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE. 2017, pp. 468–477.

[95] Chao Lin et al. "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0". In: *Journal of Network and Computer Applications* 116 (2018), pp. 42–52.

[96] Fang Liu et al. "NIST cloud computing reference architecture". In: *NIST special publication* 500.2011 (2011), p. 292.

[97] Jiajun Liu, Pingyu Jiang, and Jiewu Leng. "A framework of credit assurance mechanism for manufacturing services under social manufacturing context". In: *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE. 2017, pp. 36–40.

[98] Jing Liu and Zhentian Liu. "A survey on security verification of blockchain smart contracts". In: *IEEE Access* 7 (2019), pp. 77894–77904.

[99] Dave Locke. "Mq telemetry transport (mqtt) v3. 1 protocol specification". In: *IBM developerWorks Technical Library* (2010), p. 15.

[100] Francesco Longo et al. "Blockchain-enabled supply chain: An experimental study". In: *Computers & Industrial Engineering* 136 (2019), pp. 57–69.

[101] Yang Lu. "Industry 4.0: A survey on technologies, applications and open research issues". In: *Journal of industrial information integration* 6 (2017), pp. 1–10.

[102] Eng Keong Lua et al. "A survey and comparison of peer-to-peer overlay network schemes". In: *IEEE Communications Surveys & Tutorials* 7.2 (2005), pp. 72–93.

[103] C Lundkvist et al. "Uport: a platform for self-sovereign identity, Draft Version (2016-10-20)". In: *NY: Consensys* (2016).

[104] Loi Luu et al. "Making smart contracts smarter". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 254–269.

[105] Jiazhuo Lyu et al. "A Secure Decentralized Trustless E-Voting System Based on Smart Contract". In: *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2019, pp. 570–577.

[106] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. "Blockchain based access control". In: *IFIP international conference on distributed applications and interoperable systems*. Springer. 2017, pp. 206–220.

[107] Parikshit N Mahalle et al. "Identity authentication and capability based access control (iacac) for the internet of things". In: *Journal of Cyber Security and Mobility* 1.4 (2013), pp. 309–348.

[108] Rwan Mahmoud et al. "Internet of things (IoT) security: Current status, challenges and prospective measures". In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE. 2015, pp. 336–341.

[109] "Making Supply Chains Work Together". In: *URL: https://origintrail.io/* (2019).

[110] Carmen Martinez-Cruz et al. "A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling". In: *Information Sciences* 311 (2015), pp. 102–118.

[111] Greg Maxwell. "Confidential transactions (2015)". In: (2016).

[112] Gregory Maxwell. "Coinjoin: Bitcoin privacy for the real world". In: *URl: https://bitcointalk. org/index. php* (2013).

[113] Hartwig Mayer. "ECDSA security in bitcoin and ethereum: a research survey". In: *CoinFaabrik, June* 28 (2016), p. 126.

[114] Anthea Mayzaud, Remi Badonnel, and Isabelle Chrisment. "A Taxonomy of Attacks in RPL-based Internet of Things". In: (2016).

[115] Dennis Miller. "Blockchain and the internet of things in the industrial sector". In: *IT Professional* 20.3 (2018), pp. 15–18.

[116] Daniele Miorandi et al. "Internet of things: Vision, applications and research challenges". In: *Ad hoc networks* 10.7 (2012), pp. 1497–1516.

[117] Axel Moinet, Benoît Darties, and Jean-Luc Baril. "Blockchain based trust & authentication for decentralized sensor networks". In: *arXiv preprint arXiv:1706.01730* (2017).

[118] Saikat Mondal et al. "Blockchain inspired RFID-based information architecture for food supply chain". In: *IEEE Internet of Things Journal* 6.3 (2019), pp. 5803–5813.

[119] Malte Möser et al. "An empirical analysis of traceability in the monero blockchain". In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 143–163.

[120] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. "A computational model of trust and reputation". In: *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. IEEE. 2002, pp. 2431–2439.

