A Distributed Audit Trail for the Internet of Things

Inaugural dissertation
of the Faculty of Science,
University of Bern

presented by

Jakob Schärer

from Wädenswil ZH

Supervisor of the doctoral thesis:
Prof. Dr. Torsten Braun
Institute of Computer Science
University of Bern
Co-Referee:
Prof. Dr. Edmundo Monteiro
Department of Informatics Engineering
University of Coimbra

A Distributed Audit Trail for the Internet of Things

Inaugural dissertation
of the Faculty of Science,
University of Bern

presented by

Jakob Schärer

from Wädenswil ZH

Supervisor of the doctoral thesis:
Prof. Dr. Torsten Braun
Institute of Computer Science
University of Bern
Co-Referee:
Prof. Dr. Edmundo Monteiro
Department of Informatics Engineering
University of Coimbra

Accepted by the Faculty of Science.

Bern, 27.06.2023

The Dean
Prof. Dr. Marco Herwegh

Copyright Notice

Abstract

Sharing Internet of Things (IoT) data over open-data platforms and digital data marketplaces can reduce infrastructure investments, improve sustainability by reducing the required resources, and foster innovation. However, due to the inability to audit the authenticity, integrity, and quality of IoT data, third-party data consumers cannot assess the trustworthiness of received data. Therefore, it is challenging to use IoT data obtained from third parties for quality-relevant applications. To overcome this limitation, the IoT data must be auditable. Distributed Ledger Technology (DLT) is a promising approach for building auditable systems. However, the existing solutions do not integrate authenticity, integrity, data quality, and location into an all-encompassing auditable model and only focus on specific parts of auditability.

This thesis aims to provide a distributed audit trail that makes the IoT auditable and enables sharing of IoT data between multiple organizations for quality relevant applications. Therefore, we designed and evaluated the Veritaa framework. The Veritaa framework comprises the Graph of Trust (GoT) as distributed audit trail and a DLT to immutably store the transactions that build the GoT. The contributions of this thesis are summarized as follows. First, we designed and evaluated the GoT a DLT-based Distributed Public Key Infrastructure (DPKI) with a signature store. Second, we designed a Distributed Calibration Certificate Infrastructure (DCCI) based on the GoT, which makes quality-relevant maintenance information of IoT devices auditable. Third, we designed an Auditable Positioning System (APS) to make positions in the IoT auditable. Finally, we designed an Location Verification System (LVS) to verify location claims and prevent physical layer attacks against the APS. All these components are integrated into the GoT and build the distributed audit trail.

We implemented a real-world testbed to evaluate the proposed distributed audit trail. This testbed comprises several custom-built IoT devices connectable over Long Range Wide Area Network (LoRaWAN) or Long-Term Evolution Category M1 (LTE Cat M1), and a Bluetooth Low Energy (BLE)-based Angle of Arrival (AoA) positioning system. All these low-power devices can manage their identity and secure their data on the distributed audit trail using the IoT client of the Veritaa framework. The experiments suggest that a distributed audit trail is feasible and secure, and the low-power IoT devices are capable of performing the required cryptographic functions. Furthermore, the energy overhead introduced by making the IoT auditable is limited and reasonable for quality-relevant applications.

Contents

List of Figures **xiii**

Acronyms **xvii**

1 Introduction **1**
1.1 Motivation 2
1.2 Problem Statement 6
1.2.1 Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store 7
1.2.2 Identity and Signature Management of IoT devices . . . 7
1.2.3 Auditable Quality-Related Metadata 8
1.2.4 Auditable Positioning 9
1.3 Thesis Contributions 10
1.3.1 A Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store 11
1.3.2 Identity and Signature Management of IoT devices . . . 12
1.3.3 Distributed Calibration Certificate Infrastructure 13
1.3.4 Auditable Positioning System 14
1.3.5 SecureAoX 14
1.4 Thesis Outline 15

2 Background and Related Work **19**
2.1 Distributed Ledger Technology 19
2.2 Internet of Things 23
2.3 Public Key Infrastructures 24
2.3.1 General Public Key Infrastructures based on Distributed Ledger Technology 26
2.3.2 Public Key Infrastructures for the Internet of Things . . . 28

2.4 Distributed Calibration Certificate Infrastructure 30
2.5 Positioning based on Angle of Arrival 36
2.6 Proof of Location 38
2.7 Location Verification System 41
2.8 Conclusion 45

3 A Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store 47
3.1 Introduction 47
3.2 System Model 48
3.2.1 The Graph of Trust 49
3.2.2 Acyclic Block Confirmation Graph 58
3.2.3 Time Considerations 60
3.3 Implementation 64
3.4 Evaluation 65
3.4.1 Testbed 65
3.4.2 Stability 66
3.4.3 Throughput 68
3.4.4 Block Discovery Services 70
3.4.5 Security Analysis 71
3.5 Conclusion 80

4 Identity and Signature Management of IoT devices 81
4.1 Introduction 81
4.2 System Model 82
4.2.1 Identity of IoT Devices 82
4.2.2 Graph of Trust and the Internet of Things 83
4.2.3 Signing Measurements 85
4.2.4 Tamper protection 86
4.2.5 Associated Full Node 87
4.2.6 Statements 88
4.2.7 Time Considerations 89
4.3 Implementation 93
4.3.1 Testbed 93
4.3.2 Identity of Microcontrollers 95
4.3.3 Pairing 96

4.4 Evaluation . . . 98
4.4.1 Security Analysis . . . 98
4.4.2 Key Size . . . 99
4.4.3 Hardware Accelerated Cryptography . . . 100
4.4.4 Transmission Time to Distributed Ledger Technology . . . 101
4.4.5 Energy Consumption . . . 103
4.5 Conclusion . . . 106

5 Distributed Calibration Certificate Infrastructure 109
5.1 Introduction . . . 109
5.2 System Model . . . 110
5.2.1 Distributed Public Key Infrastructure for Metrology . . . 110
5.2.2 Digital Calibration Certificate Signatures . . . 112
5.2.3 Accreditation and Audits . . . 113
5.2.4 Metrological Traceability . . . 113
5.2.5 Multi-Signatures . . . 114
5.2.6 Measurement Values . . . 115
5.2.7 Digital Calibration Certificate Retraction . . . 116
5.2.8 Validation . . . 116
5.3 Implementation . . . 117
5.4 Evaluation . . . 119
5.4.1 Scalablilty . . . 119
5.4.2 Validation Performance . . . 120
5.5 Conclusion . . . 121

6 Auditable Positioning System 123
6.1 Introduction . . . 123
6.2 System Model . . . 124
6.2.1 Authentication and Proof of Location . . . 124
6.2.2 Secure Direction Finding . . . 127
6.2.3 Data Provenance . . . 128
6.2.4 Low Power Scan . . . 129
6.2.5 Sampling and Storage Requirements for Auditability . . . 130
6.3 Evaluation . . . 132
6.3.1 Testbed . . . 132
6.3.2 Energy Consumption . . . 133

6.3.3 Sampling and Storage Rate 135
6.3.4 Security Analysis 137
6.4 Conclusion 139

7 SecureAoX: A Location Veritifaction System for AoA Positioning 141
7.1 Introduction 141
7.2 System Model 142
7.2.1 Architecture 142
7.2.2 Independent Location Verification Information 143
7.2.3 Detecting Manipulated Positions 145
7.3 Evaluation 147
7.3.1 Experiment Setup 147
7.3.2 Performance Analysis 149
7.3.3 Security Analysis 152
7.4 Conclusion 154

8 Conclusions and Future Work 157
8.1 Summary of Contributions 157
8.2 Future Work 162

Bibliography 165

List of Figures

1.1 Auditable system 5
1.2 Overview of the Veritaa framework 10

2.1 Blockchain 19
2.2 BlockDAG 21
2.3 Relations between entities relevant for metrological traceability [68] ©2022 IEEE 31
2.4 Positioning using phase shift of a CW [85] © 2022 IFIP 37
2.5 Setup of an AoA-based positioning system 37
2.6 BLE Packet with a CTE 38
2.7 Difference between Proof of Location and Location Verification System 42

3.1 Elements of the GoT 49
3.2 Self-revocation of an identity claim 56
3.3 Revocation of a declaration 56
3.4 Document signatures on the GoT 57
3.5 Hash tree of a block of the ABCG [113] ©2020 IEEE 59
3.6 Fragmentation of the ABCG by discovery services [114] 62
3.7 Time assignment [114] 63
3.8 Implementation of a Veritaa Node [113] ©2020 IEEE 64
3.9 Veritaa testbed [113] ©2020 IEEE 66
3.10 Evaluation of churn [113] ©2020 IEEE 67
3.11 Average time to process a transaction [113] © 2020 IEEE 69
3.12 Throughput in relation to the block size [113] © 2020 IEEE . . . 70
3.13 Throughput comparison with other distributed ledgers [113] ©2020 IEEE 71
3.14 Block discovery delays [114] 72

3.15 An example GoT with a group of attackers [114] 74
3.16 Reputation Distribution [114] 75
3.17 Quality of reputation in relation to links and attacker cluster size [114] 77
3.18 Quality of reputation in relation to number erroneous trust links [114] 78

4.1 Real-world relations of an IoT device 84
4.2 Example GoT of a combined model for the PKI [136] ©2022 IFIP 84
4.3 Signing measurements and manipulated identity claim B′ 86
4.4 Hash tree extended with statements [136] © 2022 IFIP 88
4.5 Timeline for securing an event observation on a DLT 89
4.6 Secured time synchronization 90
4.7 Time synchronization declarations 92
4.8 Spatial resolution in sensor networks 92
4.9 IoT testbed and architecture overview [136] © 2022 IFIP 93
4.10 Identity claim using Silicon Labs SecureVault [136] ©2022 IFIP . 95
4.11 The sequence of pairing an IoT device with its AFN [136] ©2022 IFIP 96
4.12 Performance of different elliptic curves 100
4.13 Performance with and without hardware acceleration 101
4.14 Required time to send a statement to the DLT. [136] ©2022 IFIP . 102
4.15 LoRaWAN tx current consumption [136] © 2022 IFIP 104
4.16 LTE Cat-M1 tx current consumption [136] © 2022 IFIP 104
4.17 Required energy to transmit a message [136] © 2022 IFIP 105

5.1 Metrological traceability on the GoT [68] © 2022 IEEE 111
5.2 Digital Calibration Certificate Signature [68] © 2022 IEEE 112
5.3 Multi-Signatures [68] ©2022 IEEE 114
5.4 Measurement Signatures 115
5.5 Digital Calibration Certificate Retraction 116
5.6 DCCI architecture 117
5.7 Sidechains 118
5.8 Redundantly stored transactions [68] ©2022 IEEE 120
5.9 Validation times [68] ©2022 IEEE 121

6.1 Challenge packet [149] ©2023 IEEE . . . 125
6.2 Response packet [149] ©2023 IEEE . . . 125
6.3 Challenge update . . . 125
6.4 Asset tag position estimation . . . 126
6.5 GoT position trace . . . 128
6.6 Low power listening for advertisement updates [149] ©2023 IEEE 130
6.7 APS logical architecture [149] ©2023 IEEE . . . 132
6.8 Current analysis . . . 133
6.9 Energy consumption . . . 134
6.10 Average power consumption. Confidence intervals at level 99% for five experimental repetitions. . . . 134
6.11 PS data rate at different sampling rates . . . 135
6.12 Data rate of static and dynamic sampling rates for different mobility profiles . . . 137

7.1 Architecture of SecureAoX [85] ©2022 IFIP . . . 143
7.2 Process to update a challenge and measure independent location verification information [85] ©2022 IFIP . . . 144
7.3 Process to advertise AoA and location verification [85] ©2022 IFIP 145
7.4 Contradiction in AoA and Angle of Departure (AoD) angles [85] © 2022 IFIP . . . 145
7.5 Error ε between the AoA computed by the anchors and the AoD provided by the tag (or the attacker) to the verifier. [85] ©2022 IFIP . . . 146
7.6 Example of simulated scenario with one attacker and 10 position estimations from the anchors. [85] ©2022 IFIP . . . 148
7.7 Attacker to asset tag distance distribution [85] ©2022 IFIP . . . 149
7.8 Results of the Monte Carlo simulation [85] © 2022 IFIP . . . 149
7.9 ROC curve for different accuracies [85] ©2022 IFIP . . . 151
7.10 AoA Accuracy with LOS in relation to distance [85] ©2022 IFIP 152
7.11 Accuracy over distance [85] ©2022 IFIP . . . 153
7.12 Area under the ROC curve [85] ©2022 IFIP . . . 153

8.1 Example GoT with the most important elements . . . 161

Acronyms

ABCG	Acyclic Block Confirmation Graph
AC	Accreditation Certificate
ACK	Acknowledgments
AFN	Associated Full Node
AoA	Angle of Arrival
AoD	Angle of Departure
AoX	Angle of Arrival and Departure
APS	Auditable Positioning System
B-GPS	Blockchain-Based Global Positioning System
BIPM	Bureau International des Poids et Mesures
BLE	Bluetooth Low Energy
BFS	Breadth First Search
CA	Certificate Authority
CL	Calibration Laboratory
CRC	Cyclic Redundancy Check
CSS	Chirp Spread Spectrum
CTE	Constant Tone Extension
CW	Continuous Wave
D-LRT	Differential Likelihood Ratio Test
D-SI	Digital System of Units
DAG	Directed Acyclic Graph
DCC	Digital Calibration Certificate

DCCI	Distributed Calibration Certificate Infrastructure
DDoS	Distributed Denial of Service
DLT	Distributed Ledger Technology
DPKI	Distributed Public Key Infrastructure
EMC	European Metrology Cloud
EMDC	Enhanced Malicious Node Detection Clustering
EU	European Union
GFM	Greedy Filtering by Matrix
GFT	Greedy Filtering by Trustability-indicator
GoT	Graph of Trust
GPS	Global Positioning System
GW	Gateway
ILAC	International Laboratory Accreditation Cooperation
IoT	Internet of Things
IPS	Indoor Positioning System
IQ	In-phase and Quadrature component
ISM	Industrial, Scientific, and Medical
LoRaWAN	Long Range Wide Area Network
LTE	Long-Term Evolution
LTE Cat M1	Long-Term Evolution Category M1
LVS-DA	Location Verification System with Directional Antennas
LVS	Location Verification System
MFR	Manufacturer
MI	Measurement Instrument
MNDC	Malicious Node Detection Clustering
MNO	Mobile Network Operator
MQTT	Message Queuing Telemetry Transport

NACK	Negative Acknowledgments
NAS	National Accreditation Services
NB-IoT	Narrowband Internet of Things
NMI	National Metrology Institute
NN-LVS	Neural Network based Location Verification
NTP	Network Time Protocol
O	Owner
OCSP	Online Certificate Status Protocol
OPR	Operators
PDU	Protocol Data Unit
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PoL	Proof of Location
PoS	Proof of Stack
PoW	Proof of Work
PPCoin	Peer-to-Peer Coin
PS	Positioning System
PUF	Physical Unclonable Function
RPL	Routing Protocol for Low-Power and Lossy Networks
RSS	Receiver Signal Strength
SN	Sensor Node
SDN	Software Defined Networking
SI	International System of Units
TDAG	Transaction Directed Acyclic Graph
ToA	Time of Arrival
ToF	Time of Flight
TTN	The Things Network

VANET	Vehicle Ad-hoc Network
VPN	Virtual Private Network
WoT	Web of Trust
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Chapter 1

Introduction

The IoT is growing rapidly and becoming an integral part of our daily lives. This rapid growth is accompanied by new IoT applications that use the data of the IoT to provide new services. Many of these emerging IoT applications require similar IoT data to provide different services and could benefit from sharing the infrastructure. Sharing the infrastructure can lead to improved sustainability, cost savings, efficiency, and scalability. Additionally, shared infrastructure can foster innovation and collaboration between different stakeholders. However, third-party data processing carries the risk of the data being forged or manipulated. Furthermore, it could be that the IoT devices used for measurement are poorly maintained and, therefore, the data is of poor quality. Consequently, it is challenging to use data from third-party data producers. The data's authenticity, integrity, and quality must be auditable to overcome this limitation and to ensure the trustworthiness of data being shared between different stakeholders.

In this thesis, we study how DLT can be used to build a distributed audit trail. This audit trail enables consumers of data to verify the authenticity, integrity, and quality of received data, even when a third party owns the device producing the data. Mainly, we focus on IoT devices with very limited computational resources connected over low-power networks.

1.1 Motivation

Today, all kinds of electronic devices can be connected through the IoT. This connection enables applications without needing spatial proximity to the IoT devices. The applications that follow the IoT paradigm are manifold and range from sensors with low complexity, like temperature sensors, to systems of high complexity, like telesurgery systems [1]. The first known application following this paradigm dates back to the 1980s when D. Nichols connected a coke vending machine to the ARPANET, a precursor of today's Internet, to check if it is empty before taking the "long way" to the machine [2], [3]. Even with the first IoT applications available, it took until 1999 when Kevin Ashton first coined the term Internet of Things [4]. Since the first connection of a device to the IoT, much research and development have been done on the IoT.

In the early phase of the IoT the research was mainly focussed on connecting sensors over Wireless Sensor Networks (WSNs). These WSNs are typically sensor nodes that form an ad-hoc network. In these networks, the data packets are routed over the other sensors to a gateway using protocols like the Routing Protocol for Low-Power and Lossy Networks (RPL) [5], and the gateway then forwards the data packets to the Internet. More recent work suggested using Software Defined Networking (SDN) in low-power WSNs to optimize routing [6], [7]. For example, it was proposed that SDN can be used to route traffic around sensor nodes with low energy supply in order to extend their lifetime [8].

While for the WSNs, the operators mostly had to build their own infrastructure with gateways, today, networks exist that provide gateways to connect devices at almost every place on earth. Without the need to build and maintain network infrastructure, the IoT becomes a real use-case for many applications in the fields of industry, mobility, smart city, agriculture, and healthcare, to name a few. Consequently, the IoT is approaching its adoption phase, and enterprises are starting to use the IoT. With this adoption, the number of connected devices is expected to grow fast. In 2028 it is expected that 60% of the 4G/5G connections are dominated by IoT broadband connections [9]. The number of connected IoT devices is expected to grow from $11.3 \cdot 10^9$ devices in 2021 to

$29.42 \cdot 10^9$ in 2030 [10]. Since most of the IoT devices are measuring or producing data, a growing amount of data collected by the IoT can also be expected.

Even with trends towards open data and digital data marketplaces, IoT infrastructures are still primarily built with a single purpose and only accessible by their operators and owners. Open data are data anyone can freely access, use, modify, and share for any purpose [11]. Typically, open data are made accessible on organizations' websites or in public repositories that collect open data [12]–[14]. Since open data are typically made accessible at no charge, most open data datasets are from governmental organizations. For-profit organizations can sell and buy data over digital data marketplaces. With the growing value of data, these marketplaces are becoming more important. For example, a recently proposed data marketplace concept uses micropayments with cryptocurrencies to trade real-time data [15].

Sharing IoT infrastructure by making data accessible over open data or trading them on data marketplaces has several advantages. First, it could reduce the required resources, electronic waste, and carbon dioxide footprint. Next, the operators of IoT infrastructure could benefit from lower infrastructure costs if they share the infrastructure and the maintenance costs. Furthermore, besides the original purpose of a specific IoT infrastructure, the vast amount of data produced by the IoT holds the potential for new applications. Such applications could provide new services from aggregated data from multiple sources.

Today, the primary reasons that prevent organizations from exchanging IoT data are legal restrictions, confidentiality concerns, interoperability issues, difficult verification of data provenance, and difficult verification of data quality. While the first two reasons determine if an organization is allowed and willing to share certain data, the further are technical challenges.

Legal restrictions for sharing data exist, for example, for personal data. Personal data should only be collected where required, and personal data must be adequately secured. However, for non-personal data, the European Union (EU) wants to facilitate a free flow of data in Europe, allowing companies and public administrations to store and process non-personal data wherever they choose [16]. Nevertheless, profit oriented organizations might want to keep data private because it would reveal company secrets, or when they

could lose a competitive advantage when competitors have access to the same data. However, when the data are not a company secret, they could increase turnover by selling non-personal data collected by their infrastructure.

Interoperability between different technologies and protocols is challenging, especially when various stakeholders own the infrastructure. Many incompatible, often proprietary, protocols still exist that make data sharing difficult. However, with new standards and technologies, interoperability is improving. For example, digital data marketplaces can wrap data structures of various technologies and make them accessible to data consumers.

Nevertheless, despite improved interoperability, quality-relevant applications still do not use shared data. The major reason is that verifying the data's authenticity, integrity, and quality is challenging. For open data, the incentive for data forgery might be limited. However, when data is traded on digital data marketplaces, attackers might generate fake data to sell them. Therefore, consumers using IoT data of third parties must be able to verify the authenticity and integrity of received data to assess the data's trustworthiness. The accuracy of IoT devices depends on the device type. However, when sensors are used to measure physical parameters, the accuracy of these sensors might change over time. Therefore, maintenance like calibrations must be performed in adequate intervals. While this quality-related meta information is typically available for the owners of measurement instruments used for quality-relevant applications, it is typically not available for third-party data consumers. Without this information, third-party data consumers cannot assess the quality of the received data and, therefore, also cannot use this data in quality-relevant applications. Therefore, sharing quality-related metadata could increase the demand and value of the provided data. Since this quality information increases the value of the data, it must be provided in a verifiable form to prevent attackers from forging the quality metadata.

Consequently, verifying the authenticity, integrity, and quality of IoT data received from thrid-parties is a major challenge that prevents sharing IoT data between different infrastructure owners and operators. To overcome this issue, IoT data received from third parties must be auditable to enable the data receiver to verify the authenticity, integrity, and quality of data.

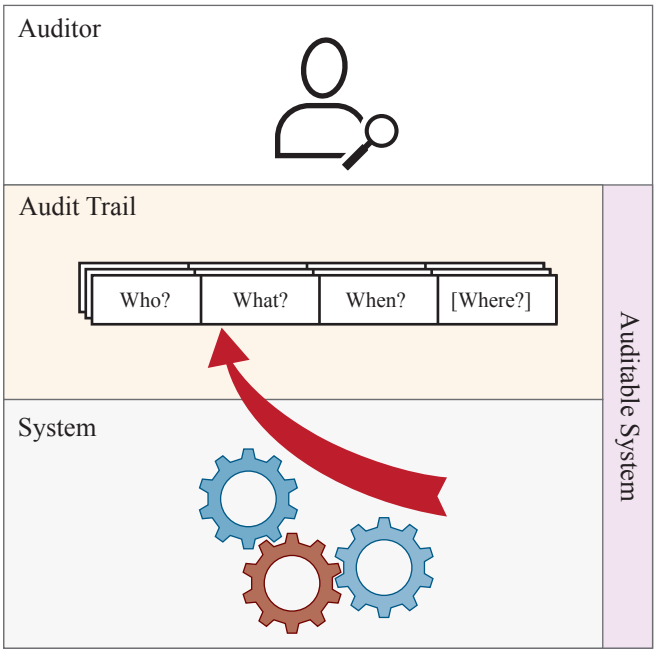


FIGURE 1.1: Auditable system

Auditability is always required when a third party must be able to verify, in this context, also called audit, that certain quality-relevant systems comply with certain standards, guidelines, or agreements. The concept of auditability is general and might originate from accounting, where third parties, like tax auditors, must audit financial reports. Auditability is understood to mean that it is possible to verify whether a system is functioning properly and, after that, that it has worked properly [17]. The audit trail or audit log is a sequential record that makes all integrity-related events in the auditable system visible [17]. Therefore, the audit trail tracks and documents the actions of entities within a system. It records who performed specific changes or observed certain events, what those changes or events were, and when they occurred. These audit records can be used to determine accountability and ensure compliance with relevant standards, guidelines, and agreements. Furthermore, with Global Positioning System (GPS) and Indoor Positioning Systems (IPSs) widely available, the audit trail of IoT systems should be able to keep track of where the changes or events happened. The position should always be auditable when the system's integrity depends on the position of an event recorded in the audit trail. For example, this could be a temperature that must

be measured at a certain position in a laboratory. Figure 1.1 shows an overview of a general auditable system. The auditor reads the audit trail to verify if the system complies with the relevant standards, guidelines, or agreements. In this thesis, the auditor is considered an oracle that performs the audit. Today, auditors are mostly human experts, but in general, also algorithms or artificial intelligence might be able to perform audits. However, this is not in the scope of this thesis. The audit trail must comprise a log of all integrity-related events for a system to be auditable. Therefore, the audit trail must be tightly coupled to the systems, and it must be immutable.

Distributed Ledger Technology is a promising approach for building the audit trail of auditable systems. In 2009 Bitcoin showed that blockchain technology could build an immutable transaction log, enabling cryptographic currencies [18]. While initially invented as an immutable log of monetary transactions, DLTs soon revealed their potential in other fields. DLTs like Ethereum [19], Hyperledger Fabric [20], and IOTA [21] can execute smart contracts enabling all kinds of applications that require an immutable audit trail.

In this thesis, we design and evaluate a distributed audit trail for the IoT. Therefore, we designed, implemented, and evaluated the Veritaa framework that integrates this audit trail. The Veritaa framework comprises the GoT as DPKI with a signature store and the Acyclic Block Confirmation Graph (ABCG) as application-specific BlockDAG. Furthermore, the system proposed in this thesis includes a DCCI, making data quality visible and auditable. Finally, we propose an APS and a LVS to make positions auditable. In this thesis, we primarily address low-power IoT devices and the challenge of making these constrained devices auditable. The challenges of building a distributed audit trail for constrained IoT devices and the contributions of this thesis are described in the following sections.

1.2 Problem Statement

This section discusses the research questions that arise when building a distributed audit trail. We first describe the problem and then formulate the research question for each issue we address in this thesis.

1.2.1 Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store

Each record of a distributed audit trail must be assignable to the entity that created it. Therefore, each entity able to create audit trail records requires a digital identity, and the entity must be able to sign the audit trail records with this identity. Asymmetric cryptography can be used to build a digital identity that is able to sign digital records securely. Data can be signed with the private key of an asymmetric key pair, and the public key can be used to verify this signature. However, to verify a signature, the verifier must authenticate that the public key belongs to the claimed identity. Public Key Infrastructures (PKIs) are used to delegate the authentication of public keys to a trusted third party, and Distributed Distributed Public Key Infrastructures (DPKIs) are used for the distributed authentication and certification of public keys.

Furthermore, to ensure the non-repudiability of audit log records, the distributed audit trail must provide an immutable signature store. DLT suits the requirements of an immutable signature store well and is also a promising approach to build a DPKI. However, previous designs of DLT-based PKIs do not focus on building an immutable audit trail and, therefore, lack a signature store. Additionally, most previous implementations depend on Blockchains with limited throughput and high energy consumption, which makes it not feasible to secure signatures of an audit trail. Therefore, an architecture optimized for DLT-based DPKI with a signature store should be designed and evaluated.

Research Question 1. *How to build a DLT-based, energy-efficient, and scalable DPKI with a signature store?*

1.2.2 Identity and Signature Management of IoT devices

DLTs provide an immutable transaction log by storing the transactions on multiple nodes and by forming a consensus about the valid transactions and their order. Therefore, the DLT nodes have to validate and store all transactions where they contribute to securing them. Most DLTs enable light nodes, sometimes referred to as Simplified Payment Verification, that only require downloading block headers and transactions relevant to them [18], [19]. However,

these DLTs still require full nodes that store, validate, and secure all transactions. In this work we define a full node as a DLT node that secures the DLT by maintaining a replica of all transactions. The set of nodes can be partitioned into multiple smaller groups of nodes called committees that operate in parallel on disjoint blocks of transactions and maintain disjoint ledgers to limit the number of transactions each node must process and store [22]. However, the nodes still require to validate and store all transactions of their committee. For many low-power IoT devices with limited bandwidth, storage, and computational capabilities, even handling a subset of all transactions is not feasible. Therefore, securing IoT data with DLTs is challenging, and the system architecture must consider the limitations of low-power IoT devices. Previous works mostly focus on securing data from low-cost, tiny computers like the Raspberry Pi 4. However, securing IoT data and managing the identities of low-power microcontrollers is not well covered in the related work and should be investigated further.

Research Question 2. *How to manage the identities and signatures of low-power IoT devices on a DLT-based DPKI with a signature store?*

1.2.3 Auditable Quality-Related Metadata

Digital signatures can be used to ensure the integrity and authenticity of data. DPKIs simplify the authentication process of large amounts of public keys by using certificates. These certificates certify that a certain public key belongs to its claimed identity. Accordingly, the digital signatures can assure that certain data were signed by a specific IoT device. However, thousands of IoT devices typically have the same device type and are exact copies except for the private key of a specific device. These devices of the same device type are generally exchangeable by any other device of the same device type. Furthermore, many device types, like various remote temperature sensors, fulfill the same purpose. While the scope of functions and the accuracy of the multiple types might vary, they still provide the same type of data. Consequently, even IoT devices of different types are exchangeable as long as their accuracy and scope of functions meet the applications' requirements. Independent of the quality of the device type and functional scope, IoT devices are systems interacting with the physical world. Therefore, the accuracy of an IoT device might

change over time, and eventually, the device breaks down. Consequently, the quality of IoT data depends on the device type and the maintenance of the device that produced it. Therefore, for third-party data consumers that procured data via a digital data marketplace or an open data platform, it is not enough to know that a certain device has signed the data and that a certain organization owns the device. More critical for third-party data consumers is to have auditable maintenance information, like metrological traceability, of the IoT device that produced the data. However, today the maintenance information is mostly only accessible by the IoT device owners and not accessible in an auditable format by third-party data consumers. Furthermore, this maintenance information is typically only linked to the IoT device and not to the measurement values. An audit trail for the IoT should enable metrological traceability for measurement values and provide this information to third-party data consumers.

Research Question 3. *How to securely manage quality-related metadata, like calibrations and other maintenance information, of IoT devices?*

1.2.4 Auditable Positioning

While the spatial position is irrelevant for many auditable systems, it can be crucial for IoT applications. For example, when an application must monitor that an asset never leaves a sterile area of surgery, then the position is integrity relevant for the application. Consequently, an audit trail for the IoT should also be able to make positions auditable. Proof of Location (PoL) and Location Verification System (LVS) focus on proving and verifying location claims. However, the subject of representing this information on DLTs is not well researched yet and requires further investigation.

Research Question 4. *How to make positions auditable with a distributed audit trail?*

The AoA positioning of BLE used in the testbed created in this thesis is vulnerable to a certain wormhole attack using reactive jammers on the physical layer. Since cryptographic functions cannot prevent this attack on the physical layer, an LVS is required to detect this attack and prevent adding manipulated positions to the audit trail. In LVSs, independently collected location information

is required to verify a location claim. BLE enables both AoA and AoD positioning. However, it has not been investigated if AoD can be used as independent location information to verify an AoA location claim (and vice-versa).

Research Question 5. *How to build an LVS for BLE-based AoA IPSs using AoD as independent verification information?*

1.3 Thesis Contributions

To investigate the various aspects of the research questions introduced in Section 1.2, we designed, implemented, and evaluated Veritaa, a framework that enables a distributed audit trail for the IoT. Figure 1.2 shows an overview of the Veritaa framework. Veritaa comprises the GoT, a graph-based model that represents the distributed audit trail, and a DLT to immutably store the transactions that form the GoT.

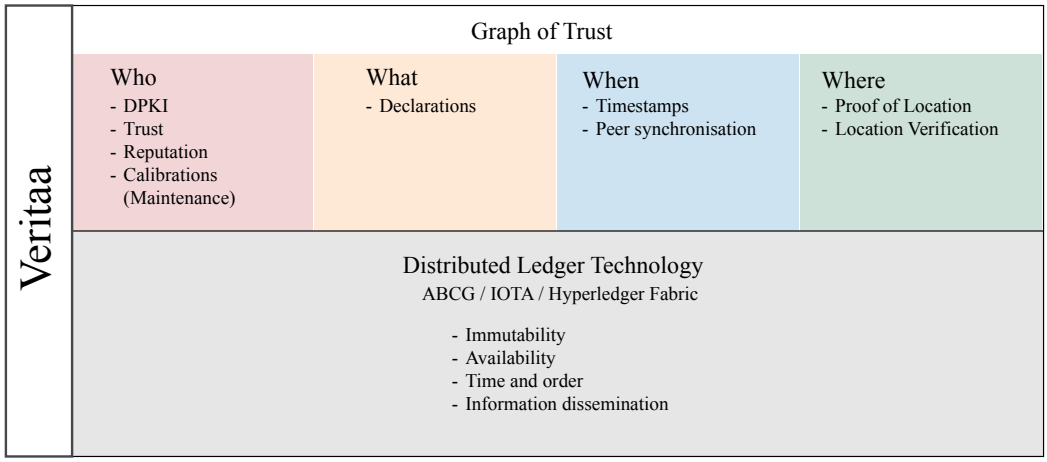


FIGURE 1.2: Overview of the Veritaa framework

The main contribution of this thesis is to design, implement, and evaluate the Veritaa framework. This main contribution can be divided into the following contributions:

1. Design, implement, and evaluate the GoT a DPKI based on the ABCG, an application specific DLT optimized to secure graph transactions
2. Optimization and evaluation of the aforementioned DPKI for low-power IoT applications

3. Design, implement, and evaluate a GoT-based DCCI to link quality-related meta information to the identities of IoT devices and their data
4. Design the data structure to represent auditable positions on the GoT, implement and evaluate a prototype for BLE-based AoA positioning
5. Design, simulate and evaluate SecureAoX, a location verification system to prevent wormhole attacks in BLE-based AoA and AoD IPSs

1.3.1 A Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store

All records of an audit trail require the signature of the entity that has either observed the documented event or performed the action that led to this event. Asymmetric encryption is used as the identity of entities. Identity claims map identifiers of entities to their corresponding public keys, and PKIs can be used to certify the authenticity of these identity claims. In order to build a PKI that meets the immutability requirement of audit trails and is able to secure signatures, we model a DLT-based DPKI with a signature store addressing research question 1.

We designed the architecture of Veritaa, a DPKI with a signature store. This architecture comprises the GoT, a model for the distributed certification of identity claims. On the GoT, entities can sign trust declarations towards authenticated identity claims of other entities. We use a reputation function based on signed trust declarations, domain vetting, and domain scoring for the distributed certification of identity claims. Domain vetting is the process of verifying that the same organization that owns an identity claim also owns a certain domain. With domain scoring, reputation can be attributed to well-known and trusted domains. Therefore, these domain-vetted organizations can act as trust anchors to bootstrap trust in the distributed certification model. To ensure the immutability of the GoT, we designed the ABCG, an application-specific BlockDAG optimized for storing graph transactions and throughput.

To evaluate the system's design, we have implemented a real-world testbed. The evaluation shows that the architecture for a DLT-base DPKI with a signatures store is feasible. Furthermore, reputation functions based on trust

votes and domain vetting are able to detect identity claims of malicious entities. Consequently, the proposed DPKI can be used to certify the authenticity of identity claims. Furthermore, the evaluation shows that the ABCG is fast, has a high throughput, and is resilient to churn. As an application-specific BlockDAG, the ABCG does not require the energy-demanding Proof of Work (PoW) and is, therefore, more energy efficient and scalable than other DLTs.

1.3.2 Identity and Signature Management of IoT devices

Since a full node of a DLT has to validate, store, and secure all transactions of the network or a subset of it, low-power IoT devices cannot handle the transactions of a full node. The light nodes enabled by some DLTs require to process all block headers and the blocks containing relevant transactions. Nevertheless, for low-power IoT applications, downloading and processing all block headers is often not feasible due to bandwidth and computational limitations. Therefore, the resource-demanding DLT tasks of the DLT-based identity management must be delegated to an Associated Full Node (AFN). The AFN is a full node that is able to process the DLT operations of associated low-power clients. To manage the identity claims and signatures of low-power IoT devices, we extended Veritaa, addressing research question 2.

We extended the architecture of Veritaa to support low-power IoT devices. To delegate resource-demanding DLT tasks to an AFN, we introduced statements in the ABCG. These statements enable low-power IoT clients to build a chain of their own transactions and forward them to their AFN, which includes the statements into blocks. Furthermore, to bootstrap the reputation of IoT devices, we introduced a new pairing declaration that enables the identity claims of IoT devices to inherit reputation from their paired identity claims.

We implemented the extensions in our real-world testbed to evaluate the extended architecture. Furthermore, we have built custom low-power IoT devices. The devices can connect to the DLT over LoRaWAN and LTE Cat-M1. The implementation shows that the proposed architecture can operate on low-power microcontrollers with limited computational capabilities and bandwidth. Therefore, the proposed solution is feasible for a wide range of

applications. In the experiments, we measured the time and energy required to sign and forward the signed statements required to secure audit records on the distributed audit trail. The comparison of forwarding secured and not secured data showed that, indeed, some overhead is introduced by securing the data, but it is limited and an acceptable tradeoff for increased security and quality. Therefore, we conclude that securing low-power IoT data on Veritaa is feasible.

1.3.3 Distributed Calibration Certificate Infrastructure

For third-party consumers of IoT data, auditable quality information is important to evaluate the value of quality-relevant received data. However, this metadata is often not available or not accessible to third-party consumers of IoT. Therefore, we design a model to represent calibration certificates and other quality-related metadata on the GoT addressing research question 3.

We designed a GoT-based model of a DCCI to link calibration information to IoT devices' identities managed with the GoT. With this new system, third-party data consumers can access and audit the quality-related metadata of a sensor. In contrast to the related work, our proposed solution is based on a graph model representing a DPKI and the calibration certificates simultaneously. Furthermore, our model enables the integration of accreditation certificates and National Accreditation Servicess (NASs) that certify the compliance of calibration laboratories with the relevant standards. This enables the validation of calibration laboratories' capabilities, and the well-known NAS act as trust anchors, improving the integrated DPKI. Furthermore, in contrast to previous work, our proposed solution comprises multi-signature schemes and approval processes.

We implemented the proposed DCCI into the prototype of Veritaa for evaluation. The evaluation shows that the proposed DCCI is feasible and scalable. Furthermore, the GoT enables declarations referencing GoT objects on different subgraphs. This enables loose coupling between the subgraphs of various sidechains and even between subgraphs of different DLTs. The ability of the GoT to be split into multiple fragments enables scalability and security.

1.3.4 Auditable Positioning System

Typically, audit records comprise a timestamp, data that describes the documented event, an identifier, and a signature of the record creator. However, for some auditable systems, the system's integrity also depends on the location of the IoT device that signed the event. We design an GoT-based model to make positioning auditable, addressing research question 4.

We designed APS to enable auditable positions in a distributed audit trail. This novel APS cannot only secure a PoL but also link the position to the raw data used to estimate these positions. This makes the positioning algorithm transparent for auditing entities and enables data provenance. Furthermore, auditors can audit and certify the location of anchor points required for the PoL, which enables auditable positioning in an industrial application without the need for global wireless infrastructure.

To evaluate the performance of the proposed APS, we extended the Veritaa testbed. We measured the additional energy required to transmit secured position claims, the impact of the position advertising frequency on system energy consumption, and the impact of the positioning sampling frequency on the required system storage data rate. Finally, we evaluate adaptive storage and sampling rates that optimize disk usage and energy consumption.

1.3.5 SecureAoX

The APS proposed in this thesis uses a cryptographic PoL to enable auditable positions. This PoL proves that a certain device was close to an anchor point. However, cryptographic functions cannot secure the physical layer of BLE-based AoA positioning. Therefore, this positioning is still vulnerable to a specific wormhole attack where the attacker jams and records a position packet and replays it at a different location.

An LVS that utilizes independently measured position information to verify a claimed position can be used to detect and protect the system from this kind of attack. Compared to other positioning systems, the AoA positioning bears the opportunity to measure the AoD as independent location information. This

has the advantage that no additional hardware is required. However, the related work does not cover using AoD as independent location information. Therefore, we propose a LVS for BLE-based positioning addressing research question 5.

We propose SecureAoX, an LVS for BLE-based AoA and AoD positioning systems. SecureAoX uses AoA for the positioning and AoD measurements as independent position information for the verification or vice versa. Since the AoA is measured on the anchor points and the AoD on the asset tag, AoA and AoD are independent measurements. When a position has not been manipulated, the distance between the AoA and AoD positions should be within the expected error of the IPS. Positions larger than the expected error are classified as attacks. The SecureAoX extends the APS introduced in Section 1.3.4 by an LVS for the detection of manipulated positions relative to the anchor points. Therefore, the extended APS is not only able to secure the proximity of an asset tag and anchor points but also its relative position. Consequently, this extension improves the accuracy of the APS.

We have implemented a simulator to evaluate the proposed LVS. We use Monte Carlo simulations to evaluate the performance of the proposed systems. The results show that the system is able to detect attacks that move the manipulated position further apart than the positioning system's accuracy from the original position. To cover different IPSs we simulate various accuracies. To relate the data to real-world results, we measured the accuracy of our BLE-based AoA IPS.

1.4 Thesis Outline

This section provides a brief overview of the thesis structure. The rest of the thesis is structured as follows.

Chapter 2 provides an overview of the related work and introduces the theoretical background required to understand the distributed audit trail proposed in this thesis.

Chapter 3 addresses research question 1 by introducing the Veritaa framework an DLT-based DPKI with a signature store. This chapter describes the fundamentals of the GoT and focuses on how it is used to build a DPKI. Furthermore, it presents the ABCG, shows how the transactions that form the GoT are securely stored on the ABCG, and how the blocks are timestamped. Finally, the evaluation shows the proposed system's stability, performance, and security.

Chapter 4 discusses the optimization of Veritaa for the IoT addressing research question 2. This chapter explains the peculiarities of IoT devices and how their identities can be managed on the GoT. Furthermore, the limitations of DLT timestamps are discussed, and how trust and synchronization declarations can be used to overcome these limitations. Finally, a real-world testbed with custom IoT hardware is presented and used to evaluate the system's performance. The evaluation focuses on the feasibility of the required cryptographic functions and the energy overhead introduced by securing the IoT data on the GoT.

Chapter 5 presents the architecture of a DCCI addressing research question 3. The DCCI uses the GoT to represent calibration and accreditation certificates. These certificates are quality-related metadata that enable metrological traceability and, therefore, auditability of IoT data. Furthermore, schemes to represent multi-signatures and approvals on the GoT, often required in metrology-related business processes, are introduced. Finally, this chapter explains how the GoT can be split into multiple sidechains and how this affects the scalability of the GoT.

Chapter 6 describes the architecture of an APS addressing research question 4. The APS uses a PoL to prove an asset tags position and secures this information on the GoT. The model to secure the positions on the GoT enables linking processed and raw data, making data provenance visible and auditable. Furthermore, this chapter defines the requirements that must be met to achieve audibility when the motion of an asset tag is tracked on a distributed audit trail. Finally, the proposed APS's energy consumption is evaluated.

Chapter 7 introduces SecureAoX, an LVS for BLE-based AoA positioning, addressing research question 5. SecureAoX extends the APS introduced in Chapter 6 with a LVS to detect manipulated positions and therefore increase the

accuracy of the APS. First, the model of SecureAoX is presented. Then, a simulator is built to evaluate the proposed system. The evaluation shows that SecureAoX can detect manipulations that move the manipulated position further apart from the attacked position than the positioning system's accuracy.

Finally, Chapter 8 concludes this thesis by summarizing the contributions and provides an overview of the future work that can be researched in the field of distributed audit trails to deal with the limitations given by IoT devices.

Chapter 2

Background and Related Work

2.1 Distributed Ledger Technology

A DLT is a distributed database that shares, replicates, and synchronizes transactions across multiple locations and between various entities. Typically, the DLT comprises nodes connected over a peer-to-peer network using gossip protocols to disseminate data. To achieve a replica of the database on all participating nodes, they must agree on what transactions are added to the DLT. In addition, for general-purpose or application-specific DLTs that require a total order of transactions, the DLT nodes also must form a consensus about the order of the transactions.

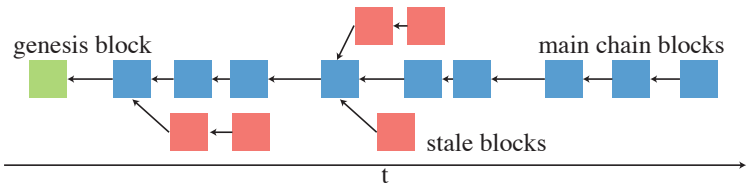


FIGURE 2.1: Blockchain

Blockchains are the most prominent form of DLTs. In blockchains, transactions are collected in blocks, and each block confirms its predecessor by adding the previous block's hash in its own header. Figure 2.1 shows an example of a blockchain. The first block of the blockchain is called the genesis block. The genesis block was created when the blockchain was set up. All other blocks reference the hash of the previous block in the blockchain. Referencing the previous block's hash ensures that the previous block cannot be altered anymore. Since newly created blocks require some time to disseminate through

the peer-to-peer network of the DLT, eventually, two nodes create valid blocks confirming the same block. These two valid blocks create a fork, and new blocks can be appended to both ends. However, the nodes always append blocks to the longest chain, so one chain will succeed. Valid blocks not confirmed by the main chain are called stale blocks.

Since each block can only have one successor, the network requires a consensus algorithm like PoW [18] or Proof of Stack (PoS) [23] to decide which node is allowed to create, also named as mine or mint, the next block. When PoW is used to determine which node can create the next block, the nodes have to prove that they have performed a certain amount of "work" to secure the network. Typically, the PoW is done by solving a computationally difficult mathematical task. This task's difficulty can be adjusted to ensure a certain block generation rate. The first node that is able to solve this mathematical task is allowed to generate the next block. The cryptocurrency Bitcoin is built on a blockchain with PoW as a consensus algorithm [18]. The success of Bitcoin revealed the drawbacks of PoW. The arms race between the miners led to an enormous energy consumption [24]–[27], and the throughput limited to seven transactions per second limits its scalability [28].

With PoS, validators can deposit cryptocurrency coins to secure the network. The deposited coins are called a stake and act as collateral. The more coins a validator has staked, the more likely it is able to create a new block and get coins as a reward. The collateral can be destroyed if a validator does not behave honestly. Therefore, the validators have an economic incentive to secure the network and behave honestly. With this consensus algorithm, no computationally difficult task must be solved, and therefore, less energy is required to secure the DLT. Several Blockchains have implemented different variations of PoS. Peer-to-Peer Coin (PPCoin) was an early cryptocurrency that implemented PoS [29]. Furthermore, Ethereum is a popular blockchain using PoS. Ethereum was initially using PoW as a consensus algorithm. However, due to scalability and efficiency issues of PoW, Ethereum migrated to PoS [30]. Ethereum is a blockchain popular for its generalized technology on which all transaction-based state machine concepts can be built [19]. In Ethereum, smart contracts written in a Turing complete programming language can be executed [19].

Securing the DLT with PoW or PoS is expensive. With PoW, the miners have to invest in hardware and energy, and with PoS, the validators must deposit capital in the form of cryptocurrency. Most blockchains award the creation of blocks with a certain amount of crypto coins to incentivize securing the DLT.