[121] Einar Mykletun, Joao Girao, and Dirk Westhoff. "Public key based cryptoschemes for data concealment in wireless sensor networks". In: *2006 IEEE International Conference on Communications*. Vol. 5. IEEE. 2006, pp. 2288–2295.

[122] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. Tech. rep. Manubot, 2019.

[123] Zeinab Nehai, Pierre-Yves Piriou, and Frederic Daumas. "Model-checking of smart contracts". 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)*. IEEE. 2018, pp. 980–987.

[124] Michele Nitti, Roberto Girau, and Luigi Atzori. "Trustworthiness management in the social internet of things". In: *IEEE Transactions on knowledge and data engineering* 26.5 (2013), pp. 1253–1266.

[125] Michele Nitti, Roberto Girau, and Luigi Atzori. "Trustworthiness management in the social internet of things". In: *IEEE Transactions on knowledge and data engineering* 26.5 (2013), pp. 1253–1266.

[126] Shen Noether, Adam Mackenzie, et al. "Ring confidential transactions". In: *Ledger* 1 (2016), pp. 1–18.

[127] J David Nuñez-Gonzalez, Manuel Graña, and Bruno Apolloni. "Reputation features for trust prediction in social networks". In: *Neurocomputing* 166 (2015), pp. 1–7.

[128] Sylvia Osborn. "Mandatory access control and role-based access control revisited". In: *Proceedings of the second ACM workshop on Role-based access control*. 1997, pp. 31–40.

[129] Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman. "FairAccess: a new Blockchain-based access control framework for the Internet of Things". In: *Security and Communication Networks* 9.18 (2016), pp. 5943–5964.

[130] Aafaf Ouaddah et al. "Access control in the Internet of Things: Big challenges and new opportunities". In: *Computer Networks* 112 (2017), pp. 237–262.

[131] Pascal Paillier. "Public-key cryptosystems based on composite degree residuosity classes". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1999, pp. 223–238.

[132] Gerardo Pardo-Castellote. *OMG data-distribution service (DDS): Architectural overview*. Tech. rep. REAL-TIME INNOVATIONS INC SUNNYVALE CA, 2004.

[133] Torben Pryds Pedersen. "Non-interactive and information-theoretic secure verifiable secret sharing". In: *Annual International Cryptology Conference*. Springer. 1991, pp. 129–140.

[134] Sven Plaga et al. "Securing future decentralised industrial IoT infrastructures: Challenges and free open source solutions". In: *Future Generation Computer Systems* 93 (2019), pp. 596–608.

[135] Amir Pnueli. "The temporal logic of programs". In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. IEEE. 1977, pp. 46–57.

[136] Sherif Emad Abdel Rafey, Ayman Abdel-Hamid, and Mohamad Abou El-Nasr. "CBSTM-IoT: Context-based social trust model for the Internet of Things". In: *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*. IEEE. 2016, pp. 1–8.

[137] Lei Ren et al. "Cloud manufacturing: key characteristics and applications". In: *International journal of computer integrated manufacturing* 30.6 (2017), pp. 501–515.

[138] Karen Renaud and Dora Gálvez-Cruz. "Privacy: Aspects, definitions and a multifaceted privacy preservation approach". In: *2010 Information Security for South Africa*. IEEE. 2010, pp. 1–8.

[139] Ronald L Rivest, Adi Shamir, and Yael Tauman. "How to leak a secret". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2001, pp. 552–565.

[140] Rodrigo Roman, Jianying Zhou, and Javier Lopez. "On the features and challenges of security and privacy in distributed internet of things". In: *Computer Networks* 57.10 (2013), pp. 2266–2279.

[141] T Ruffing, P Moreno-Sanchez, and A Kate. "Coinshuffle: Practical decentralized coin mixing for bitcoin". In: *European Symposium on Research in Computer Security*. Springer. 2014, pp. 345–364.

[142] Yosra Ben Saied et al. "Trust management system design for the Internet of Things: A context-aware and multi-service approach". In: *Computers & Security* 39 (2013), pp. 351–365.

[143] Peter Saint-Andre. "Extensible messaging and presence protocol (XMPP): Core". In: (2011).

[144] Nat Sakimura et al. "OpenID Connect Core 1.0 incorporating errata set 1". In: *The OpenID Foundation, specification* (2014).

[145] Nat Sakimura et al. "OpenID Connect Core 1.0 incorporating errata set 1". In: *The OpenID Foundation, specification* 335 (2014).

[146] Ravi Sandhu and Qamar Munawer. "How to do discretionary access control using roles". In: *Proceedings of the third ACM workshop on Role-based access control*. 1998, pp. 47–54.

[147] Ravi S Sandhu et al. "Role-based access control models". In: *Computer* 29.2 (1996), pp. 38–47.

[148] Eli Ben Sasson et al. "Zerocash: Decentralized anonymous payments from bitcoin". In: *2014 IEEE Symposium on Security and Privacy*. IEEE. 2014, pp. 459–474.

[149] Abylay Satybaldy, Mariusz Nowostawski, and Jørgen Ellingsen. "Self-Sovereign Identity Systems Evaluation framework". In: ().

[150] K Schwab. "The Fourth Industrial Revolution by Klaus Schwab". In: *Geneva: World Economic Forum*. 2016.

[151] T Scott. "TradeLens: How IBM and Maersk Are Sharing Blockchain to Build a Global Trade Platform". In: 12 (2018).

[152] SECG SEC. "2: Recommended elliptic curve domain parameters". In: *Standards for Efficient Cryptography Group, Certicom Corp* (2000).

[153] Sebastian Seeber et al. "Towards a trust computing architecture for rpl in cyber physical systems". In: *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. IEEE. 2013, pp. 134–137.

[154] Anuj Sehgal et al. "Addressing DODAG inconsistency attacks in RPL networks". In: *2014 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE. 2014, pp. 1–8.

[155] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT". In: *Journal of Network and Computer Applications* (2019), p. 102481.

[156] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. "A Comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT". In: *Journal of Network and Computer Applications* 149 (2020), p. 102481.

[157] Sehrish Shafeeq, Masoom Alam, and Abid Khan. "Privacy aware decentralized access control system". In: *Future Generation Computer Systems* (2019).

[158] "ShoCard. Secure Enterprise Identity Authentication". In: *URL: https://shocard.com/* (2019).

[159] Sabrina Sicari et al. "Security, privacy and trust in Internet of Things: The road ahead". In: *Computer networks* 76 (2015), pp. 146–164.

[160] Jianping Song et al. "WirelessHART: Applying wireless technology in real-time industrial process control". In: *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE. 2008, pp. 377–386.

[161] John A Stankovic. "Research directions for the internet of things". In: *IEEE Internet of Things Journal* 1.1 (2014), pp. 3–9.

[162] Samuel Steffen et al. "zkay: Specifying and Enforcing Data Privacy in Smart Contracts". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2019, pp. 1759–1776.

[163] Jiliang Tang et al. "Exploiting homophily effect for trust prediction". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM. 2013, pp. 53–62.

[164] Lane Thames and Dirk Schaefer. "Software-defined cloud manufacturing for industry 4.0". In: *Procedia cirp* 52 (2016), pp. 12–17.

[165] Pascal Thubert et al. "Ietf 6tsch: Combining ipv6 connectivity with industrial performance". In: *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE. 2013, pp. 541–546.

[166] Andrew Tobin and Drummond Reed. "The inevitable rise of self-sovereign identity". In: *The Sovrin Foundation* 29.2016 (2016).

[167] Khalifa Toumi, César Andrés, and Ana Cavalli. "Trust-orbac: A trust access control model in multi-organization environments". In: *International Conference on Information Systems Security*. Springer. 2012, pp. 89–103.

[168] Khalifa Toumi, Ana Cavalli, and Mazen EL Maarabani. "Role based interoperability security policies in collaborative systems". In: *2012 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. 2012, pp. 471–477.

[169] Khalifa Toumi et al. "A vector based model approach for defining trust in multi-organization environments". In: *2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS)*. IEEE. 2012, pp. 1–8.