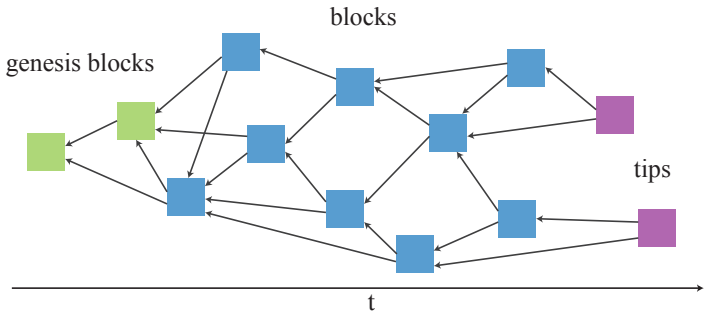


FIGURE 2.2: BlockDAG

BlockDAGs are DLTs where blocks confirm multiple other blocks [31] and therefore build a Directed Acyclic Graph (DAG) instead of a single blockchain. Figure 2.2 shows an example of a BlockDAG. A BlockDAG where each block references n previous blocks requires n genesis blocks, which cannot confirm n blocks since there are too few. Afterwards each block confirms n previous blocks. Tips are blocks that are not confirmed yet. In contrast to a blockchain, a BlockDAG has multiple tips. Therefore, all valid blocks with no reference are tips, and the BlockDAG does not have staled valid blocks. Another approach is the Transaction Directed Acyclic Graph (TDAG), where the DAG is built on transactions instead of blocks [31]. An advantage of DAG-based DLTs is that it is easier to append new blocks to a DAG than to the end of a blockchain. This makes DAG-based DLTs more suitable for nodes with constrained resources, enables higher throughput and improves energy efficiency. However, in contrast to blockchains, the DAG data structures do not inherently contain a total order of transactions. Therefore, in applications where the node state depends on the execution order of the transactions, the network requires to build consensus about the transaction's order. For example, the cryptocurrency IOTA is built on a TDAG, named Tangle [21]. Since the DAG data structure does not provide a total order of the transactions, IOTA initially used a centralized coordinator to resolve conflicting transactions. At the time of writing,

IOTA is migrating to a new decentralized consensus algorithm named coordicide [32]. Other DAG-based DLTs already provide a total order of transactions. For example, the hash graph of Hedera uses a "gossip about gossip" and virtual voting to bring the network to consensus on the timestamp of any event with the efficiency of bandwidth usage without centralizing around any entity or group of entities [33]. Hedera uses a gossip protocol to disseminate transactions through the DLT network. Afterwards, Hedera nodes that receive transactions from other nodes add a timestamp to the transaction hash and disseminate this information to the other nodes in the network. The Hedera nodes then use this gossip about when gossip was received to build consensus about the timestamp of a transaction. This protocol to build consensus about the timestamp is named "gossip about gossip".

DLTs can be classified into a few basic classes. On the one hand, DLTs can be public or private. In a public DLT, the read access to the transactions is open for everyone, and in a private DLT, the read access is limited to a few users, for example, the users of an organization. Different techniques have been proposed to enable private DLTs. For example, some proposed using networks with limited access, i.e., Virtual Private Networks (VPNs), to make DLTs private [34] and other DLTs like Hyperledger Fabric [35] use encrypted channels to limit the access to certain data of the DLT. On the other hand, DLTs can be permissionless or permissioned. In a permissionless DLT, everybody can set up a node and participate in the network. In a permissioned DLT, only nodes with permission can access the network and commit transactions. Hyperledger Fabric is a distributed operating system for permissioned blockchains [20]. To manage the permissions, Hyperledger Fabric uses a membership service provider that maintains the digital identities of all nodes, and all messages exchanged among the nodes are authenticated using digital signatures.

The existing literature on DLT encompasses many different DLTs. Most of these DLTs are either general-purpose DLTs that support smart contracts and all kinds of DLT applications. Due to the fast-growing cryptocurrency market, many DLTs specific for handling cryptocurrencies have been introduced. The general-purpose and cryptocurrency-specific DLTs must build consensus about the absolute order of transactions to prevent a double spending attack.

Building this consensus about the absolute order of transactions has the drawback that it requires energy, especially for PoW-based DLTs, high transaction costs, and limits the scalability. Applications that, by design, do not have transactions prone to double spending or execution could benefit from application-specific DLTs. For example, a distributed audit trail does not require an absolute order of transactions. Therefore, Veritaa can benefit from an application specific DLT.

2.2 Internet of Things

The IoT is the network that connects all kinds of electronic devices to the Internet. With many new applications in industry, smart home, smart city, autonomous driving, and more, the IoT grows quickly. These applications require different devices with distinct requirements. For example, some devices might be complex and connected to a powerful energy source, while others fulfill a simple purpose and run on a single coin cell battery for years. However, all IoT devices have in common that they eventually need to exchange information with other IoT devices, applications, or users. Today network technologies specialized for different IoT applications with various requirements exist. Especially low-power network technologies have undergone strong development. Today, technologies use the Chirp Spread Spectrum (CSS) to enable an efficient tradeoff between energy consumption, range, and datarate [36]–[38] for low-power communication. LoRaWAN uses CSS and a star-of-stars topology to connect IoT devices with gateways and servers [38]. Several LoRaWANs have been launched. Besides LoRaWANs operated by national Internet service providers, The Things Network (TTN) [39] and Helium [40] are global networks. In contrast to the other networks, Helium incentivizes the operation of gateways using a cryptocurrency that is mined by providing additional network coverage and forwarding data [40]. With this incentivization, Helium could reach much more coverage than TTN [41], [42].

Other technologies like Narrowband Internet of Things (NB-IoT) and LTE Cat M1 use cellular networks to connect devices to the IoT. These technologies are designed to provide low-cost, low-power, and reliable communication solutions for a wide range of IoT devices. NB-IoT is a narrowband IoT network built

from existing Long-Term Evolution (LTE) functionalities. The design goals of NB-IoT are to improve indoor coverage, enable low cost-devices, enable long battery life of low-power devices, and support a massive number of low-throughput devices per cell [43]. To achieve these design goals, NB-IoT uses a narrow band of $180kHz$, a reduced peak data rate of less than $100kbps$, and a latency budget of $10s$ [43]. While NB-IoT is designed to provide high coverage and enable ultra-low-cost devices LTE Cat M1 is intended for mid-range IoT applications and can support voice and video services [44]. Therefore, LTE Cat M1 utilizes a broader band of $1.08MHz$ and supports a data rate up to $1Mbps$ [45]. In addition, latencies bellow $10ms$ are reportedly feasible with LTE Cat M1 [46]. With these characteristics, LTE Cat M1 enables IoT applications that require a higher data rate, such as gateways connecting WSNs to the IoT.

In contrast to LoRaWAN, both NB-IoT and LTE Cat M1 have a higher data rate. While LTE Cat M1 and NB-IoT operate in the licensed frequency range, LoRaWAN operates in the unlicensed Industrial, Scientific, and Medical (ISM) band. Consequently, deploying new LoRaWAN gateways and increasing the network coverage to remote areas is easier with LoRaWAN. However, LTE Cat M1 and NB-IoT are deployed by the Mobile Network Operators (MNOs), which provide better ground coverage and enable easier roaming at the cost of higher fees.

Concluding, the choice of technology depends on the specific requirements of the IoT application, such as data rate, power consumption, operational costs, and coverage.

2.3 Public Key Infrastructures

In the wave of digitalization and with the fast growth of the IoT, digital documents and data have become essential and valuable resources. The value of data mainly results from their high availability, easy accessibility, and the possibility to move them effortlessly all over the world. Nevertheless, verifying their integrity and authenticity is difficult when data are moved over different channels. For quality-related auditable applications, it is of great importance that the integrity and authenticity of data can be verified. Hashes are used

to check the integrity of data and digital signatures to check their authenticity. Asymmetric encryption with private and public keys is used for digital signatures. Nevertheless, as long as public keys are not mapped to entities, their signatures are just unknown entities' signatures. An identity claim maps a public key to an identifier of an entity. However, as long as the identity claim is not authenticated, it is just a claim that everyone could have created. It would be possible to authenticate all public keys manually in a small world, but this is neither applicable nor scalable in most cases.

PKIs are required to authenticate large numbers of identity claims. Traditional PKIs can be distinguished by their certification model. PKIs use Certificate Authoritys (CAs) who sign certificates, and Distributed Public Key Infrastructures often rely on the decentralized Web of Trust (WoT).

The X.509 [47] standard is a popular PKI. X.509 certificates are defined as data structures that bind public-key values to subjects and trusted CAs assert these bindings [48]. The CAs are trusted in tree-shaped hierarchies. A root CA is asserting subordinate CAs by validating their identity. With over one thousand CAs all over the world [49] misfortune or bad intentions cannot be excluded, and the security of some CAs might be compromised. In the past, several incidents showed flaws in CA-based PKIs [50]. For example, hackers could get access to servers of the CA DigiNotar and use this CA's infrastructure to issue false certificates of high-profile domains like Google [51]. Incidents like this show the limitations of hierarchical systems regarding security and reliability.

Pretty Good Privacy (PGP) [52], together with the Web of Trust (WoT) [53], [54], is a popular DPKI. In the WoT, the public keys are certified by other entities [53], [54]. While the WoT provides transparency about the trust relations between entities, it lacks scalability and usability. Furthermore, public keys cannot be easily revoked with the WoT [55].

DLT-based PKI is a promising approach to overcome the drawbacks of conventional PKI. With a DLT, the management of the PKI can be distributed between multiple stakeholders, reducing the risk of a single point of failure. Furthermore, certificate revocations can directly be published on the DLT. The peer-to-peer network of the DLT ensures fast dissemination of such revocation;

therefore, no revocation lists are required. Managing these revocation lists in conventional PKI was often cumbersome and slow.

The literature covering PKIs based on DLT can mainly be grouped into general PKI and application-specific PKI. In this thesis, we propose a distributed audit trail that uses a PKI to manage the identities of both organizational entities, such as calibration laboratories and device owners, and IoT devices. To highlight the required optimizations for an IoT specific PKI, we discuss general PKI separately from the IoT optimizations.

2.3.1 General Public Key Infrastructures based on Distributed Ledger Technology

Several PKI based on DLTs have been proposed in the literature. The decentralization of the single point of failure in CA-based PKI and the certificate transparency is stated as the main advantage of DLT-based PKI. DLT-based PKIs can be categorized by how the certificate requests are certified. Like in conventional PKIs, the certification is mainly achieved through trusted CAs or the WoT. However, some architectures combine the CA and the WoT certification models to reduce the risk of a single point of failure at the trusted CAs and to bootstrap reputation in the WoT.

Kubilay et al. proposed CertLedger, a PKI architecture with certificate transparency based on a blockchain [56]. Their work aims to eliminate split-world attacks and provide certificate and revocation transparency. CertLedger is based on an Ethereum smart contract. CAs perform the tasks to check the identity of a domain owner and the certification of certificate requests. All certification and revocation actions are transparently stored on the blockchain. The CertLedger board observes all actions and can untrust malicious CAs. The authors show a complete architecture for CA-based PKI built on a smart contract and propose a governance model.

In [57], the authors design and implement a Blockchain-based DPKI on top of Hyperledger Fabric. To reduce the risk of a single point of failure, they propose a scheme where multiple trusted CAs certify certificate requests. For the distributed certification, they use multiple signatures of different network operators (CAs) to sign public keys and store them on the blockchain.

PBCert is a privacy-reserving blockchain-based PKI with a focus on scalability [58]. To reach scalability, PBCert is separated into a control and storage plane. In the control plane, all certificate operations are stored on a blockchain. Certification requests are sent to trusted CAs, and the certificates are then stored in an Ethereum blockchain. The certificate revocations are stored on Online Certificate Status Protocol (OCSP) servers' in the storage plane, and the OCSP servers' root hashes are secured on the blockchain of the control plane. In PBCert, obscure responses to clients' requests are used to preserve privacy. Therefore, the client only sends the first n bits of the certificate hash to an OCSP server, and the server responds with a BloomFilter of matching hashes.

A complete blockchain-based PKI system that supports the X.509 standard [47] has been implemented in an Ethereum smart contract [48]. All certificates, including those of the CA's, are stored on the blockchain. Therefore, the misbehavior of a CA is tracked on the blockchain and can be noticed by all participating entities. In the smart contracts of each CA, lists with all issued and revoked certificates are contained [55]. This improves the security of the tree-based PKI system.

Bitcoin has been used to enhance PGP and the related WoT. A new certificate format based on Bitcoin has been introduced that allows a user to verify PGP certificates using Bitcoin identity-verification transactions [59].

The Smart Contract-based PKI and Identity System (SCPKI) is an alternative PKI system based on a decentralized and transparent design using a WoT model and a smart contract on the Ethereum blockchain. In SCPKI, entities can verify fine-grained attributes (such as phone number, company, or domain name) of another entity's identity [60]. The verification of the attributes results in a WoT like data structure. However, this work focuses on asserting specific attributes rather than on deriving public-key certificates. Additionally, the authors mention the operational costs of running the smart contract on Ethereum as a challenge and publish a full and cheaper light implementation.

Several DLT based PKIs have been proposed in the literature. These PKIs mostly follow the traditional CA or WoT model and use the DLT as a distributed database to increase certificate transparency and enable fast revocation. Additionally, PKIs that follow the CA model often use smart contracts

to control and govern CAs. With all certificate actions transparently stored on the DLT, governing boards can observe the CAs' behavior and revoke their permissions. PKIs that follow the WoT model mostly use DLTs as distributed databases and for fast revocation. These revocation lists were difficult to realize in traditional WoTs. Most of the investigated projects rely on Ethereum smart contracts or existing general-purpose DLTs.

The literature study reveals some limitations that require further research for a DLT-based PKI used in an audit trail. First, most related works do not provide the immutable signature storage required to enable the non-repudiable signatures of an audit trail. Next, the WoT-based PKI in the related work does not provide a solution to bootstrap reputation in the system. For new trust-based systems, it is challenging to build reputation [61]. Consequently, it is difficult for new identity claims to gain reputation and certification. Finally, most related work depends on general-purpose DLTs with high energy consumption or transaction costs. This limits the scalability of the PKI.

2.3.2 Public Key Infrastructures for the Internet of Things

Some of the DLT-based PKIs proposed in the literature focus on the IoT. In Blockchain-based Public Key Infrastructure solutions for IoT, the authors implement and evaluate three different PKIs on Emercoin and Ethereum [62]. The first proposed PKI uses the Emercoin Name Value Store to store certificates on the blockchain. The other proposed solutions rely on Ethereum smart contracts. To meet the requirements of IoT applications, they evaluated two Ethereum architectures: one with a trusted remote node and one with an Ethereum Light Sync Node. However, the authors focus on storing self-signed certificates and are not providing a solution to authenticate and certify public keys. Therefore, it is difficult for third parties to verify the authenticity of the stored certificates.

In blockchain for IoT security and privacy, Dorrie et al. show how their blockchain can be used in Smart Home applications [63]. Their solution comprises local blockchains, a home miner, and local storage. The home miner generates shared keys for the devices and controls the access of the data. The

focus of their work lies in the confidentiality, integrity, and availability of sensor data. They do not provide a PKI solution that enables third parties to check the authenticity of data.

Bouras et al. proposed a lightweight blockchain-based IoT identity management approach [64]. In their work, they suggest using a consortium (permissioned) blockchain to manage the identities of IoT devices. Their solution separates the membership service and the identity management protocol. The membership service is used to manage the blockchain nodes and network admins. Their identity management comprises registration, verification, and revocation of certificates. They have implemented a proof-of-concept prototype using Hyperledger fabric.

Smart contracts of the Ethereum blockchain have been used to build an Identity Management System for IoT devices [65]. The architecture comprises globally unique identifiers, identity management throughout the lifecycle, and ownership management of IoT devices. The proposed system relies on public-key cryptography. The manufacturer generates the key pair at production, requests a certificate from a CA, and then stores it on the created IoT device. Afterwards, the manufacturer registers the device identity on the blockchain. Finally, the manufacturer uses ownership management to transfer the device ownership to the customer when the device is sold.

Multiple DLT-based PKIs for the IoT have been proposed in the literature. In most of the related work, the public keys are certified by an entity, for example, the manufacturer of the device. However, to evaluate the quality of data measured by IoT, devices owned and operated by third parties, the validator requires more information than the authenticity of a public key. For example, the maintenance, and with that, the operator of an IoT device is essential for its well functioning. Therefore, multiple parties must be able to declare their relation to IoT devices to establish trust in IoT devices beyond their infrastructures. The DPKI requires an immutable signature store to enable non-repudiable signatures. Furthermore, most related work does not cover the identity of the IoT device itself.

2.4 Distributed Calibration Certificate Infrastructure

Humans have always observed and measured their environment. With the development of architecture and trade, comparable measurements have become necessary. To make measurements comparable, people must measure with the same units. The cubit is perhaps the oldest and longest-lived example of a standard measurement unit, used to measure the length in Egypt at around 2750 BC [66]. The accuracy was already taken seriously back then, and the cubit had to be calibrated at each full moon [66]. All measurements must be compared with the same standard, and this standard must be measurable with the required precision to ensure accuracy. Since 2018, all base units of the International System of Units (SI) have been defined by natural constants [67].

Experiments to derive a base unit from its corresponding natural constant are complex and expensive. Therefore, National Metrology Institutes (NMIs) perform these experiments and maintain the national standards. In addition, the NMIs calibrate the standards of Calibration Laboratorys (CLs), and the CLs calibrate the Measurement Instruments (MIs) used in the industry. An MI is a device to measure a physical parameter. The calibration of an MI with intermediate standards results in a chain of calibrations where each step in the calibration chain contributes to the measurement uncertainty. Following the chain of calibrations in reverse order enables metrological traceability. Considering the accuracy of a MI and the uncertainty introduced by the chain of calibrations, measurement values of two calibrated MIs can be compared. To assert worldwide comparability of measurements, the Bureau International des Poids et Mesures (BIPM) works with the NMIs of the BIPM's Member States and strategic partners worldwide.

Figure 2.3 shows the relations between the BIPM, the NMIs, CLs, Manufacturers (MFRs), and MIs. The calibration hierarchy follows the relations from the NMIs down to the MI. With each calibration, the uncertainty of the measurement is increased, and consequently, the MIs higher in the hierarchy have higher accuracy.

The CLs require accreditation for the international acceptance of calibration

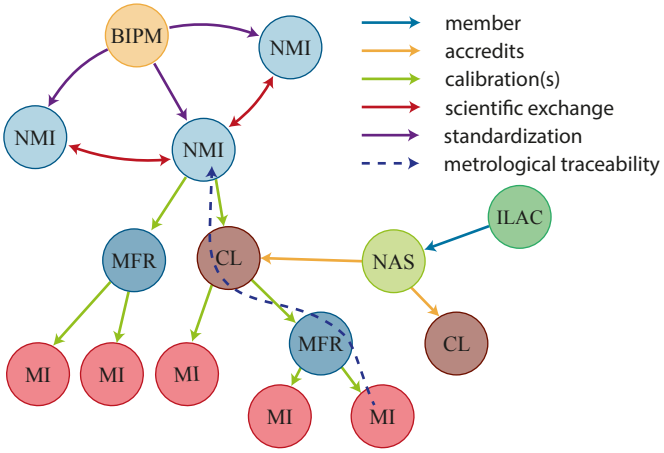


FIGURE 2.3: Relations between entities relevant for metrological traceability [68] © 2022 IEEE

and test certificates. The accreditation asserts that the laboratories not only have accurate measurement equipment but also are in compliance with relevant standards (like the ISO/IEC 17025 for calibration) to ensure quality. NAS, united by International Laboratory Accreditation Cooperation (ILAC), accredit the CLs.

To ensure the quality of the measurement results, MIs should be calibrated when they are manufactured and periodically afterwards. The periodicity depends (among others) on the risk of failure, environmental conditions, and frequency of use. Since the calibration is only a snapshot of the MI accuracy, measurements after the last calibration are potentially wrong when a failure is detected. For example, a failure is detected in a sensor monitoring the temperature of a temperature-sensitive chemical in a production line. Then it might not be clear when the failure occurred, and the last calibration is the point where it was checked that the sensor has been measuring properly. In the worst case, since the last calibration, the whole production is damaged and must be discarded. In this case, frequent calibration reduces the damage in case of failure.

With the growth of the IoT, the digitization of metrological traceability gains attraction. The literature reveals several related works that cover digitizing metrology. Gadelrab et al. conducted a comprehensive analysis and survey about the requirements and the state of research towards the new generation

of Digital Calibration Certificates (DCCs) [69]. Different units and unit conversion are major challenges when metrology data is exchanged between systems. A Digital System of Units (D-SI) metadata model has been proposed to overcome these problems [70]. The D-SI has a machine-readable format for the exchange of metrological data. It is an Extensible Markup Language (XML)-based format that comprises the measurement value and SI-Unit information. Calibrations are used to trace metrological data back to the national standards maintained by the NMIs, and calibration certificates document these calibrations. Most of the calibration certificates are still paper-based or in a non-machine-readable format. Consequently, the accuracy information provided by the calibration certificates is not machine-readable, and it is not easy to process them further. To overcome the issue of non-machine-readable calibration certificates, the DCC has been proposed [71], [72]. The DCC is a machine-readable, XML-based calibration certificate. The DCC contains all the mandatory fields of a paper-based certificate and can be signed with cryptographic signatures. In addition, the DCC comprises administrative data, measurement results, comments, and a document annex area. Listing 2.1 shows an excerpt of a DCC with some important information. For more details, the authors of the DCC provide several examples [73] and an XSD schema [72].

```
<dcc:digitalCalibrationCertificate
  xsi:schemaLocation="https://ptb.de/dcc/v3.1.1/dcc.xsd"
  schemaVersion="3.1.1">
  <dcc:administrativeData>
    ...
    <dcc:calibrationLaboratory>
      <dcc:contact>
        <dcc:name>
          <dcc:content>
            Calibration Laboratory Ltd
          </dcc:content>
        </dcc:name>
        ...
      </dcc:contact>
    </dcc:calibrationLaboratory>
    ...
  </dcc:administrativeData>
  <dcc:measurementResults>
    <dcc:measurementResult>
```

```
      <dcc:name>
        <dcc:content>Measurement Results</dcc:content>
      </dcc:name>
      ...
      <dcc:results>
        ...
            <si:realListXMLList>
              <si:valueXMLList>
                298.15 303.61 340.32
              </si:valueXMLList>
              <si:unitXMLList>\kelvin</si:unitXMLList>
            </si:realListXMLList>
        ...
      </dcc:results>
    </dcc:measurementResult>
  </dcc:measurementResults>
</dcc:digitalCalibrationCertificate>
```

LISTING 2.1: Excerpt with some important data of a DCC

The manual authentication of all relevant public keys is not feasible with many different stakeholders in the field of metrology. Therefore, a PKI is required to validate the DCC signatures. The European Metrology Cloud (EMC) is an initiative to direct and uniform the digitization of European legal metrology. It was proposed that the EMC should use a consortium (permissioned) blockchain to secure the DCC data [74]. A demonstrator showed the first version of the EMC. In this demonstrator, a tree-based CA architecture was used to authenticate the public keys of calibration laboratories, and the demonstrator used a permissioned DLT protected with a VPN [34].

Several blockchain-based approaches to secure metrological traceability have been proposed recently. In blockchain applications for legal metrology, Peters et al. discuss if DLTs have the potential to disrupt legal metrology [75]. On the example of smart meters, they explain the challenges in legal metrology and conclude that DLTs are promising to solve several of these challenges. According to their research, the main advantages of DLTs are decentralized audit trails, software verification, and an update process based on smart contracts.

Takatsuji et al. proposed to use blockchain technology to visualize the metrological traceability [76]. To visualize metrological traceability, they store certificate metadata, and the certificate hashes in transactions of a blockchain. Additionally, they store the transaction ID of parent certificates symmetrically encrypted in the certificate transaction. These parent certificates enable upstream certificate information only accessible to persons having the symmetric key. While this enables recursive access to all certificates higher in the certificate hierarchy, it also has the drawback that the symmetric key must be shared among all that should have access to the certificates. Sharing symmetric keys bears the several risks since it can get lost, and it is not possible to revoke access for entities when the access was granted.