[170] Joydeep Tripathi, Jaudelice Cavalcante de Oliveira, and Jean-Philippe Vasseur. "A performance evaluation study of rpl: Routing protocol for low power and lossy networks". In: *2010 44th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2010, pp. 1–6.

[171] Anitha Varghese and Deepaknath Tandur. "Wireless requirements and challenges in Industry 4.0". In: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE. 2014, pp. 634–638.

[172] Steve Vinoski. "Advanced message queuing protocol". In: *IEEE Internet Computing* 6 (2006), pp. 87–89.

[173] Jiafu Wan et al. "Software-defined industrial internet of things in the context of industry 4.0". In: *IEEE Sensors Journal* 16.20 (2016), pp. 7373–7380.

[174] Licheng Wang et al. "Cryptographic primitives in blockchains". In: *Journal of Network and Computer Applications* 127 (2019), pp. 43–58.

[175] Qin Wang et al. "Preserving transaction privacy in bitcoin". In: *Future Generation Computer Systems* (2017).

[176] Shiyong Wang et al. "Implementing smart factory of industrie 4.0: an outlook". In: *International Journal of Distributed Sensor Networks* 12.1 (2016), p. 3159805.

[177] Yating Wang et al. "CATrust: Context-aware trust management for service-oriented ad hoc networks". In: *IEEE Transactions on Services Computing* 11.6 (2016), pp. 908–921.

[178] T Winter et al. *RPL: IPv6 Routing Protocol for Low Power and Lossy Networks. draft-ietf-roll-rpl-19*. 2011.

[179] Miao Wu et al. "Research on the architecture of Internet of Things". In: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. Vol. 5. IEEE. 2010, pp. V5–484.

[180] Qi Xia et al. "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments". In: *Information* 8.2 (2017), p. 44.

[181] Li Da Xu, Eric L Xu, and Ling Li. "Industry 4.0: state of the art and future trends". In: *International Journal of Production Research* 56.8 (2018), pp. 2941–2962.

[182] Xun Xu. "From cloud computing to cloud manufacturing". In: *Robotics and computer-integrated manufacturing* 28.1 (2012), pp. 75–86.

[183] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. "A survey on trust management for Internet of Things". In: *Journal of network and computer applications* 42 (2014), pp. 120–134.

[184] Xuanxia Yao, Zhi Chen, and Ye Tian. "A lightweight attribute-based encryption scheme for the Internet of Things". In: *Future Generation Computer Systems* 49 (2015), pp. 104–112.

[185] Shen Yin and Okyay Kaynak. "Big data for modern industry: challenges and trends [point of view]". In: *Proceedings of the IEEE* 103.2 (2015), pp. 143–146.

[186] Eric Yuan and Jin Tong. "Attributed based access control (ABAC) for web services". In: *IEEE International Conference on Web Services (ICWS'05)*. IEEE. 2005.

[187] Sheping Zhai et al. "Research on the Application of Cryptography on the Blockchain". In: *Journal of Physics: Conference Series*. Vol. 1168. 3. IOP Publishing. 2019, p. 032077.

[188] Fan Zhang et al. "Town crier: An authenticated data feed for smart contracts". In: *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*. 2016, pp. 270–282.

[189] Jie Zhang. "A survey on trust management for vanets". In: *2011 IEEE International Conference on Advanced Information Networking and Applications*. IEEE. 2011, pp. 105–112.

[190] Zibin Zheng et al. "Blockchain challenges and opportunities: A survey". In: *International Journal of Web and Grid Services* 14.4 (2018), pp. 352–375.

[191] Yan Zhu et al. "TBAC: transaction-based access control on blockchain for resource sharing with cryptographically decentralized authorization". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. IEEE. 2018, pp. 535–544.

[192] Lucia Ziyuan. "Web of Trust: ConsenSys Talks Ethereum Future, Presents uPort Blockchain Project". In: *URl: https://cointelegraph. com/news/web-of-trust-consensys-talks-ethereum-future-presentsuport-blockchain-project* ().