Mustapää et al. propose a comprehensive conceptual solution based on DCCs, D-SI, and cryptographic digital identifiers for validation of data quality and trustworthiness [77]. Their work highlights that building a global PKI is one of the major open challenges.

Other related work goes further and proposes to secure more IoT functionality with DLTs. Shah et al. proposed to use Ethereum smart contracts for Secure Calibration in Safety-Critical IoT [78]. Their work highlights the importance of traceability for safety resilience in surgery robots. They propose to store the certificate chain immutably on the Ethereum blockchain. In this solution, the NMIs are the root CA and certify the measurement instruments of intermediate calibration laboratories. In addition, the authors suggest that IoT sensors can communicate directly and verify on the DLT if they have a valid DCC. The measurement equipment would be deactivated if a certificate was revoked on the DLT.

Melo et al. identify the huge amount of data, measurement of privacy-relevant data, and authentication of oracles as the main challenges in digitizing legal metrology [79]. In using blockchains to implement distributed measuring systems, Melo et al. proposed to run all legally relevant source codes of legal metrology as a smart contract on blockchains [80]. A measurement instrument comprises only the sensor, an analog-to-digital converter, and a communication module in their vision. The authors propose to send raw sensor data directly to smart contracts that further process the data. An advantage of this

model is that notified bodies can audit the legally relevant source code. As a proof of concept, they implemented a speedometer on Hyperledger fabric. They conclude that blockchain-based DMS is a promising approach, but large data measurements, privacy, and the authentication of oracles (sensors owned by other entities) are still open challenges. Blockchain-based PKI for smart meters has been proposed to provide trust in fuel dispensers [81]. In the proposed architecture, the CA is replaced by a blockchain where permissioned endorsers can directly commit certificates. The permissioned endorsers are manufacturers, entities from society, and institutions responsible for assuring the correct behavior of smart meters.

A prototype for dual digital traceability of metrology data using X.509 and IOTA has been built [82]. The signed fingerprints of DCCs in the prototype are immutably stored on the IOTA ledger. A CA-based PKI is used to manage and certify the public keys. The authors propose to use the NMIs as root certificate authorities.

Work	**DCC Signatures**	**Metrological Traceability**	**Certificate Retractability**	**Audit Certificates**	**Multi-Signature**	**Secure Measurements**	**PKI**
DCCI	✔	✔	✔	✔	✔	✔	DPKI with domain vetting of NMIs and NASs as trust anchors
[34], [74]	✔	✔					CA based PKI
[76]	✔	✔					
[77]	✔	✔				✔	PKI as open challenge see potential to use NMIs and NAS as trust anchors
[81]	✔	✔	✔			✔	Blockchain based with endorsers
[82]	✔	✔	✔				CA based PKI with NMI as root CA

TABLE 2.1: Related Work [68] © 2022 IEEE

The related work has proposed several approaches to digitize and secure

metrological traceability. While some of the related work is application-specific, others propose general calibration certificate infrastructures. Table 2.2 provides a brief overview of the general solutions proposed in the related work. This overview shows that most related work is able to store DCC signatures and represent metrological traceability. However, most of the related work falls short in providing an applicable PKI and some even state that building a PKI for metrology is one of the major open research challenges. Furthermore, most of the related work falls short of supporting multi-signatures and approval processes often required in metrology. Finally, only a few of the solutions proposed in the related work are able to secure measurement values of the calibrated MIs and, therefore, enable metrological traceability for measurement values. A drawback of all solutions investigated in this literature study is that no solutions can cover all requirements with their system model.

2.5 Positioning based on Angle of Arrival

With the growing importance of the IoT and especially the industrial IoT, IPS have become widely used for asset tracking. Among others, indoor positioning finds applications in robotics, medicine, logistics, tracking, localize-and-fetch in warehouses, and the navigation of unmanned ground vehicles [83], [84]. A typical radio-based IPS comprises anchor points that estimate the relative distances or directions of the tracked asset tags and a positioner that uses mutlilateration or multiangulation to estimate a position based on these relative distances or direction estimates. For the distance and direction estimation, the anchor points measure physical properties like free-space path loss, Time of Arrival (ToA), Time of Flight (ToF), or the phase shift between the antennas of an array. In this thesis, we focus on direction finding based on phase shift. Direction finding based on phase shift depends on the physical property that electromagnetic waves propagate at the speed of light. Therefore, two antennas observe a phase shift depending on their distance when they measure a signal. This phase shift is used in AoA and AoD to estimate the direction of an asset tag. For simplicity, AoX addresses both AoA and AoD in this thesis.

Figure 2.4a illustrates AoA. To measure the angle θ_{A}, the sender T_0 transmits a Continuous Wave (CW), and the antenna array $T_1, \ldots, T_K$ measures the signal.

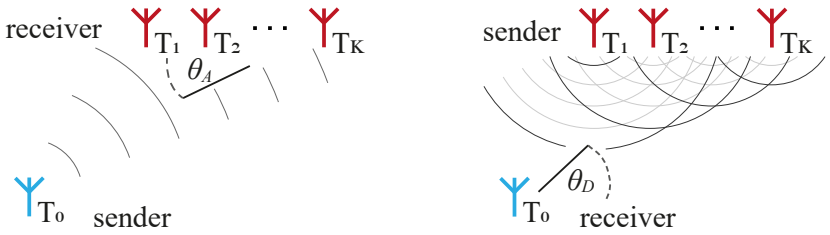


FIGURE 2.4: Positioning using phase shift of a CW [85]

The phase shift between the antennas $T_1, \ldots, T_K$ defines angle θ_A.

Figure 2.4b illustrates AoD. In contrast to AoA, with AoD, the CW is switched through the antennas of the antenna array during transmission $T_1, ..., T_K$. The switching of the CW through the antenna array results in a phase shift between the switching slots. This phase shift is measured on the receiver side and can be used to determine the direction of the signal.

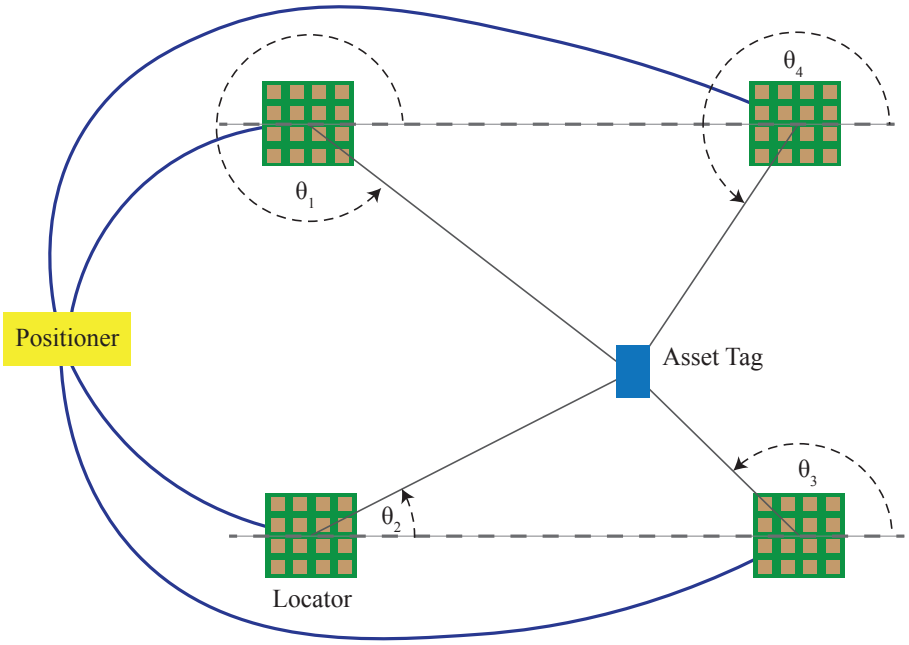


FIGURE 2.5: Setup of an AoA-based positioning system

Multiple locators at known positions and with a known orientation can act as anchor points that measure the AoA of a CW sent from an asset tag. A positioner can use the AoAs of the locators to estimate the asset tag's position

using multiangulation. Figure 2.5 shows an example setup of a positioner that uses the AoAs measured by locators to estimate the position of an asset tag.

FIGURE 2.6: BLE Packet with a CTE

The standardization of AoA and AoD in Bluetooth 5.1 [86] made positioning based on phase shift available in a wide range of inexpensive off-the-shelf low-power IoT devices. Constant Tone Extension (CTE) was introduced to enable positioning based on phase shift in BLE. The CTE is a CW appended after the Cyclic Redundancy Check (CRC) of BLE packets. Figure 2.6 shows a BLE packet with enabled CTE. Typically, the locators have an antenna array, and the asset tags have a simple BLE module with one antenna. This enables low-power, cheap, and small asset tags.

2.6 Proof of Location

A Positioning System (PS) is a system to measure the spatial position of objects. In addition to the IPSs, the PS also comprise large-scale systems like the GPS. While PSs are becoming increasingly popular, it is still challenging for independent organizations to verify whether the positions claimed by sensors are genuine or have been manipulated. Suppose users can claim that they are located at an arbitrary position, and the location-based application is unable to verify the genuineness of the position. In that case, several issues may arise, ranging from mild offenses such as cheating in augmented-reality online games [87] to more serious ones, such as circumventing international sanctions [88] or influencing the navigation system of an autonomous vehicle [89].

For quality-relevant applications in medical and industrial applications, positions and sensor data must be auditable to enable auditors to verify a system's state and, therefore, assert product or service quality. For example, in a geofencing medical application, an insurer (i.e., a third party) must be able to verify that surgical instruments have never left the sterile area during surgery by verifying their claimed location. A PoL verifiable by an independent third party is required to make a position auditable. A PoL is cryptographic proof

that a tracked asset tag was present at the claimed location. Therefore, anchor points in proximity of the claimed location verify if the monitored asset tag is at this location. For example, the anchor points can use the property that a radio signal has a limited range to verify if the monitored asset tag is in their vicinity.

The first research on verifiable positions was already made in 2001 when Zugenmaier et al. proposed to enhance applications with approved location stamps [90]. Later in 2009, Saroiu and Wolman pioneered the concept of location proofs [91], a mechanism to enable users to certify their location history and prevent malicious users from claiming forged locations. To provide a PoL, the device, which position claim should be proved, must be authenticated. The device's authentication can either be based on cryptographic identities or radio fingerprints of the monitored device, like unique imperfections of the analog front-end [92]. The location proofs are provided by wireless infrastructure that authenticates the tracked device and certifies the physical closeness of the tracked device. Therefore, a trusted wireless infrastructure was required to perform the PoL. Recent research in PoL systems moves towards replacing the trusted wireless infrastructure with decentralized and DLT-based systems. Amoretti et al. proposed a Blockchain-based PoL system that guarantees both location trustworthiness and user privacy preservation [93].

Large-scale wireless infrastructure is required to provide PoL at all locations where auditable positioning could be used. DLT is a promising approach to decentralize the construction of such infrastructure, remove the requirement to trust a single entity, and enable an immutable audit log. Since the auditing entity needs to verify the sensor's historical data, the system must store its claims on a public immutable audit trail.

The research around location proofs has evolved in the past two decades, and several research projects securing PoLs have been proposed. Zafar et al. provide a comprehensive overview of the research in the field location proofs [94].

A PoL is essential to make positioning auditable. The PoL is often done by exchanging cryptographically signed data between a tracked asset tag and anchor points of a trusted infrastructure over a radio connection with a limited

range [91], [95]–[97]. The signed data proves that the asset tag is close to the anchor point that has signed the data. Other works use the trusted infrastructure of roadside units in Vehicle Ad-hoc Networks (VANETs) for periodic location proofs, combined with plausibility checking of movement between proofs [98]. A major challenge of these infrastructures is that a trusted entity must set up a large-scale infrastructure with anchor points at all locations where location proofs are used. However, building such infrastructure is associated with large investments, which entail risks for the operators and slow down the adaptation process.

In contrast to the infrastructure-based solutions, collaborative approaches to establish trust in positions can be found in the field of VANETs [99], and mobile networks [93], [100], [101]. With these collaborative methods, other users act as witnesses for the location of the node whose position must be proven.

Some PoL systems forward the location proofs to the tracked device, which then can use this location certificate to prove to a third party that it was at a given time at a claimed position. But most PoL systems do not securely store these location proofs [91], [96], [97], [99]. However, for a PS to be auditable, an auditor must be able to determine the position of an asset tag in history. Therefore, the PoL must be stored immutably, non-repudiable, and accessible for the auditor in an audit trail.

Some of the related work uses a trusted centralized server to store the location proofs and to provide them to the verifiers [95], [101]. The drawback of central servers is that in an auditability scenario, the auditors and the audited entities have to trust the operator of the server not to remove location proofs from the server. By using DLTs, the entities no longer have to trust a third party. Several papers proposed to secure location proofs with DLTs [93], [102]. The location proofs are usually secured by signing the hash of the location proof and saving it immutably on a DLT.

In order to overcome the challenge of building a global anchor point infrastructure, some projects incentivize the operation of anchor points by rewarding the operators with cryptocurrencies when they provide network coverage or perform a PoL [40], [103]. However, the incentivized infrastructures need to

prevent the simulation of location proofs or proof of coverage. The current approach to solve these challenges is that only certified manufacturers can make anchor points.

In the literature, the PoL often proves that the tracked asset tag is within the range of a certain anchor point [91], [93], [97], [99], [101]. While this is suitable for applications that do not require high accuracy, it is a limitation for applications depending on precise positions, like indoor positioning in industrial applications.

Other related work depends on GPS. Blockchain-Based Global Positioning System (B-GPS) was designed to improve the integrity and transparency of geodetic surveys [102]. Using GPS for positioning applications has the advantage of being globally available outdoors, but it also comes with the drawback that it cannot be used for indoor applications.

In summary, the literature review reveals the following drawbacks of the existing solutions. PoL systems proposed in the literature either require large-scale wireless infrastructure or use a decentralized collaborative approach, where asset tags witness and prove each other's locations. The large-scale wireless infrastructure has the drawback that it requires large investments and is not suitable for industrial indoor applications. The fully decentralized location proofs have the drawback of requiring asset tags of multiple independent entities at the same location to build trustworthy location proofs, which is often not the case on company premises. For industrial applications, permanently installed and auditable anchor points could overcome these drawbacks. Finally, non of the solutions found in the literature covered BLE-based AoA positioning.

2.7 Location Verification System

The PoL presented in Section 2.6 can cryptographically prove that an asset tag is within the communication range of the locators that performed the PoL by using cryptographic functions. However, physical parameters must be measured to estimate the distance or angle between locators and asset tags. These physical parameters can not be secured by cryptographic means and are, therefore, vulnerable to attacks on the physical layer. For example, an attacker could

replay a packet used for positioning at a different location without manipulating the cryptography and, therefore, manipulating a position. A LVS uses independently measured location information to verify the feasibility of claimed locations and mitigate such attacks.

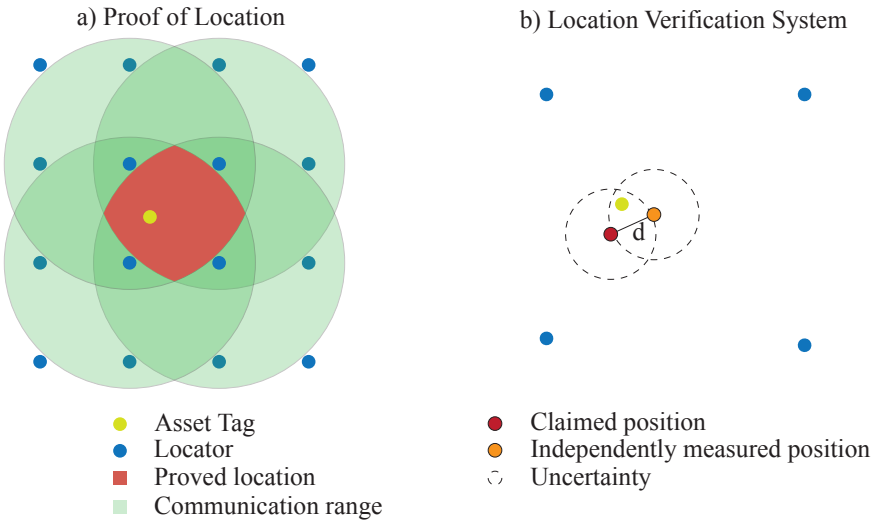


FIGURE 2.7: Difference between Proof of Location and Location Verification System

PoL and LVS complement each other in building accurate auditable positioning. Figure 2.7 shows the difference between a PoL and a LVS. The PoL cryptographically proves that the asset tag is within the intersections of the communication range of the locators that performed the PoL. In contrast to the PoL, the LVS verifies that the distance d between the claimed position and the independent verification information (in this case, the independent verification position) is smaller than the uncertainty of the positioning system. If d is larger than expected, the LVS classifies the claimed position as manipulation. Note that the asset tag is within the uncertainties of the claimed and the verification position due to measurement uncertainties. In summary, the LVS is more accurate than the PoL, but it can not cryptographically prove the location of the asset tag. Therefore, accurate auditable positioning requires a PoL and a LVS combined.

An LVS can detect attacks on positions and overcome the security issue of manipulated or forged positions. Typically, an LVS requires independently measured position information to verify claimed positions. When an attacker

successfully manipulates the claimed position, this manipulated position will contradict the independently measured position of the LVS. Therefore, the attacker must manipulate both the claimed position and the independent position information to manipulate a position unrecognized. Since the independently measured positions depend on the position of the same asset tag, the range of possible manipulations is even limited when the attacker successfully manipulates the claimed position and the independent verification information. Consequently, the complexity of the attack is increased, and the potential risk is limited by using an LVS. Furthermore, an LVS can classify attacked positions, drop these position claims, and raise alerts.

The literature covers various LVSs for VANETs, IoT, and WSNs. Wang et al. [104] reviewed existing secure localization algorithms in WSNs. Typically, the LVSs proposed in the literature assume that the tracked object has claimed a position, for example, measured by GPS, which the LVS then verifies by independent location information, like the known positions of the direct neighbors.

Wei et al. proposed two location verification algorithms for WSNs that perform both in-spot and in-region location verifications [105]. They proposed the Greedy Filtering by Matrix (GFM) and Greedy Filtering by Trustability-indicator (GFT) algorithms that rely on the inconsistencies between the sensors' claimed positions and the positions implied by their neighbors' observations. The GFM and GFT algorithms assume that the sensors have a fixed communication range and that the neighbor lists should contain all nodes within a sensor node's range. Therefore, an attack is detected when a node appears in the neighbor lists of nodes not within the claimed position range. Since GFM and GFT depend on the property that not all nodes are in the range of a tracked sensor, the accuracy depends on the density of deployed sensor nodes.

A Location Verification System with Directional Antennas (LVS-DA) has been proposed for VANETs [106]. In LVS-DA, the base stations of a VANET are equipped with multiple directional antennas, and the direction of a signal is estimated based on the Receiver Signal Strength (RSS) values measured on the different directional antennas. The known location of the base stations and the direction of the received signal is used to verify if a car's claimed position is true. Also, Hu et al. used directional antennas to prevent wormhole attacks in

WSNs [107].

An LVS has also been proposed for RSS-based IPS. Yan et al. proved that the unknown transmission power does not affect the detection performance of their RSS-based LVS algorithm Differential Likelihood Ratio Test (D-LRT) [108]. NN-LVS is a Neural Network based Location Verification System for Vehicular Networks [109]. NN-LVS depends on inconsistencies in the ToA at multiple base stations. The authors propose using AoA and RSS information to enhance their artificial intelligence model in future work. Liu et al. proposed the Malicious Node Detection Clustering (MNDC) and Enhanced Malicious Node Detection Clustering (EMDC) algorithms for secure range-based localization in WSNs [110]. MNDC is based on location clustering and consistency evaluation via sequential probability ratio testing. Furthermore, EMDC is an enhanced version of the MNDC that modifies the calculation of reference error.

Wu et al. propose a blocking level-based location verification scheme for WSNs [111]. They proposed using an anchor point, a polar coordinate system, and a moving obstacle to verify the nodes' location. As an obstacle, a metal plate rotates around the anchor point, and when the metal plate is in the line of sight to the asset tag, the RSS value drops. This drop in the RSS level is used to verify if the sensor is in the expected direction. MoVers is a location verification algorithm that uses the mobility of verifiers to verify locations [112]. For MoVers no additional hardware is required. However, this algorithm requires at least two moving verifiers to verify locations.

The study of the literature revealed different approaches to build LVSs. The LVS proposed in the related work are mainly application specific and depend on the used positioning system. Therefore, the LVSs have diverse properties. Table 2.2 provides an overview of the key properties of the related work.

The evaluation of the related work revealed a few drawbacks. First, most proposed LVS depend on additional hardware, like directional antennas or moving metal plates as obstacles, to measure independent location information to verify claimed locations. This additional hardware raises the total cost of the positioning solution. Furthermore, the independent location information is often measured with lower accuracy than the claimed position. Therefore, the

TABLE 2.2: Related Work [85] © 2022 IFIP

Work	Algorithm	Verification Information	Use Case
	SecureAoX	AoD	IPS, low-power
[105]	GFM, GFT	Expected list of neighbors based on fixed communication range	WSN
[106]	LVS-DA	Directional Antennas	VANET
[108]	D-LRT	RSS	WSN
[109]	NN-LVS	ToA	VANET, IoT
[110]	MNDC, EMDC	ToA and RSS	WSN
[111]	blocking obstacle	Moving obstacle to manipulate RSS in specific direction	WSN, IoT
[112]	MoVers	Moving verifiers	IoT, access control, environmental sensing

LVS only limits the range of possible manipulations. For example, directional antennas only verify that the direction of the asset tag is correct but not the distance. Finally, the proposed LVS are not suitable for BLE-based AoA positioning.

2.8 Conclusion

In this chapter, we presented the state-of-the-art needed to address the research questions of this thesis. Therefore, we provided an overview of the background and the related work required to understand the components of our proposed system distributed audit trail for the IoT. The study of the related work revealed a few drawbacks to be further investigated. In the following, we summarize the main drawbacks of the related work we address in this thesis.

Even if several DLT-based DPKIs have been proposed in the literature, most do not provide an integrated immutable signature store. However, to make declarations, measurement values, calibrations, and positions auditable, the signatures must be non-repudiable. Otherwise, a signer could delete a signature and claim that the signature never existed. In an auditable system, a

signer should be able to revoke signatures transparently but not delete them. Therefore, the signatures must be immutably stored.

Most related work depends on proofs-of-concept prototypes, simulations, and testbeds. However, the testbeds used in the related work to evaluate IoT applications are based on the tiny, dual-display computer Raspberry Pi. Compared to low-power microcontrollers, the Raspberry Pi is a powerful computer with relatively high energy consumption. Therefore, research is missing that evaluates the feasibility of a DLT-based audit trail for low-power devices. In this context, especially the energy overhead introduced by the required cryptographic functions should be evaluated.

In the literature, the proposed systems focus on building auditability for one specific component (DPKI, metrological traceability, auditable positioning). A broader and more abstract model of an audit trail should combine all these components in one audit trail. The different components could benefit from each other's data in a compound distributed audit trail. For example, for a third-party data consumer, the identity of the device that measured the data might be less important when it cannot verify the authenticity and integrity of data and has access to auditable quality information like calibration certificates. Therefore, quality information linked to the identity claim could improve a DPKI. Especially the related work covering metrological traceability and auditable positioning mention a PKI for the IoT as an open challenge. A more general distributed audit trail model enabling the integration of PKI and these applications is a promising approach for solving this open challenge.

The literature that discusses auditable and verifiable positioning covers multiple different positioning technologies. But, the BLE-based AoA positioning is not covered. However, BLE radio modules are cheap and energy efficient. Therefore, they are available in many IoT devices, and BLE-based AoA positioning should be auditable.

To overcome these issues in the related work and to answer the research questions, we design and evaluate the Veritaa framework in this thesis. Veritaa is a framework that enables a distributed audit trail for the IoT.

Chapter 3

A Distributed Ledger Technology based Distributed Public Key Infrastructure with a Signature Store

3.1 Introduction

This chapter presents the architecture of a DLT-based DPKI with a signature store to solve the challenges of audit trails, addressing research question 1: *How to build a DLT-based, energy-efficient, and scalable DPKI with a signature store?*

Each record of an audit trail is signed by its responsible entity. For digital audit trails, the auditors require a PKI to verify these signatures. This PKI is responsible for managing a public key during its life cycle. The life cycle of a public key includes its creation, authentication, certification, utilization, and deprecation. Since the auditor must be able to verify the validity of a signature in the past, it is crucial to know the state of the public key when a signature was created. Therefore, the change log of the public key's life cycle and the signatures must be stored non-repudiably and immutably to preserve the signature's validity. DLTs are a promising approach to immutably store a transaction log and, therefore, suitable for a DPKI with a signature store.

Therefore we proposed the Veritaa framework, a DLT-based DPKI with a signature store [113][1],[114] [2]. The proposed architecture is organized into the GoT representing the DPKI, the ABCG to immutably store graph transactions, and a peer-to-peer network to disseminate information. The GoT is built out of a few elementary graph transactions, which enable the distributed management of the DPKI. In contrast to the related work, the transactions of the GoT are very expressive and enable the integration of a PKI and a distributed audit trail in the same model. As we discuss in the following chapters, this integration of PKI and audit trail has the advantage of improving the quality of the PKI and the distributed audit trail.

The main contributions of this chapter are:

- Design and implementation of Veritaa, a system to manage digital identities and their signatures
- Design and implementation of the ABCG, a BlockDAG based DLT optimized to store append-only graph transactions
- Design of a reputation-based certification model based on the GoT
- The evaluation of the proposed architecture

3.2 System Model

Veritaa is a high-performance and scalable DPKI with a signature store. This means that it also integrates an immutable database to store public declarations that have been signed by key pairs managed by Veritaa. The Veritaa framework comprises the Graph of Trust (GoT), the Acyclic Block Confirmation Graph (ABCG) to store the GoT immutably, and a peer-to-peer network to connect multiple Veritaa nodes and exchange data.

[1] Partially reproduced in this chapter - © 2020 IEEE
[2] Partially reproduced in this chapter

3.2.1 The Graph of Trust

The GoT is a novel model that represents a distributed audit trail secured with a DLT. For this purpose, the entities contributing to the audit trail non-repudiably store and publish identity claims, data hashes, and declarations in the GoT. These elements can be used to represent real-world relations between entities and data.

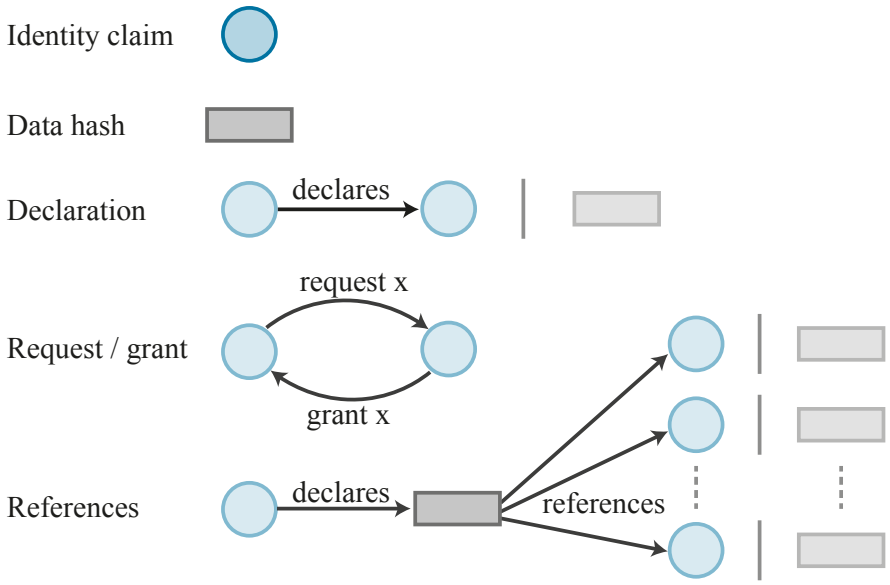


FIGURE 3.1: Elements of the GoT

Elements

The GoT is composed of a few elementary transactions. Each participant of the Veritaa network can create and commit such transactions. Therefore, the GoT is created collaboratively with all network participants. The elements of the GoT are simple but expressive. Therefore, these elements can be used to represent real-world relations between entities and the data hashes of digital documents. Figure 3.1 shows the elements of the GoT. The GoT consists of identity claims, data hashes, declarations, and request-grant declarations. An identity claim is a vertex of the GoT, which binds an entity's identifying information to the entity's public key. A declaration is a directed edge from an identity claim to another identity claim or a data hash. The request-grant declaration is a declaration between two identities where two entities show on the

GoT that they agree on a certain attribute x. Therefore, entity A signs a request attribute declaration towards entity B's identity claim. If entity B agrees to this request, it signs a grant declaration with the same attribute towards A. The request-grant declaration is only valid if both transactions are signed and not revoked. This request-grant delcarations are required to integrate IoT devices into the GoT and is discussed in more detail in Chapter 4. Furthermore, a declaration towards a data hash can reference other data hashes or identity claims. These references enable the signer to provide additional relational information to the declaration. For example, a calibration laboratory can reference the measurement instrument calibrated from the calibration certificate's hash, or an artificial intelligence algorithm can reference the raw data hashes from the processed data's hash. The reference relations are explained in more detail in the Chapters 5 and 6.

Authentication

An identity claim is a public key associated with the name of an entity to which it allegedly belongs. As each participant of the Veritaa network can create identity claims for all entities, the identity claims must be authenticated. To authenticate an identity claim, domain vetting can be used. In this context, we refer to domain vetting as the process of verifying that the identity claim is owned by the same entity that owns a specified domain. If the domain is known and trusted by the auditor, it can be used to bootstrap the reputation of this identity claim. To enable domain vetting, the creator can add a validation domain to an identity claim. To prove that the same entity owns the domain and the identity claim, the creator has to solve a challenge that only the domain owner can solve. As a challenge, the creator can either add the public key's hash to the well-known folder [115] in the HTTP URL of the domain or add this hash to a DNS TXT record. Each participant of the network can also authenticate its peers manually (for example, by phone or passport validation). When an agent has authenticated another agent's identity claim, it can publicly declare this authentication on the GoT with a signed trust declaration.

Certification

The mapping of digital entities to real-world organizations, devices, and people is a big challenge. In Veritaa, all entities with access to a Veritaa node can create identity claims. However, the authenticity of these identity claims is not yet certified, and the DPKI requires a model to validate and certify their identity. In Veritaa, we propose using trust, reputation, and domain validation information to certify identity claims. Domain validation proves that the creator of an identity claim owns a particular domain. When the used domain has a good reputation, for example, a company's domain, a third party can use this information to authenticate an identity claim.

Trust To authenticate public keys, an agent either has to trust a third party or authenticate each public key independently. Due to the huge number of identity claims, it is impossible to authenticate all public keys manually. Therefore, it is necessary to trust other entities to perform the authentication properly. In a centralized PKI, all have to trust that the central entity performs the authentication of identity claims correctly. Any event that destroys trust in this central trusted entity causes the centralized PKI to collapse. In contrast, in DPKIs, trust is distributed over all participating entities. Consequently, if some entities are not trusted anymore, this should only affect the identity claims certified by the entities that lost the network's trust. Therefore, a DPKI is more resilient to the failure of a certifying entity.

Trust is an important concept in the process of certifying the authenticity of third parties' public keys. Merriam-Webster defines trust as assured reliance on the character, ability, strength, or truth of someone or something [116]. A little more specific, Gambeta defines: trust is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before it can monitor such action (or independently of its capacity ever to be able to monitor it) and in a context in which it affects its own action [117]. Merriam-Webster and Gambeta define trust as a binary relation. An agent either trusts another agent or not. Gambeta's definition highlights that trust depends on a subjective probability threshold defined by the trusting agent. If a potentially trusting agent A thinks that the probability that a trusted agent B performs a particular action is above

this threshold, then A trusts B and otherwise not. It is up to A's discretion to define this threshold, and its level depends on the risk when this trust is wrong. Different agents have varying priorities, and consequently, the definition of this threshold is subjective. In the real world, the definition of this threshold and the estimation of the probability that a certain agent performs a particular action is also subjective. In literature, different trust measures are used to estimate this probability. Some authors propose to classify the trustworthiness of entities in distrust, ignorance, minimal, average, good, and complete, where the evaluated entity is compared with the rest of the entities [118]. Furthermore, others propose using the mode of authentication to define a confidence level. For example, authentication over a telephone line has a lower confidence level than a passport verification [53].

In this chapter, we limit the trust relations to a simple declaration of trust from entity A towards entity B and use an additional domain validation scoring to weigh the trust relationships. Nevertheless, a confidence level for trust could be added by simply weighing the trust declarations. Additionally, we define that the (honest) signatory should have authenticated the identity claim out of the band for each signed trust declaration. A trust declaration must never be signed on information derived from the GoT.

Gambeta's definition also mentions that trust is context-sensitive [117]. For example, trusting the mailman to deliver a letter does not imply trusting him to fly a plane. Some certainly can, but most cannot. In the process of authenticating identity claims in the GoT, two quite close but still very different contexts of trust can be found. The first context occurs when A signs a trust relationship towards B, then it assures that it has authenticated that the public key belongs to the entity it claims, and, therefore, it trusts this identity claim is valid. The second context of trust can be found when A trusts B that its trust statements are valid. Trust that an identity claim belongs to a certain entity is not the same as trusting an entity to authenticate identity claims properly. The WoT uses recommendation links to assess how much one trusts that another agent correctly authenticates third parties [53], [54]. However, even if this recommendation links in the WoT have a depth property that can be larger than one hop, the subjectivity of trust limits this scheme's applicability to one hop. Consequently, trust can be used to certify the authenticity of close identity claims,

but it is difficult to draw conclusions about identity claims where the trust chain is long.

To overcome trust subjectivity and the short range of trust chains, we propose to use the public trust declarations in the GoT as reputation voting and the domain validation information to derive certification information.

Reputation Trust describes what one agent thinks of another, and reputation describes how the public sees an agent. Merriam-Webster defines reputation as overall quality or character as seen or judged by people in general [119]. The public declarations of trust can be considered recommendations. From these recommendations, a reputation can be derived.

While trust is a subjective relationship that is difficult to evaluate, reputation is what the public thinks about an entity, and it can be measured. Algorithms like HITS [120], closeness centrality [121], Eigenvector centrality [122], EigenTrust [123], and PageRank [124] have been around for years and gained importance together with the fast growth of the Internet and Social Media. All these algorithms measure the centrality of nodes in a network. The centrality tells how well a node is connected in a network and how much influence it has. This influence can be considered as reputation in a graph with trust declarations.

If it is possible to build a reputation system that prevents attackers from gaining an unjustified reputation, this reputation system can be used for distributed certification. The advantage of reputation is that it can be used to reason about the authenticity of distant identity claims.

While reputation can certify the authenticity of distant identity claims, trust can be used to certify the authenticity of identity claims with short certification paths. For most applications, a combination of reputation and trust will be meaningful. For example, let us assume that an agent wants to authenticate the signature of a local weather station. Since this local weather station is only a device, it is not domain validated, and it does not have many incoming trust declarations. However, the countrywide meteorology service is well-reputed and has a domain-validated identity claim. The agent is now able to authenticate the meteorology service over reputation. From the countrywide

meteorology service's identity claim, the agent can now follow the trust declaration to the regional department's identity claim and, therefore, authenticate this regional department's identity claim. Finally, the agent finds a trust declaration from the regional identity claim to the weather station. Since all the trust declarations followed in this example are within the same organization, the subjectivity remains limited. This example shows a distributed approach to certify the authenticity of the organization's identity claim and then uses a centralized approach to certify the identities within the organization's infrastructure.

Distributed Certification with Trust and Reputation In this paragraph, we discuss how trust and reputation are used in Veritaa to certify the authenticity of identity claims.

For the certification, attackers must not be able to manipulate their own reputation. In Veritaa, each full node has an immutable copy of the GoT and is able to calculate each identity claim's reputation. This reputation is calculated with reputation functions discussed in the previous section. Since its creator must sign each trust declaration, an attacker cannot forge trust declarations for other identity claims. However, with unweighted trust declarations, attackers could form malicious clusters where they sign trust declarations to each other's identity claims and accumulate reputation. In order to prevent these malicious clusters from gaining reputation, clusters containing identity claims known, trusted, or authenticated by the auditor must be weighted higher than clusters with all unknown identity claims. To calculate the weights, we use a score function. First, the identity claims must be ranked by the auditor's level of trust, and then a higher score must be attributed to the higher-ranked identity claims. In this work, we use domain validation information to rank identity claims. Therefore, domain vetting is used to authenticate that the same entity owns a particular domain and an identity claim. The auditor then uses an ordered list of domains to rank the identity claims accordingly. Depending on the application, this list could be the list of the globally most visited websites [125], domains relevant to metrology, or any other arbitrary list containing trustworthy domains for the given application. The ranks of the identity claims are then used to calculate a score that can be used to weigh trust votes.

To score the domains, we use the function shown in Equation 3.1 where $v \in V$ are all identity claims, m is the number of domain validated identity claims, $n = |V|$, and $rank(v) \in \{0, 1, ..., m\}$ is the ranking function that assigns a rank to each identity claim. The best-ranked domain is mapped to the lowest value. Multiple methods can be used to rank the domains. For example, the traffic of domains or an arbitrary list of the validator could be used.

$$score(v) = \begin{cases} \frac{1}{m}(m - rank(v)) & \text{if v is domain validated} \\ \\ \frac{1}{n} & \text{else} \end{cases} \tag{3.1}$$

The *score* function is used to define the trust relations' weight and initialize the random walks of the different Eigenvector-based algorithms. Therefore, we define C as the weighted adjacency matrix. Let A be the adjacency matrix of all identity claims $v \in V$ and trust links from the GoT. Equation 3.2 shows how C is defined.

$$c_{ij} = a_{ij} \cdot score(v_i) \tag{3.2}$$

With this weighted adjacency matrix C, the reputation is calculated. A reputation function is a function that maps the weighted adjacency matrix together with domain scoring to a vector that contains the reputation of all identity claims.

In this chapter, we apply the Eigenvector Centrality [122], EigenTrust [123], and PageRank [124] on the weighted adjacency matrix C and initialize the random walks with the score function. The reputation distribution that results from the reputation function can be used for distributed certification, where the more influential nodes have a higher reputation according to the algorithm. In the evaluation, we show that the algorithms can keep the reputation of attackers low (see Section 3.4.5). Since all identity claims will get some reputation, a threshold is required to decide which identity claims are certified. For example, a reputation threshold could be defined, or it could be defined that only those identity claims are certified that have a higher reputation than

a specific domain-validated identity claim. The choice for the right threshold is at the auditor's discretion and depends on the risk of a wrong certification.

Certificate Revocation

Since the private key that belongs to an identity claim is not in the scope of Veritaa's security, it must be assumed that some keys are lost or compromised. Therefore, the GoT must be able to revoke broken certificates. In non-DLT-based DPKIs, revocation lists are usually used to revoke broken certificates. These lists are challenging to maintain, and the time to disseminate the revocation information is typically long [126]. DLT-based DPKIs have the advantage that certificate revocation can directly be stored on the DLT.

FIGURE 3.2: Self-revocation of an identity claim

In Veritaa, an owner of a private key can directly revoke the corresponding certificate on the DLT. Figure 3.2 shows entity A on the GoT that has posted a self-revocation to declare that the identity claim is revoked. Each graph reader can directly recognize that this public key is no longer valid.

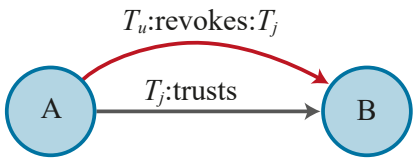


FIGURE 3.3: Revocation of a declaration

An entity can also revoke the declarations it has made. Figure 3.3 shows an example of a revoked transaction. With transaction T_j, entity A signed a declaration of trust towards entity B. Entity A signed the revocation transaction T_u that references the revoked transaction T_j to revoke this transaction. Note that it is possible to revoke all kinds of transactions with this approach. A revokes transaction T_u is only valid if it was signed by the same entity that has signed the original transaction T_j. A reader of the GoT considers a declaration as valid if it is signed by the identity claim it starts from and as long as no revocation

transaction exists. Note auditors validating the system's state in history will consider the transaction as valid in the time between the original transaction T_j was created and the time the revocation transaction T_u was signed. Therefore, when a declaration has been valid for a certain amount of time, the auditors are able to verify that. This enables a transparent revocation of transactions.

Signed Declarations

Besides the public key management, Veritaa can also be used to immutably store signed declarations. By securing the declarations on the GoT, it is possible to determine the signing key's life-cycle state when the declaration has been signed. This has the advantage that the declaration can still be audited when the key is deprecated.

On the GoT, different declarations can be signed. A declaration is a relation between two identity claims or an identity claim and a data hash. The relation always starts from the signing identity claim and ends in another identity claim or a data hash. The declarations have a type of a finite set. For example, the type of relations between two identity claims is in the set $\{trusts, audits, validates\}$, and the type of relations between an identity claim and a data hash is in the set $\{issues, approves, signs\}$.

A hash function is used to generate data hashes to enable declarations referencing objects that are not part of the GoT. Since the hash of data changes when the data is changed, the declarations are always for a specific version. Therefore, verifying if a declaration has been changed after the declaration towards the hash was signed is trivial.

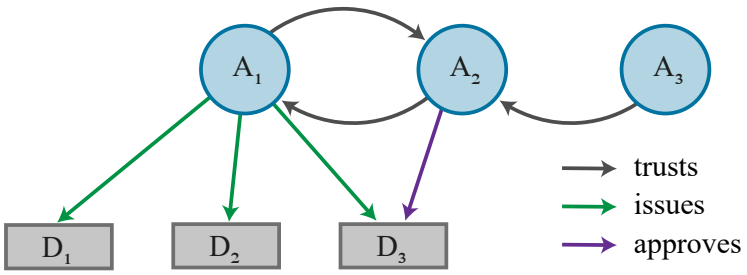


FIGURE 3.4: Document signatures on the GoT

Figure 3.4 shows an example where A_1 has issued three documents. A_2 declares by the approval of D_3 that it agrees with the contents of this document. Since A_3 has authenticated the identity claim of A_2 it can verify that A_2 has signed an approval declaration D_3.

3.2.2 Acyclic Block Confirmation Graph

To ensure the integrity of the GoT, it is of the highest importance that all transactions can neither be reversed nor altered. Erroneous transactions must be transparently revoked so that auditors can comprehend all changes. Therefore, the database that stores the GoT must immutably store all transactions that build the GoT. Veritaa uses the ABCG as DLT to immutably store the GoT. The ABCG is an application-specific BlockDAG based DLT optimized to store the append-only transactions to that build the GoT.

Transactions

The GoT is built out of append-only transactions that contain either an identity claim or a relationship between two identity claims or an identity claim and a data hash. Each transaction that contains an identity claim is self-signed by the contained identity claim. The self-signature proves that a matching private key for the identity claim exists. Each transaction that contains a relationship is signed by the identity claim where the relationship starts. This ensures that only owners of the private key can create declarations for an identity claim. All network participants should be able to create new identity claims, and all private key owners of an identity claim can create and sign declarations towards all hashes.

Blocks

The ABCG is a BlockDAG where each block confirms three blocks. The advantage of the ABCG over a single blockchain is that new blocks do not necessarily need to be appended at the tips of the ABCG. Consequently, the nodes can always commit new blocks, and the creators must not compete. When the honest nodes agree on always confirming the three blocks with locally the least

confirmations, all blocks should be confirmed over time. A block with no confirmation at all is called a tip [21]. In a peer-to-peer network, the information does not propagate instantaneously, and therefore, the nodes are not always synchronized perfectly. Some nodes might have tips in their local copy of the DLT that are already confirmed at other nodes. Consequently, not all nodes share the same tips. The total number of tips in the network depends on the time required for a new block to disseminate through the whole network and the number of blocks confirmed by each new block [21]. Nevertheless, since Veritaa only requires transactions to be replicated on all nodes and immutably stored, as long as the blocks are confirmed, the number of tips does not affect the operation of Veritaa.

Confirmation is of fundamental importance to ensure the non-repudiability and immutability of the transactions. The confirmation of a block is done by adding the hash of the confirmed block in the confirming block's header. As the hashes of the confirmed blocks are contained in the block's hash, it is secured that the previous blocks cannot be changed. If a block in the ABCG was changed, its hash would change, and, therefore, the consecutive blocks would point to a non-existing block. A hash tree secures the immutability of the transactions in a block. Figure 3.5 shows an example of some blocks. Similar to the blocks, each transaction contains the hash of the previous transaction. The hash of the last transaction T_{Xn} is contained in the block hash, and, therefore, the block hash would change if a single transaction was changed. This hash tree ensures that all transactions cannot be altered after a block is confirmed.

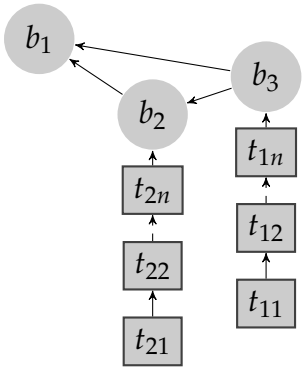


FIGURE 3.5: Hash tree of a block of the ABCG [113] © 2020 IEEE

Consensus

In Veritaa, consensus is used to achieve the same state of the ABCG and the GoT on all synchronized nodes in the presence of malicious or faulty nodes. Honest nodes only append and forward valid blocks to reach a consensus. A block is valid if all of the following conditions are met:

- All transactions and the hash tree in the block are valid.
- The block confirms at least three valid and no invalid blocks.
- The block is signed by an identity claim that exists in the local copy of the ABCG or the block to add.
- For public Veritaa networks, a difficulty requirement (PoW) can be set to increase the computational cost to create blocks and protect the network from spam. Note that since the nodes do not need to compete for the next block, this difficulty does not lead to an arms race.

Two nodes are synchronized when they have the same tips. When two nodes have the same tips, they have the same set of transactions in their local copy of the ABCG. The transactions contain the instructions to build the GoT. Since only transactions that either add a vertex or an edge to the GoT exist, the order of execution does not matter. For simplicity, add first all vertices and then the edges. Consequently, two synchronized nodes that apply all transactions result in the same state of the GoT.

3.2.3 Time Considerations

Veritaa with the GoT is responsible for the management and authentication of identity claims. Each identity claim has a life cycle that follows the following states: inception, authentication, certification, utilization, and deprecation. Several reasons can lead to the deprecation of a key pair. A key pair could be replaced due to its age, a lost private key, or when the private key has been compromised. Signatures that have been created with an identity claim are only valid if the identity claim was certified and not deprecated at the time of signing. It is essential to know the identity claim's state at the time of a signature's creation to verify its validity. Additionally, assigning a timestamp to

each signature should be possible to map digital signatures with real-world events.

In the context of the DPKI presented in this chapter, we distinguish three different times. First, when an event happened in the real world. Second, when the transaction related to this event is committed in a block to the ABCG. Finally, the time when the block reached its finality (i.e., it was immutably written to the distributed ledger).

When a new block is committed to the Veritaa network, a gossip-based protocol is used to disseminate it. With gossip-based dissemination, data is spread exponentially fast through the network [127]. With this quick dissemination of new blocks, and all valid blocks accepted by honest nodes, the time between a new block has been committed to the network, and its finality is short. However, note that the time between a real-world event and its first occurrence on the ledger is hardly controllable by a distributed ledger.

In the ABCG, new blocks can be appended to all valid blocks. Therefore, new blocks can also be added to old blocks, and not only to tips. This property makes the protocol fault-tolerant and enables applications with no permanent or direct Internet connection, for example, behind firewalls, in low-power applications, or at remote sites. However, the DAG structure of the ABCG does neither provide a total order of transactions nor time information.

In Veritaa, block discovery services are used to enhance the ABCG with time information. A block discovery service creates a timestamp in a configurable interval. Each block discovery service can use its own interval, and each full node can run such a service.

A valid timestamp transaction is always signed by a valid identity claim found on the GoT, and its time is always greater than the time of its previous timestamp transaction. Additionally, all blocks with a timestamp transaction of a honest operator will always confirm all tips. By confirming all tips, the ABCG is divided into partitions, and each block can uniquely be assigned to the partition that belongs to a specific interval.

Each timestamp block can be considered a snapshot. Since a honest time discovery service confirms all tips, it is possible to reach all valid blocks that have

been in the local ABCG copy starting from a given timestamp. The Breadth First Search (BFS) algorithm can be used [128] to discover all blocks that can be reached from a given timestamp. The time of the first timestamp that discovered a block is the block's discovery time. The discovery time is used to sort the blocks.

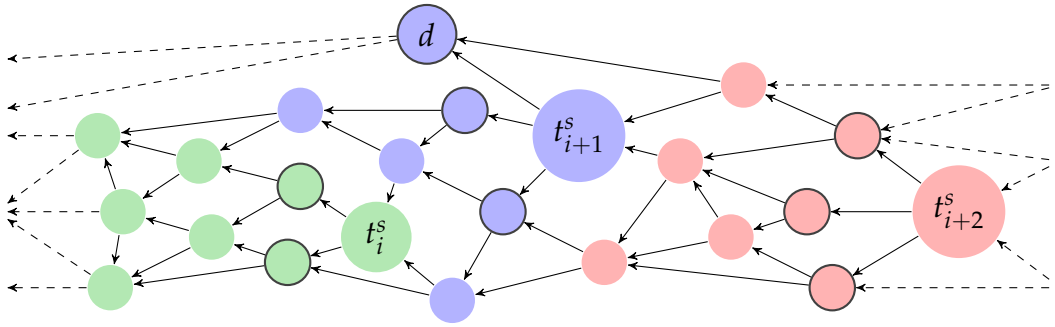


FIGURE 3.6: Fragmentation of the ABCG by discovery services [114]

Figure 3.6 shows a small section of the ABCG and explains how example timestamp blocks partition the GoT. The dashed arrows on the left indicate the references to the blocks of the previous intervals, and the dashed arrows on the right are the references from the subsequent blocks. The timestamps in this example belong to the same discovery service. To make this example graph more readable, the blocks confirm only two previous blocks instead of three, like in our implementation. The blocks with a border are the tips at the time when the timestamp was created. The assignment of blocks to timestamps is straightforward. All blocks that can be reached from the timestamp i of the discovery service s are assigned to timestamp t^s_i if they are not already assigned to another timestamp t^s_j with $t^s_j < t^s_i$. With this algorithm, the green blocks belong to timestamp t^s_i, the blue to t^s_{i+1} and the red to t^s_{i+2}. The interval terminated by t^s_{i+1} also shows an example of a newly discovered block d that has been appended to two much older blocks, indicated by the dashed arrows that reference back to the past. The reasons for this delay can be manifold. For example, the block was delivered with an opportunistic routing protocol, the creating node did not have a permanent Internet connection, or the block was intentionally appended to older blocks. However, this is no problem with the proposed algorithm and block d will be assigned to the first timestamp after it has been discovered on the ABCG.

All full nodes of the Veritaa network can start a discovery service. With multiple discovery services in the network and all timestamps publicly available on the ABCG, each reader of the ABCG can assign each block uniquely to one interval of each discovery service. The different timestamps assigned to one block can be considered as a vote for when the network discovered the block.

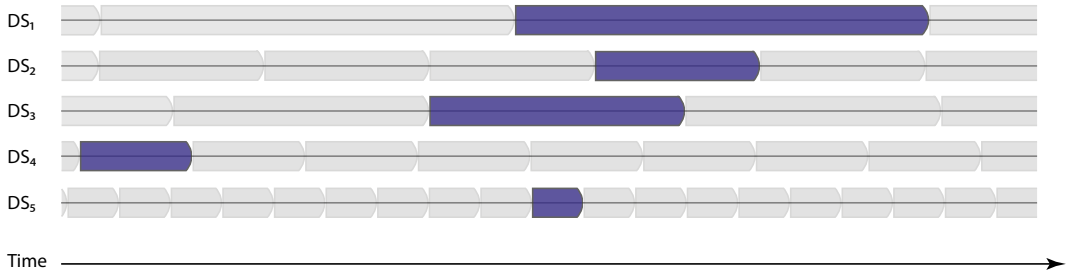


FIGURE 3.7: Time assignment [114]

Figure 3.7 shows an example of five discovery services DS_i. Each service uses a different discovery interval. The interval when the block b has been discovered is highlighted in violet. Due to inaccurate clocks, propagation delays, or even manipulation, the block discovery services DS_i assign different timestamps to b. These timestamps and the reputation of the block discovery services' identity claims can be used to build consensus about the timestamp of b.

To build consensus about the block's timestamp, the auditor can calculate the median of the timestamps of the n most reputed discovery services. For example, let us assume $n = 3$ and DS_1, DS_2, and DS_4 are the most trusted discovery services in the example of Figure 3.7. Then the timestamp of DS_2 would be assigned to the block in question.

With the proposed approach, the blocks can be ordered and timestamped up to a certain resolution. The resolution depends on the interval length and the number of trusted discovery services. There is a tradeoff between the time resolution and the overhead generated by the timestamp required to achieve this resolution. By using different discovery intervals, the requirements for many applications can be met. For example, for the signature on a working contract, a daily timestamp might be sufficient, and for an auditable sensor infrastructure, a resolution below one minute might be useful. If the absolute ordering of blocks is required, the GoT could be implemented on a general-purpose DLT at the cost of energy efficiency and flexibility.

3.3 Implementation

Veritaa is a distributed system where several nodes exchange information over a peer-to-peer network. Figure 3.8 shows the implementation of a Veritaa node. Applications can access the core functionalities of Veritaa over the northbound (NB) consumer API. A web application is used to operate the Veritaa node in the reference implementation.

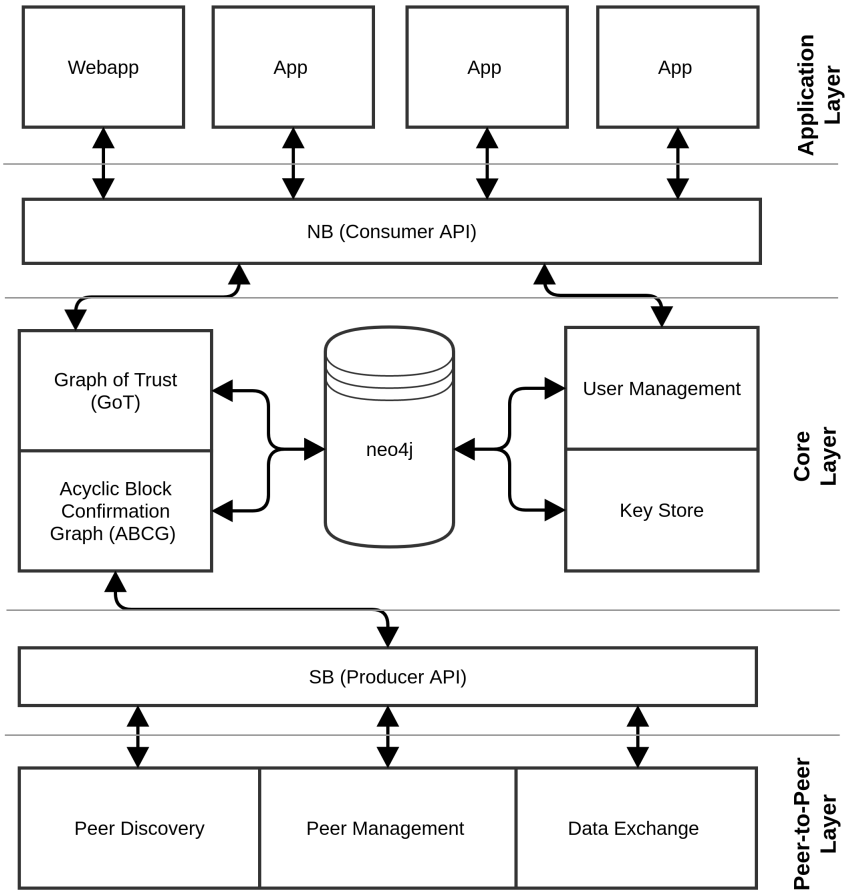


FIGURE 3.8: Implementation of a Veritaa Node [113] © 2020 IEEE

The northbound API is a REST API. In the core layer, the actions received by the NB API are executed on the GoT or the user management component. The user data and the private keys are not stored in the ledger and are not exchanged over the peer-to-peer network. Neo4j is used to store the data locally. Neo4j is a graph database optimized for storing and accessing data with graph

structure [129]. The ABCG consists of a block scheme that references three previous blocks. These confirmations ensure the immutability of the distributed ledger.

We use the Graph Data Science (GDS)[130] library of neo4j and the Efficient Java Matrix Library (EJML)[131] to calculate the reputation. For encryption and signatures, we use ECDSA. As hash function, we use SHA2-256.

The peer-to-peer layer is used to exchange blocks with peers. The peer-to-peer layer consists of peer discovery, peer management, and data exchange components. The peer-to-peer network is unstructured and was implemented with Java IO and UDP datagrams. Each node has a predefined list of known hosts. This list is used to connect to the peer-to-peer network. At startup, a node selects one host from the list of known hosts and requests new peers. The node then connects to some of these received nodes. By continuously discovering the neighborhood, each node maintains a set of peers. The peers are selected by their availability and round trip time.

3.4 Evaluation

3.4.1 Testbed

To evaluate Veritaa and all its components, a testbed was created. If not other mentioned, the testbed was deployed to a single-station Kubernetes server with 128GB RAM and an Intel Xeon E5-1650V2. It was possible to deploy up to 70 Veritaa nodes on this setup, but bigger networks are possible by vertically scaling the Kubernetes cluster. Figure 3.9 shows the testbed consisting of a Kubernetes cluster that runs multiple Veritaa nodes as pods. A specialized root pod initializes the ABCG and acts as the known host. Multiple other nodes are created by replication.

When the testbed is started, the root pod initializes the ABCG, and afterwards, the other pods are started. To connect to the network, pods request an initial set of peers from the root pod, and subsequently, more peers are requested from existing peers. Each pod contains an activity simulation component. This

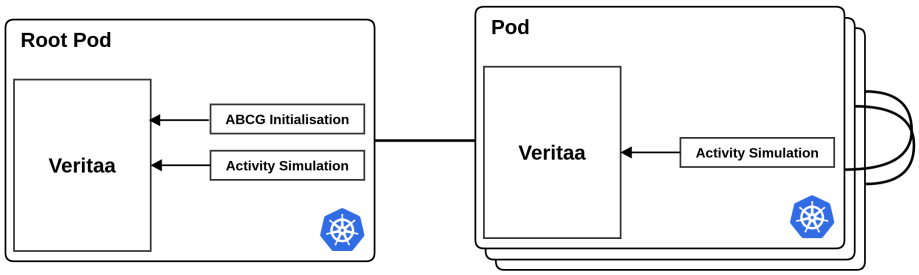


FIGURE 3.9: Veritaa testbed [113] © 2020 IEEE

component is used to generate traffic in Veritaa. To generate traffic, nodes create and sign random transactions and post them to the Veritaa network.

3.4.2 Stability

To measure the network's stability, the number of tips and the clustering coefficient [132] of the peer-to-peer network are analyzed.

The clustering coefficient defines how well a network is connected. Let $G = (V, E)$ be a graph where V is the set of vertexes and $E \subseteq V \times V$ is the set of edges. The neighbors of $v \in V$ are defined as $N(v) = u \in V, (u, v) \in E$. The degree of v is defined as $d(v) = |N(v)|$. Equation 3.3 shows the clustering coefficient $cc(v)$ of a vertex $v \in V$ where $E_{N(v)} = E \cap (N(v) \times N(v))$ is the set of links between neighbors of v [132].

$$cc(v) = \frac{2|E_{N(v)}|}{d(v)(d(v) - 1)} \tag{3.3}$$

The clustering coefficient $cc(G)$ of a graph G, shown in Equation 3.4 is the average of the clustering coefficients of its vertices [132].

$$cc(G) = \frac{\sum_{v \in V} cc(v)}{|v \in V, d(v) > 2|} \tag{3.4}$$

Tips are blocks that are not confirmed yet. The number of tips in the network depends on the network's connectivity, the propagation delay of new blocks in the network, the number of confirmations per block, and the block creation rate. In a congested network, the number of tips will increase as other nodes

do not know about the existence of new blocks and, therefore, confirm blocks that are already confirmed on other nodes.

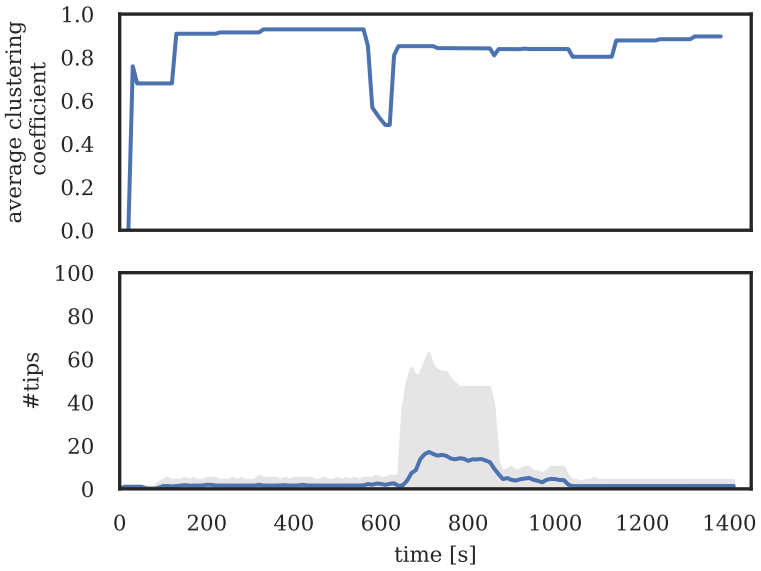


FIGURE 3.10: Evaluation of churn [113] © 2020 IEEE

To evaluate the stability of a distributed system, the simulation of joining and leaving (churn) nodes is an interesting experiment. In the following experiment, 25 Veritaa nodes were deployed in the testbed. Each node maintains a set of six peers and creates a block every 100 ms. After 500 s, 25 additional nodes were started, and after 1000 s, they were removed again. Figure 3.10 shows the average clustering coefficient [132] in the network. When the 25 additional nodes are added, the clustering coefficient drops because the nodes connect to the well-known node, which forms a cluster around this node. As soon as the additional nodes have received their peers from the well-known node, they connect to the peers, and the cluster dissolves. The impact of the removal (churn) of 25 nodes after 1000 s can be seen but has only a little impact on the average cluster coefficient as nodes are randomly removed, and all nodes are well connected with six peers out of 50.

The response of the number of tips to the joining and leaving nodes shown in Figure 3.10 is also notable. The blue line represents the average, and the grey represents the maximum and the minimum number of tips in the network (due to propagation delays, the nodes might not see the same tips). Initially, the root pod is started, and it initializes the network. Then, the initial 25 nodes

join the network. Since the global ABCG is empty at this point, no blocks need to be synchronized. When the 25 additional nodes join after 500 s, they first need to create their set of peers and synchronize with the network. To synchronize with the network, the nodes need to load all blocks of the ABCG from their peers. During the synchronization process, blocks are added to the local ABCG of a node. These blocks actually already have been confirmed on other nodes, but the confirming blocks are not loaded yet. These locally unconfirmed blocks result in a large number of tips during the synchronization process. After synchronizing the newly connected nodes, the average number of tips goes down and remains slightly higher than before the nodes joined. The higher tip count is the result of a higher propagation delay due to the increased network size. After we removed the 25 nodes, the average number of tips falls back to the level before the nodes' addition. This experiment shows that Veritaa is robust and resilient against heavy network change.

3.4.3 Throughput

The number of transactions that a distributed ledger can handle is the limiting factor of its scalability. In peer-to-peer applications like Veritaa, all nodes can create blocks anytime, but every node needs to process and store all new blocks. Therefore, the network could create blocks at a higher rate than a single node is able to validate them. Since all full nodes of Veritaa must validate and secure all blocks, the maximum number of incoming transactions that a full node can process forms an upper bound for the throughput. In order to measure this upper bound, a consumer and a producer node with 4 CPU cores each were deployed to the testbed. The producer node generates new blocks that are sent to the consumer node. The throughput was measured on the consumer node.

Figure 3.11 shows how much time was spent in the different processing states when a packet is received. Most time is required to process the block content. When the block content is processed, the GoT is updated according to the transactions in the block. To update the GoT, existing entities need to be read from the database. These reads are computationally relatively expensive, but the read time can be optimized if multiple data hashes are searched, and, therefore, the database read overhead per transaction becomes smaller when the

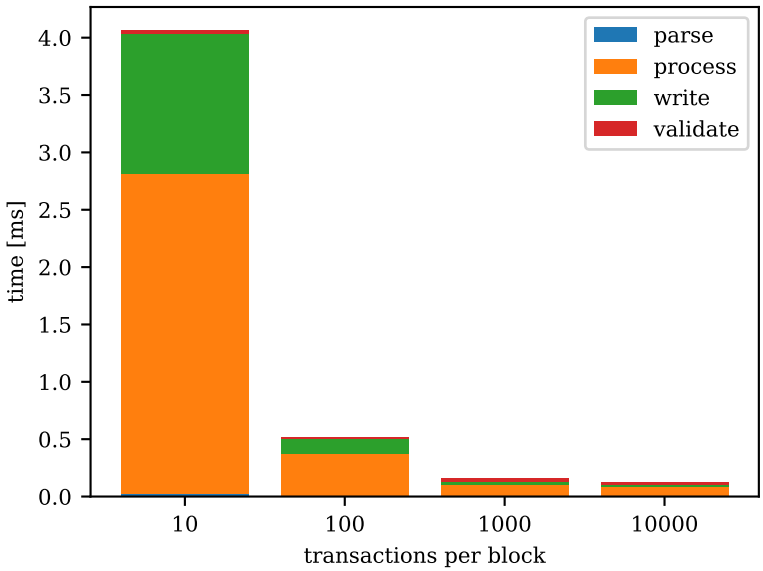


FIGURE 3.11: Average time to process a transaction [113]
© 2020 IEEE

block contains more transactions. The same accounts for writing the changes into the database. The validation process of the transactions only requires a small amount of time. As all transactions are validated separately, the validation time per transaction does not decrease much when more transactions are contained in a block. The time required to parse a block is minimal and can be neglected when the block size is increased.

Figure 3.12 shows the throughput of Veritaa in relation to the block size. The blue line represents the average and the grey line indicates the standard error. The throughput is decreasing due to overhead for a block size smaller than 1000 transactions. However, above 1000 transactions per block, the throughput is around 7000tps. The fluctuations mainly result from the database access speed that depends on the stored data.

In Figure 3.13, the throughput of some prominent DLTs is compared with Veritaa. For this experiment, the throughputs reported in the literature were used for the comparison. The throughput of Bitcoin is limited to 7tps by the 10 minutes block interval and fixed block size [28]. Depending on the block size and the number of vCPUs, Hyperledger Fabric can achieve a throughput of up to 4000tps [20]. The IOTA team states that they could measure up to 182tps but

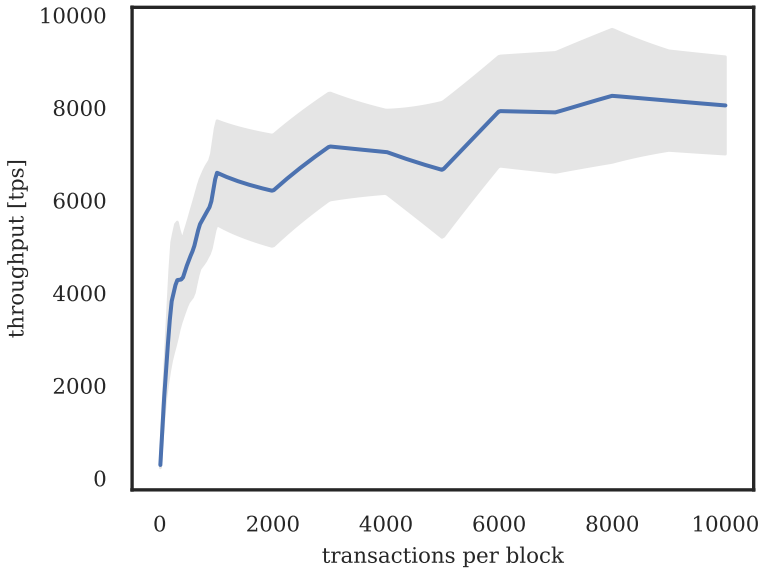


FIGURE 3.12: Throughput in relation to the block size [113]

aim to achieve 1000tps [133]. A private Ethereum blockchain can achieve a throughput of 76tps [134].

The direct comparison of Veritaa with other DLTs has some limitations. The target applications of Veritaa are DPKIs and an audit trail where a consensus with fewer constraints than in a general-purpose DLT can be used. However, an application built on top of a DLT cannot have higher throughput than the underlying system. Therefore, Veritaa outperforms all compared DLTs for the target applications of a DPKI and a distributed audit trail.

3.4.4 Block Discovery Services

In this section, we evaluate the block discovery services. For this evaluation, we have deployed Veritaa nodes with block discovery services enabled to a Kubernetes cluster distributed over four data centers one in Finland, two in Germany, and one in Switzerland. In this experiment, we evaluate the discovery delay with different settings. The discovery delay is the time between the creation of a block and its discovery in the network by block discovery services. Each block discovery service has a unique timestamp that first discovers a given block, and from all these timestamps, the discovery time is calculated.

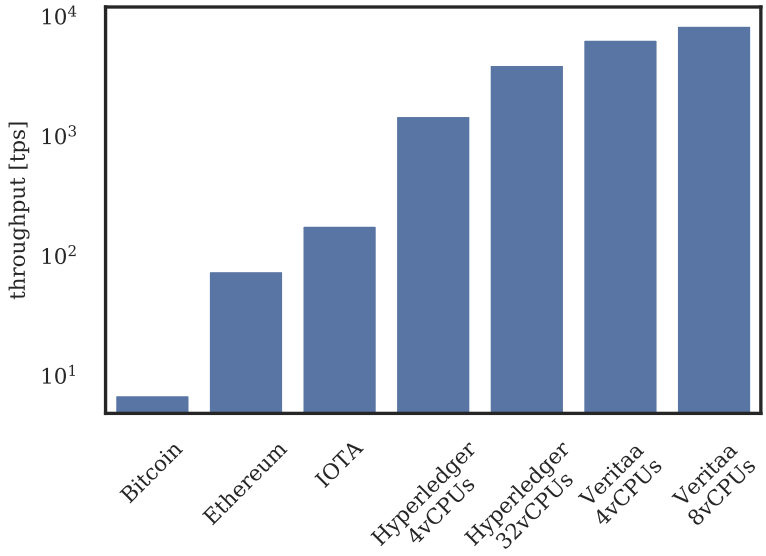


FIGURE 3.13: Throughput comparison with other distributed ledgers [113] © 2020 IEEE

This experiment compares the minimum, median, and maximum functions to derive the discovery time from all timestamps that first discovered a given block. In this experiment, all block discovery services have the same timestamp interval. The timestamp interval is the time between the creation of two timestamps by the same block discovery service. Figure 3.14 a) shows the discovery delay with a 60 s timestamp interval and b) shows the discovery delay with a 600 s timestamp interval. The median function is the most robust against attacks and has an average discovery delay of approximately half the interval length. The maximum function is the timestamp of the block discovery service that has discovered a given block at the latest. Consequently, with the maximum function, the mean of all discovery times is close to the interval length. Since the minimum function returns the first timestamp that discovered a block, the discovery delay is close to zero.

3.4.5 Security Analysis

Adversaries might attempt to attack the peer-to-peer network, the ABCG, or the GoT of Veritaa. The goal of an adversary attacking Veritaa might be: to manipulate the GoT in order to gain a higher reputation, abuse a compromised key, or interfere with the operation of the DPKI. We assume that the

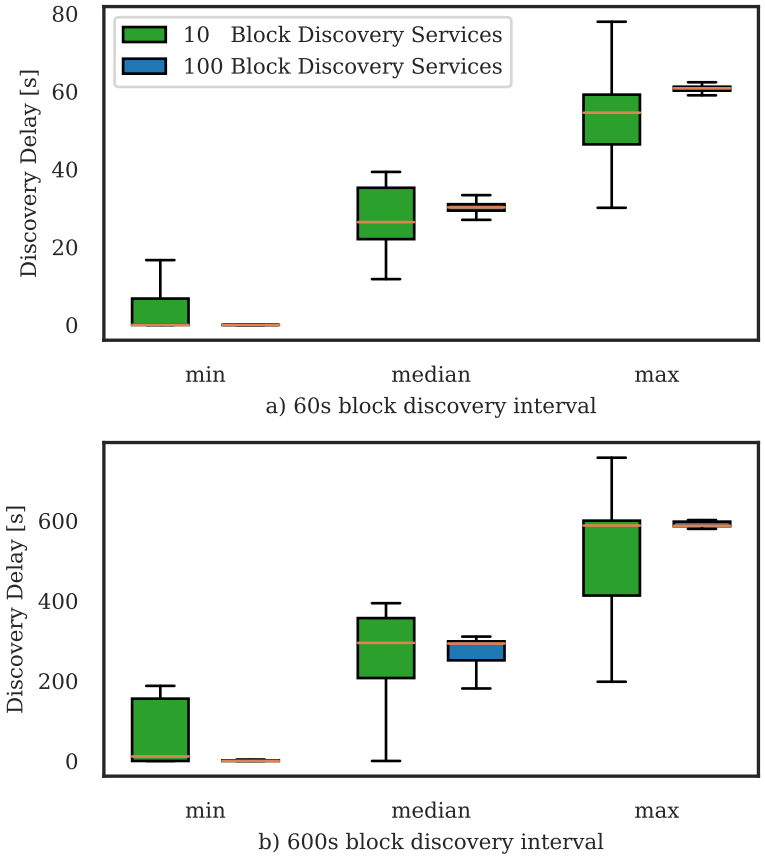


FIGURE 3.14: Block discovery delays [114]

used cryptographic functions are secure (hash functions and signatures). We also assume that each honest node can connect to the peer-to-peer network containing other honest nodes. Some of the honest nodes operate a block discovery service. Most but not all nodes are honest but curious. An adversary can eavesdrop, modify and forge transactions. Note that each node can check if a block is valid.

Distributed Denial of Service Attack

An attacker might launch a Distributed Denial of Service (DDoS) attack against some nodes in the peer-to-peer network to interfere with the operation of the DPKI. However, while the adversary might successfully take some nodes out of operation, it is unlikely to attack the whole network successfully. Since each node maintains a replica of the DLT, the DPKI remains operational.

An adversary could also try to interfere with the operation of the DPKI on the ABCG layer. In an unpermissioned network, an adversary might create a vast amount of valid but spam transactions and send them to the network. Since all Veritaa nodes maintain a replica of the ABCG, all full nodes would require adding these transactions. The resources for adding and storing those nonsense transactions would increase the operational costs. In order to prevent spam in unpermissioned networks, it is possible to add a PoW requirement to blocks. With this PoW requirement, blocks are only confirmed by honest nodes if they fulfill a particular difficulty. Thus, the PoW can increase the cost of creating spam to a level where the cost of creating spam is higher than its benefits. Note that this PoW requirement does not lead to an arms race since there is no competition for mining the next block. In a permissioned network, it is possible to revoke the authorization of suspicious nodes.

Reputation

In this chapter, we propose using domain validation information and trust declarations to build a reputation system for the certification of identity claims. A reputation algorithm assigns each identity claim a value so that the most reputed identity claim has the highest and the least reputed one has the lowest value.

To calculate all identity claims' reputation, we use the well-known Eigenvector Centrality [122], EigenTrust [123], and PageRank [124] algorithms. These algorithms basically represent a random surfer that traverses the graph and returns the probabilities in which state the surfer most likely is. In a directed graph, the random surfer might be locked in a cluster with no way out, and, therefore, PageRank and EigenTrust use a probability to start over at a random node at each step.

Attackers might try to build malicious collectives to manipulate and improve their reputation and certify illicit identity claims. This section evaluates if such a reputation function can be used to prevent these kinds of attacks.

Figure 3.15 shows an example with a collective of five attackers (red) that try to manipulate the reputation by voting for each other. The cluster with the honest identity claims contains two domain-validated identity claims (green), and two identity claims of the honest cluster have signed an erroneous trust link towards the attackers. Since they want to trap the random walker to get more reputation, the attackers do not create trust links towards the honest cluster.

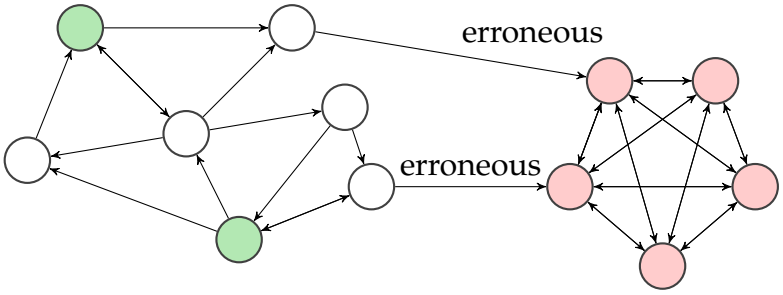


FIGURE 3.15: An example GoT with a group of attackers [114]

To improve the resilience against attacks, we initialize the centrality algorithms used to calculate the reputation with the domain scoring vector, weight the edges according to the domain scoring, and use the domain scoring for a probability where a node starts over. Consequently, the random walker is more likely to start at domain-validated identity claims and remain close to them. Furthermore, when the random walker gets stuck in a cluster, it is also more likely to start over at domain-validated identity claims.

Figure 3.16 shows example reputation distributions for Eigenvector, PageRank, and EigenTrust algorithms executed on a GoT with 100 identity claims,

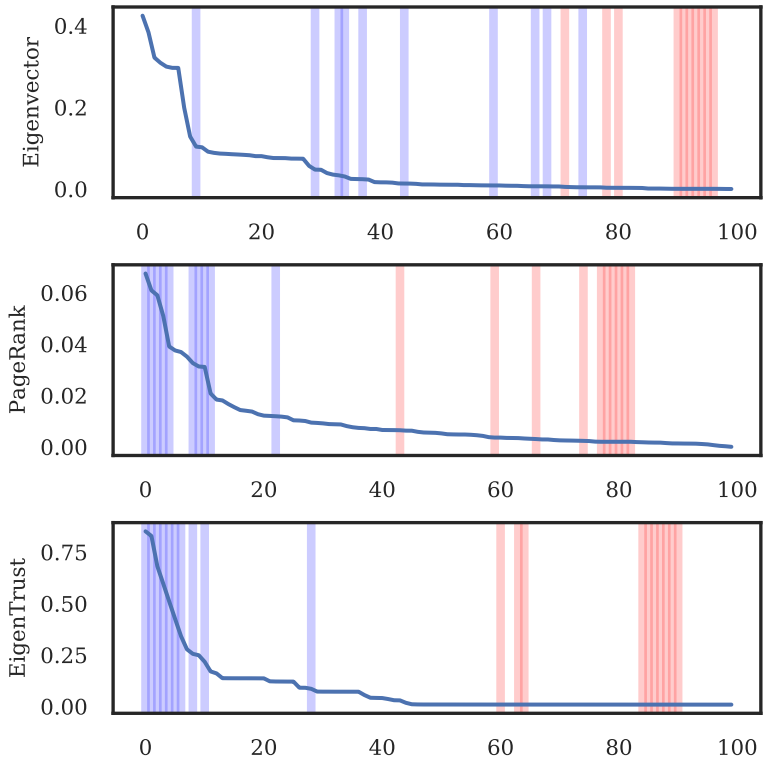


FIGURE 3.16: Reputation Distribution [114]

of which 10 are domain validated (blue) and 10 are attackers (red). In this example, five erroneous trust links have been created towards the attacking cluster. The example in Figure 3.16 shows that the proposed reputation function based on the trust declaration and the domain validation information can keep the reputation of the attackers low while the domain-validated nodes have a higher reputation. It also shows that they do not perform equally. An identity claim can have a large Eigenvector centrality if it has important or many neighbors. Therefore, the domain-validated identity claims might not be found in the top ranks. With the PageRank and the EigenTrust algorithms, the domain-validated identity claims are ranked higher, but some attacking identity claims could also gain higher ranks. We define a rank error and a certification quantity function to measure and compare each centrality algorithm's quality.

The *rank_error* function in Equation 3.5 calculates the rank error, where U is the set of attacking identity claims, $m = |U|$ is the number of the attacking identity claims, $n = |V|$ is the total number of identity claims, and $rank(v)$ is

the function that returns the rank of each identity claim according to its reputation. The best rank is 0, and the least is $n-1$. The *rank_error* results in a value between 0 and 1 where 1 is returned when the attackers occupy the m top ranks and 0 when they occupy the m lowest ranks.

$$rank\_error(U, m, n) = \frac{1}{m(n-m)} \sum_{u \in U} (n - rank(u)) - \frac{1}{2}m(m+1) \tag{3.5}$$

The $certification\_quantity$ function in Equation 3.6 measures how many identity claims are certified by one domain-validated identity claim. U is the set of attacking identity claims, W is the set of domain-validated identity claims, and $rank(v)$ is the function that returns the rank of each identity claim according to its reputation. Note that the criteria for the certification might differ by application or user requirements. In our experiments, we evaluate all identity claims with a better reputation than the best-ranked attacker as certified. Other certification strategies might be the worst-ranked domain-validated identity claim or a reputation threshold.

$$certification\_quantity(U, W) = \frac{1}{|W|} min\ rank(u) \tag{3.6}$$

We conducted two experiments with the python networkx [135] library to test the reputation functions' quality. We generated graphs with the powerlaw_cluster_graph generator of networkx. For each measurement, we have generated 50 random graphs and show the average of the results. In the first experiment, we vary the number of attacking identity claims while the number of domain-validated identity claims is fixed to 10 and the normal identity claims to 100. The number of erroneous trust links is $0.1 \cdot m$.

Figure 3.17 shows the rank error and the certification quantity for the different attacker cluster sizes. These results show a low and stable rank error with a high certification quantity for the PageRank algorithm. The number of attacking identity claims does not show a relevant impact on the quality of PageRank. EigenTrust and Eigenvector do not perform as well as the PageRank algorithm. However, up to one-third of the identity claims as attackers, these

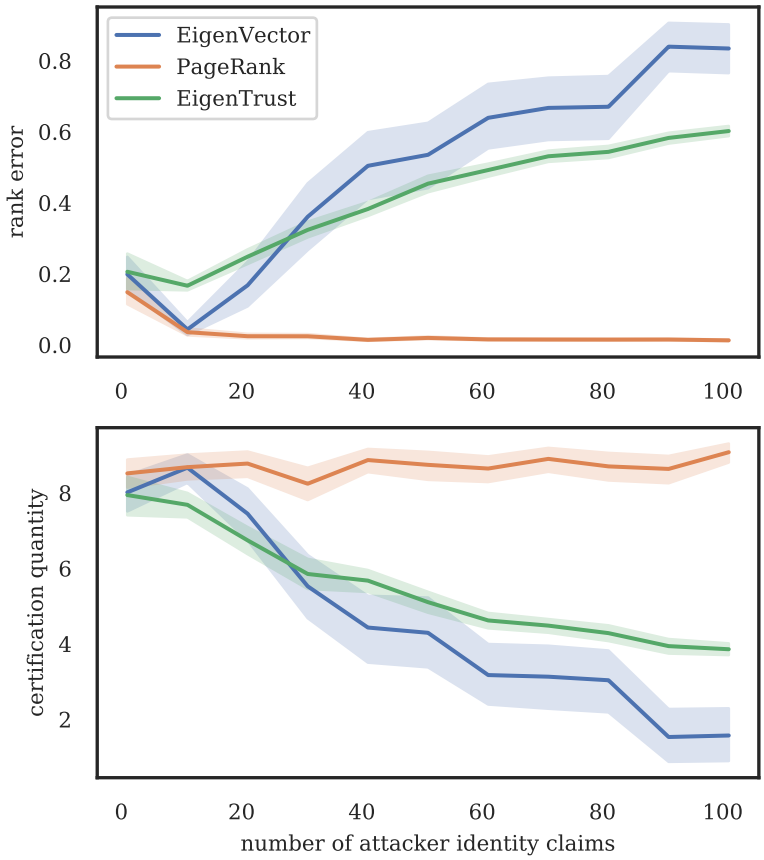


FIGURE 3.17: Quality of reputation in relation to links and attacker cluster size [114]

algorithms can still certify around 50% of the identity claims.

In the second experiment, we vary the number of erroneous trust links from the honest cluster towards the attackers while the number of attackers is fixed to 10, the number of domain-validated identity claims are set to 10, and the normal identity claims are 100. Figure 3.18 shows the result of this second experiment. Here, PageRank performs a bit worse than EigenTrust and Eigenvector Centrality, and they perform almost the same for the certification quantity. With 20 erroneous trust links, each attacker has got, on average, two reputation votes from the honest cluster. The relatively low-rank error and the decreasing certification quantity show that most attackers are still in the lower half of the reputation ranking. This shows that the cluster with the domain-validated identity claims remains more influential even if many members of the honest cluster have erroneously created trust links to attackers. Note that a

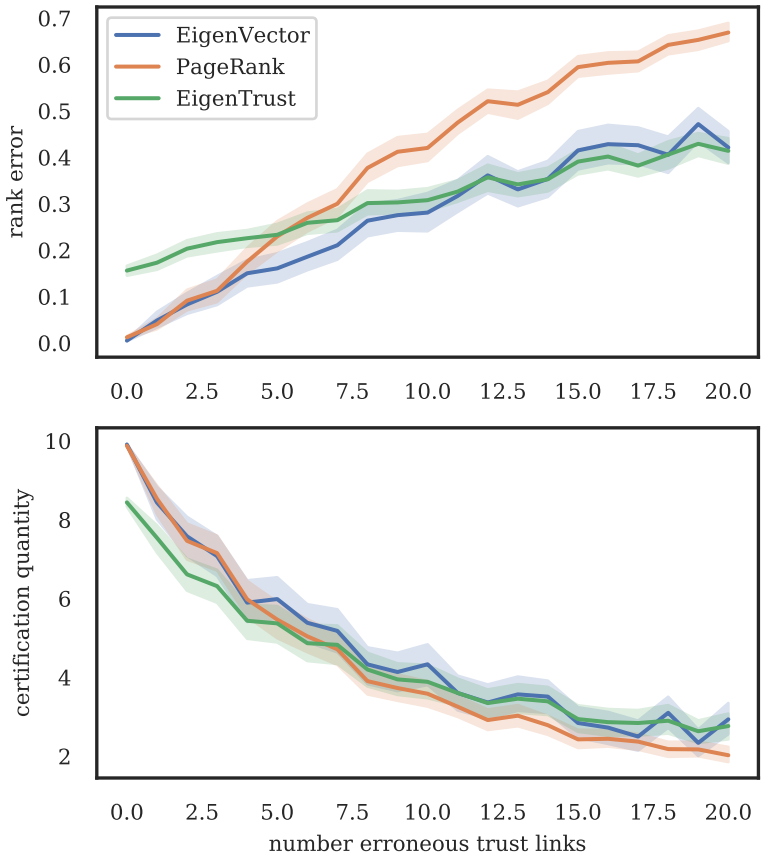


FIGURE 3.18: Quality of reputation in relation to number erroneous trust links [114]

creator of an erroneous trust link can always revoke this again, and, therefore, it should not happen that an unauthenticated attacking cluster gains so many trust votes. At some point, when a cluster has a lot of incoming trust links, it might also just be trusted.

The results show that all evaluated reputation algorithms perform pretty well in preventing a malicious collective from gaining a reputation. Therefore, reputation can be used to certify well-reputed identity claims. Nevertheless, the certification threshold must be defined by the user and depends on the application. Depending on the certification threshold, some legitimate identity claims will not have enough reputation to be certified. However, if they are closely connected over trust declarations with a certified and well-reputed identity claim, they might be certified over this certification chain.

Manipulation of Block Discovery Service

In order to claim that a signature was created at another time, an attacker might try to manipulate the block discovery times created by the block discovery services. However, when the median is used to calculate the block discovery time, the attacker needs to control more than 50% of the n most reputed discovery services. Since it is likely that the central nodes with the highest reputation also operate timestamp services, the attacker would require to compete for reputation against half of the most reputed nodes in the network.

Split-World Attack

A split-world attack is an attack where an adversary tries to show only a fraction of the GoT to a node. For example, an adversary could block a revocation transaction from reaching a node to abuse a compromised key. In order to do so, the adversary would require to create malicious nodes that do not forward the block with the revocation transaction and ensure that the attacked node only has malicious nodes as peers. Nevertheless, the attacked node can continuously update its peers, and since some honest nodes exist in the network, it eventually will synchronize with a honest node and receive the block.

Manipulation of the Graph of Trust

An adversary might try to manipulate the GoT, e.g., to delete an immutably stored signature. While it is always possible for a private key holder to transparently revoke its signed transactions, it must not be possible to delete or change them. Each node builds the GoT out of the transactions stored in the ABCG. Therefore, an adversary would require to change a block in the ABCG to change or remove a transaction. Let us assume that an adversary has altered or removed a transaction contained in block B. Then the hash of the block changes. A block with a new hash is like a new block B_m, and the blocks that have confirmed this manipulated block would still refer to B. At least the nodes that have confirmed the original block would still have B in their replica of the ABCG, and other nodes can restore it from there. Consequently, it is impossible to manipulate or delete blocks from the ABCG.

3.5 Conclusion

In this chapter, we introduced the architecture of Veritaa, a DPKI with a signature store. The GoT is used to store and represent identity claims and signed declarations. These declarations are used to non-repudiably store signatures and as trust votes towards other identity claims. The ABCG immutably secures the transactions that build the GoT. The ABCG is an application-specific BlockDAG optimized for immutably storing and replicating graph transactions.

We showed how signed trust relations and domain-validation information can build a reputation system and how this reputation can be used to derive certification. Additionally, we showed how block discovery services are used to create distributed timestamps of blocks.

The evaluation of the proposed DPKI also revealed a few limitations. First, as an application specific DLT optimized to secure graph transactions, the ABCG is very efficient in securing the GoT. However, since it is relatively new, competing against well-known general-purpose DLTs adoption is challenging. Furthermore, some applications might require a total order of transactions. The introduction of sidechains can overcome these limitations. Since declarations can reference all hashes, even if the corresponding GoT element is not stored on the same DLT, the GoT has the advantageous property that it can be split into multiple sidechains. Some of these sidechains could be secured on a general purpose DLT that provides a total order of transactions and is already widely adopted. We will discuss sidechains in more detail in Chapter 5. Finally, the proposed ABCG optimizes the energy efficiency compared to general purpose DLTs but running a ABCG node on low-power IoT devices is not feasible. To enable IoT devices, we discuss in Chapter 4 how the ABCG can be extended to support IoT devices.

Chapter 4

Identity and Signature Management of IoT devices

4.1 Introduction

In chapter 3, we built a DLT-based DPKI with a signature store. A DLT-based DPKI might also be promising to manage the identities of IoT devices. However, the limited resources of IoT devices constrain the applicability of common DLTs, and lightweight solutions are required. This chapter discusses the optimization of the architecture introduced in the previous chapter to support IoT devices and addresses research question 2: *How to manage the identities and signatures of low-power IoT devices on a DLT-based DPKI with a signature store?*

To enable low-power IoT devices, we proposed Veritaa-IoT [136] [1]. To overcome the limitations of low-power IoT devices, we offload the resource-demanding DLT operations to an AFN. Therefore, we extend the ABCG with statements and the GoT with pairing transactions. Furthermore, we design and implement a Veritaa IoT client running on low-power IoT devices.

The main contributions presented in this chapter are as follows.

- Extend the architecture proposed in Chapter 3 with statements, AFNs, and pairing to support low-power IoT clients

[1] Partially reproduced in this chapter - © 2022 IFIP

- Design and implementation of IoT identities and the corresponding identity claims using off-the-shelf microcontrollers with hardware-accelerated cryptographic functions
- Design and implementation of a real-world testbed with custom low-power hardware to evaluate the system's performance
- Evaluation of energy consumption, overhead, and delay for making IoT devices auditable using the proposed architecture

4.2 System Model

The IoT comprises devices that exchange information. While some IoT devices have plenty of resources, others are restricted in energy, computational capabilities, bandwidth, and memory. Therefore, a DPKI for the IoT must be able to manage the spectrum of all these IoT devices.

4.2.1 Identity of IoT Devices

When information is exchanged between IoT devices, it is essential that the receiver of the information can authenticate the source of information.

In order to authenticate IoT devices, each IoT device requires a unique identity to be distinguished from others. Merriam Webster defines identity as the condition of being the same with something described or asserted [137]. Consequently, identity consists of a set of properties that differentiate the distinct entities. If the identifying set of properties has the same values, they identify the same entity. Additionally, to prevent identity theft, the identifying properties must be hard to forge, which means it must only be possible for the distinct entity to obtain its identity. The required difficulty to forge an identity is often bound to the security requirement of an application. For example, today, in many applications, the identity of people is bound to their name, phone number, and address. Applications requiring higher security might use biometric information like fingerprints, retina, or the face as identifying properties.

In contrast to people, IoT devices do not have biometric information that can be used as identifying parameters. While many diverse types of IoT devices

exist, many exact copies of each type exist. Therefore, an IoT device does not necessarily possess distinguishable properties, and the identifying properties must be set at manufacturing or later. Note that serial numbers and similar identifiers often fail the hard-to-forge requirement. Nevertheless, IoT devices have the advantage that they can use cryptographic functions to build their identity. For example, a private key can be used as identifying property. The advantage of the private key as identifying property is that a digital signature can be used to verify the identity of an IoT device. The digital signature can be verified with the public key. However, before the signature can be validated, the public key must be authenticated. The authenticating entity must verify that the public key belongs to the entity it claims. Since the IoT is growing quickly, it is impossible to authenticate the public keys manually, and PKIs are required. As discussed in Chapter 3 the GoT can be used as DPKI. The following section discusses how this DPKI can be extended for the IoT.

4.2.2 Graph of Trust and the Internet of Things

In Veritaa, all information is represented on the GoT. The GoT comprises identity claims, data hashes, and their relations. The advantage of the GoT is that it inherently represents the structure of the relations between entities and data. With this information, it is possible to certify the authenticity of identity claims. In Chapter 3, we showed how trust declarations, together with domain vetting and scoring, can be used to calculate the reputation of identity claims. Finally, a reputation threshold can be used to certify the identity claims. While this WoT like approach suits well for organizations that interact with each other and possess distinguished domains, it falls short for IoT devices. IoT devices are typically operated by their owner and therefore have little interactions with other entities that could authenticate their identity. Therefore, the identity claims of IoT devices would receive little trust votes, and in this case, another approach is required. Note, to authenticate an IoT device, a distinct public key must be verified to belong to this specific physical device. In Section 4.3.3, we show how the IoT device can be paired. This pairing process also comprises the authentication of the device.

To certify the authenticity of IoT devices' identity claims, the real-world relationships of these devices must be considered. Figure 4.1 shows the typical

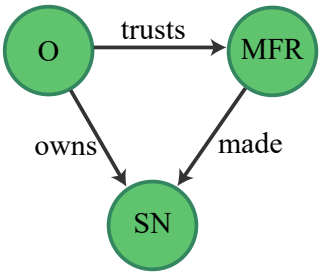


FIGURE 4.1: Real-world relations of an IoT device

real-world relations of an IoT device. The device *SN* has an owner *O* and a manufacturer *MFR*. The owner is responsible for the authentication, placement, and maintenance of the device, and the manufacturer is responsible for the hardware and the device's capabilities. Therefore, the authentication and reliability of a device depend on the owner and manufacturer.

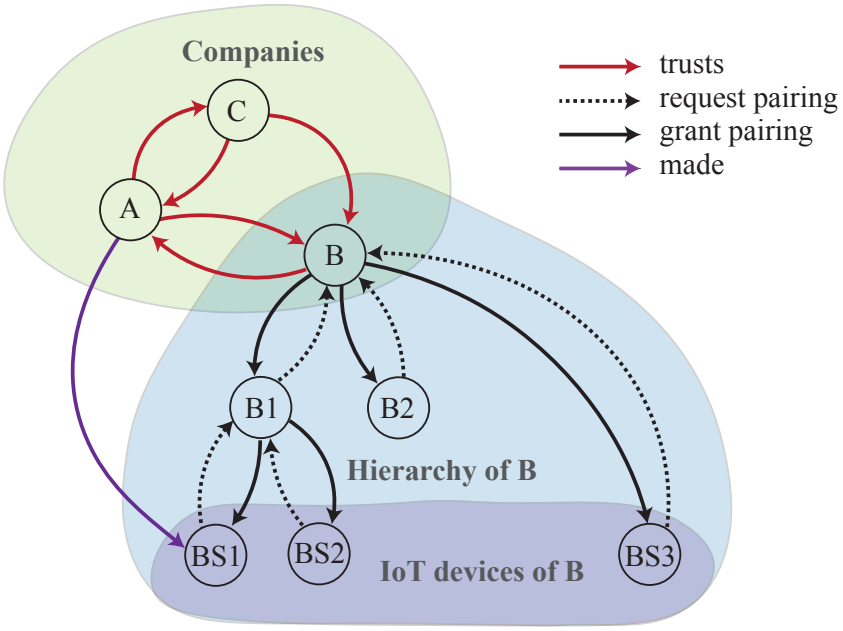


FIGURE 4.2: Example GoT of a combined model for the PKI [136]
© 2022 IFIP

The GoT has the advantage that we can use a combined approach that uses the benefits of a WoT based certification model and a CA like structure to manage the IoT identity claims of devices. Companies can use trust declarations and reputation to certify the authenticity of each other's identity claims, as proposed in Chapter 3. In this chapter, we propose to use a CA like structure to certify the authenticity of the identity claims of a company's internal hierarchy

and its IoT devices. The identity claim of the company is the root of this CA like structure, and each company can build its certificate tree. This combination enables the owners of IoT devices to certify the authenticity of the identity claims from their own IoT devices and departments. Furthermore, an auditor can decide based on the model presented in Chapter 3 if it trusts the identity claim of a company and its subordinate entities. Figure 4.2 shows an example GoT where this combined model is used. First, the companies A, B, and C use trust declarations to indicate that they have authenticated each other's identity claims. These trust votes are used to calculate reputation and derive the certification of the identity claims of A, B, and C. Second, company B used request pairing and grant pairing declarations to build a CA like certificate tree that certifies the departments of the company $B1$, $B2$, and its IoT devices $BS1$, $BS2$, and $BS3$. In this example, an auditor that needs to verify the authenticity of $BS1$'s identity claim follows the pairing declarations until it finds the identity claim of B. Additionally, the manufacturers can sign a declaration between company A and $BS1$. The auditor can use this information to reason about the data's quality.

4.2.3 Signing Measurements

Sensor measurements can be secured with the GoT to assert their integrity, authenticity, and immutability. Therefore, the IoT device calculates the hash of the measurement sample and signs a declaration of type measurement towards this hash. In Figure 4.3 IoT device B has signed measured declarations towards the measurement hashes M_1 and M_2. To achieve unique hashes for each measurement, we calculate the measurement hash from the following string: $|identity\_fingerprint|timestamp|measurement|$.

A validator can calculate a measurement sample's hash and search this hash in the GoT to verify the integrity and authenticity of the sample. When the validator has found the hash on the GoT, it can follow the declarations to see if it is linked to a trusted identity claim.

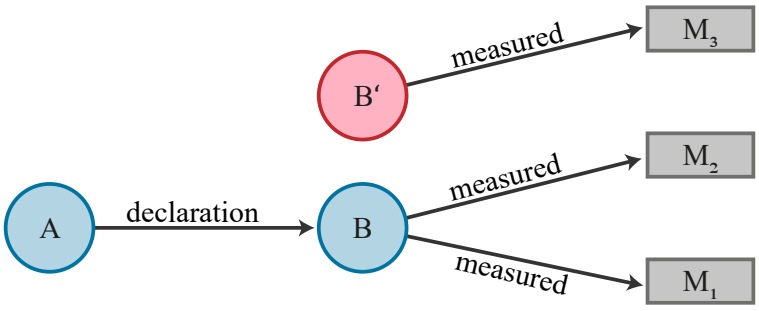


FIGURE 4.3: Signing measurements and manipulated identity claim B′

4.2.4 Tamper protection

With a fast-growing number of IoT devices connected through the IoT, eventually, some of those devices might fail. The reasons for failures are manifold. For example, a manufacturer could recognize a severe bug in the software and recall devices, or manipulation might result in the failure of a device.

While the revocation transactions, as introduced in Section 3.2.1, can be helpful to update the reputation based on external conditions, they might fall short in case of manipulation. For example, when an attacker manipulates an IoT device, this might not be recognized by the entities that have signed declarations towards its identity claim. Hence, the signed declarations might not be revoked, and trust remains. Tamper protection is required to overcome the issue of manipulated devices being trusted. This tamper protection must change the identity of the IoT device when an attacker manipulates the device. If the identity of the IoT devices changes on manipulation, the signed declarations still point to the old (not manipulated) identity and remain auditable. However, the new identity of the manipulated device starts over with no incoming trust declarations and, therefore, zero trust. Figure 4.3 shows an example where the manipulation of the IoT device B changed its identity to B'. Since it has no incoming declarations, B' is not trusted. However, the declarations of B, from before the manipulation, can still be audited.

To assure that an object's identity changes when the object has been manipulated, the identity must comprise at least one identifying property which always changes when an unauthorized change is made to the object. When an

identifying property changes, also the identity changes. In terms of IoT devices, the device's private key must change when the device is manipulated to prevent a manipulated device from creating signatures in the name of the original device. Furthermore, the identity must be secured, not accessible, and unclonable. In Section 4.3.2, we show how the tamper detection system is used to make an identity unusable in case of manipulation.

4.2.5 Associated Full Node

Devices connected to the IoT are diverse in terms of functionalities and available resources. Infrastructure for the IoT must be designed to support all different types of devices. Since some IoT devices have limited computational capabilities, available energy, and storage, a DPKI for the IoT requires considering these constraints. Running a full DLT node requires the device to receive, validate and store all blocks and transactions of the DLT. Also, the operation of limited light nodes requires the device to exchange data with the peers of the DLT network. This data exchange is not feasible for low-power devices connected over protocols like LoRaWAN with only a few down packets daily. In order to overcome this limitation, it is necessary to delegate the resource-demanding parts of the DLT to a full node and only manage the identities and signatures on the IoT device itself. We define a full node associated with IoT devices an Associated Full Node (AFN). The AFN is a full node that handles the computationally expensive and bandwidth-intensive DLT functions of its associated IoT devices. When the AFN is operated by the same entity that operates an IoT device, and the AFN and IoT device are paired, then the IoT device and the AFN can inherently trust each other. However, it must be asserted that information exchanged between the IoT device and the AFN cannot be dropped or manipulated. Since the AFN is paired with the IoT device, they have exchanged public keys and can verify each other's signatures. A valid signature asserts that a packet has not been manipulated. Acknowledgments (ACK) and Negative Acknowledgments (NACK) are used to assert that no information was dropped. The transactions reference the previous transaction to enable the NACK, as it is explained in the next section.

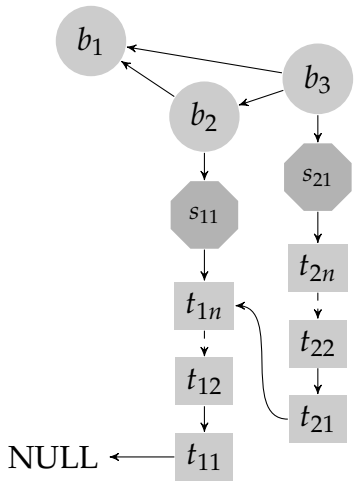


FIGURE 4.4: Hash tree extended with statements [136]

4.2.6 Statements

In order to enable power-constrained devices, we extend the ABCG with statements. A statement comprises a list of transactions, the id of the creator, and its signature. A statement's transactions are chained by referencing the hash of the previous transaction, and the creator signs the hash of the last transaction. Additionally, the IoT devices link the statements by referencing the last transaction of the previous statement with the first transaction of the new statement. Therefore, each IoT device creates its own chain of transactions. This chain of transactions asserts that statements cannot be dropped before they are secured on the DLT. If an AFN recognizes a missing referenced transaction, it sends a NACK to the corresponding IoT device, and the IoT device can resend the missing statement. Figure 4.4 shows an example hash tree of some blocks with two statements and their references. Each block contains a list of statements, and each statement is a signed list of transactions.

Compared to sending single transactions, the statement has the advantage that an IoT device does not have to sign each transaction individually. Consequently, the IoT device does not need to transmit the signature with each transaction, and, therefore, the energy required for this transmission can be saved. Furthermore, in contrast to committing complete blocks to a DLT, a statement has less overhead, and the AFN can perform the PoW required to protect the DLT from spam. Therefore, the statements enable offloading the

DLT-overhead to the AFNs.

4.2.7 Time Considerations

Section 3.2.3 discussed block discovery services for building timestamp consensus on a BlockDAG. These timestamps indicate the time when the block was detected on the DLT but not the time when an event observation, documented and secured with a declaration, was made. When the required temporal accuracy is lower than the temporal accuracy provided by the block rate and delay to commit a block to the DLT then a block discovery timestamp can be used as an event timestamp. Nevertheless, some IoT applications might require higher temporal accuracy. However, IoT applications can have long delays between the observation of an event and when the observation is secured on the DLT.

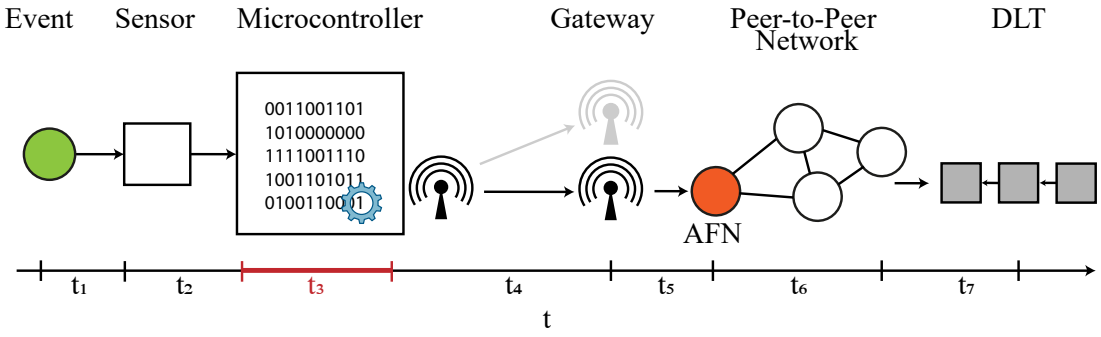


FIGURE 4.5: Timeline for securing an event observation on a DLT

Figure 4.5 shows the timeline from an event to its documentation on the DLT with a sensor directly connected to the AFN over a low-power IoT network like LoRaWAN or LTE Cat M1. Table 4.1 comprises the description of the different phases that must be passed through to secure a measurement on the DLT. Some of these phases depend on the technology used. For example, as shown in the evaluation Section 4.4.4, the time to secure an event observation as a signed declaration on the GoT depends significantly on the used network technology and can introduce long delays. These delays are not in the control of the DLT or cryptography. Consequently, the block discovery services' timestamps can only provide limited temporal accuracy for events observed by the IoT. Additionally, the firmware of the microcontroller could intentionally or

Time	Description
t_1	The time the physical signal requires to propagate from the events origin to the sensor. This time depends on the physical property (e.g., speed of light or sound) and the spatial distance.
t_2	The time to send the measurement from the sensor to the microcontroller.
t_3	Time to process the information on the microcontroller. This comprises the processing of raw data, the creation of a transaction, the creation of a signed statement, and copying as a packet to the output buffer.
t_4	The transmission of the packet from the radio module to a gateway. The time depends on the used protocol and bandwidth.
t_5	The time required to forward a packet from the gateway to the AFN of the IoT device
t_6	The time to process the packet on the AFN. This comprises appending the statement in a block and committing the block to the DLT.
t_7	The time to confirm the block on the DLT. This comprises the dissemination of the block over the peer-to-peer network of the DLT and waiting until another block has confirmed the block.

TABLE 4.1: Description of the steps required to secure a measurement value

unintentionally delay the transmission of the measurement at any length (increase t_3). Consequently, the remaining components of the overall delay are insignificant, and the microcontroller must provide an auditable timestamp. The clock synchronization must be secured and documented on the audit trail to make timestamps auditable.

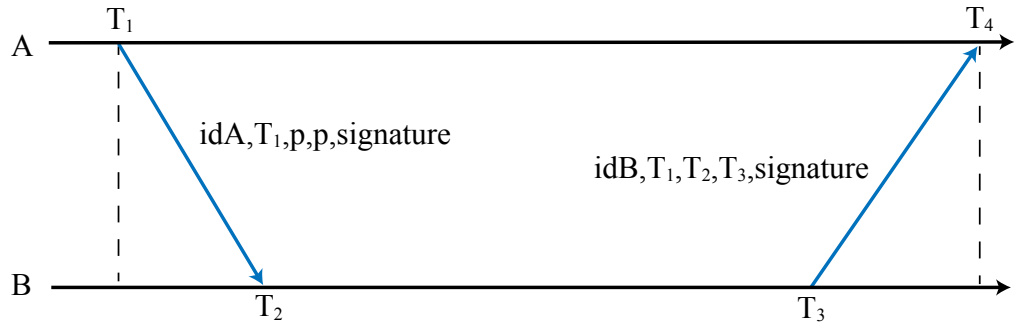


FIGURE 4.6: Secured time synchronization

Under the assumption of a symmetric propagation delay, the clock offset $\theta = \frac{1}{2}[(T_2 - T_1) + (T_3 - T_4)]$, and round trip delay $\delta = (T_4 - T_1) - (T_3 - T_2)$ can be

measured between the two clocks using standard methods [138]. These values can then be used to synchronize two clocks. Figure 4.6 shows an example where device A measures δ and its θ to B. For the measurement, device A sends a packet with the transmission timestamp T_1 to B. B saves the timestamp T_2 and sends back a packet with T_1, T_2, and T_3, the timestamp when it transmitted the replay packet. The request packet is padded to the same length as the response packet to ensure similar airtime. When A has received the response from B, it has all information required to calculate θ and δ. A can adjust its clock by adding θ to synchronize with B. To improve accuracy, A can measure θ multiple times and add the average for the synchronization.

To ensure that the clock offset and round trip delay measurements cannot be manipulated, we extended the request and response packet with the identity hashes idA and idB of the devices that sent the packets and a signature of the identity hash and the timestamps in the packet. With this information, both devices involved in the synchronization can verify the authenticity of the packets using the GoT.

The synchronization is represented as request time synchronization and grant time synchronization declarations on the GoT. These declarations make synchronization information accessible to auditors and make the synchronization auditable. To request a synchronization, the IoT device measures the clock offset θ_{AB} to a time server, updates its clock accordingly, and creates a request time synchronization declaration. When the time server receives a synchronization request, it also measures the clock offset to the IoT device and creates a grant time synchronization declaration. In the grant time synchronization declaration, the time server also adds the θ_{BA} it has measured. Consequently, the auditor can audit how accurately the IoT device has synchronized with the time server. The time server is a Veritaa full node with an accurate clock, for example, synchronized over the Network Time Protocol (NTP). Note that this representation of the synchronization on the GoT can also be used for other standard synchronization algorithms. Figure 4.7 shows an example request time synchronization and grant time synchronization declaration between an IoT device A and a time server B.

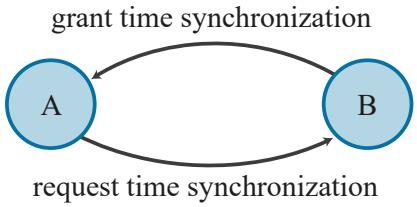


FIGURE 4.7: Time synchronization declarations

Multiple sensors that witness the same event could also be used to build consensus about the time of the event's occurrence. Similar to the block discovery services that build consensus about the block timestamp. Let S_o be all sensors that have observed a certain event and t_s the time when sensor $s \in S_o$ has observed the event. Then we can define the consensus timestamp $t_c = mean(t_s) \forall s \in S_o$. Nevertheless, using multiple sensors as witnesses of the same event has the limitation that multiple sensors have to be in the sensing range of the event, which requires a high density of sensors.

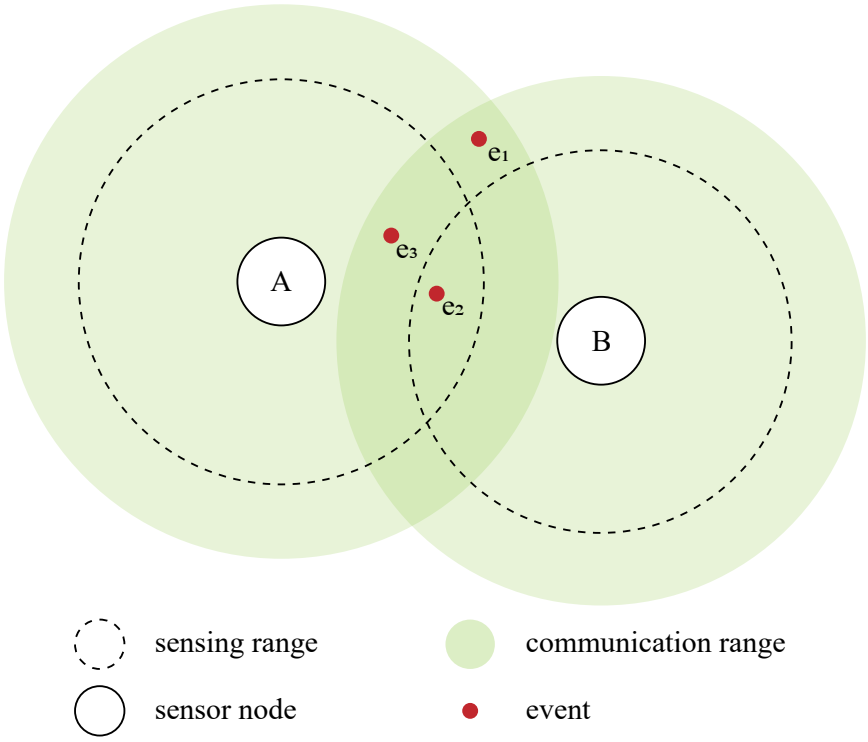


FIGURE 4.8: Spatial resolution in sensor networks

Figure 4.8 shows the spatial resolution of two sensors. Typically, a sensor has a communication and a sensing range. In reality, these ranges are not circular

and do not have the same reach. The sensing range depends on the used sensor and the measured physical parameter. In this example, event e_1 is neither in the sensing range of A nor B. Therefore, e_1 is not observed by the network. The event e_2 is in the range of A and B. The event e_3 is in the sensing range of B and, therefore, measured by B but not A. Two sensors within communication range could perform a PoL, discussed in more detail in Chapter 6. However, since the communication range is not congruent with the sensing range, the PoL does not provide enough information to decide if two sensors have observed the same event. Consequently, when consensus about time does not exist yet, grouping observations of multiple sensors by events is challenging and raises interesting research questions not covered in this thesis.

4.3 Implementation

4.3.1 Testbed

To evaluate our proposed architecture and future algorithms and applications, we extended the Veritaa testbed. Figure 4.9 shows the architecture of our testbed that extends Veritaa with low-power IoT devices. The testbed includes the Veritaa network, application servers, and IoT devices. The Veritaa network comprises several full nodes and some AFNs capable of handling IoT devices. In this testbed, all nodes of the Veritaa network are running on a Kubernetes cluster.

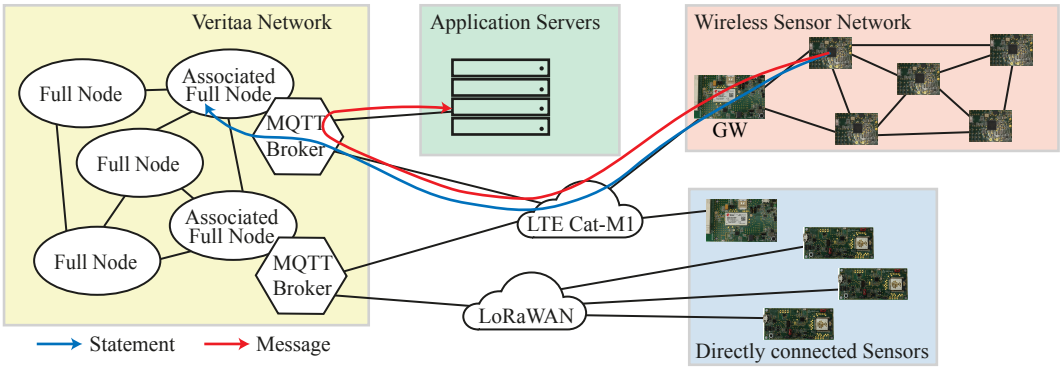


FIGURE 4.9: IoT testbed and architecture overview [136]

The AFNs run Message Queuing Telemetry Transport (MQTT) brokers to receive statements from their associated IoT devices and to exchange control data with these devices. Veritaa assures the integrity and authenticity of messages by immutably storing their hashes and the creator's signature on a DLT. However, the messages are not stored on the DLT. While the message contains data like sensor values, the signed statements ensure the message's authenticity, integrity, and immutability. Therefore, we distinguish between the signed statements directed to an AFN and the message directed to an application server. The message (red) and the statement (blue) are exchanged over the MQTT broker of the AFN. The AFN and the application server subscribe to related topics to receive published statements and messages from their associated IoT devices.

The architecture is designed to be flexible and support all kinds of IoT devices and IoT infrastructure. In this testbed, the IoT devices are connected over LTE Cat M1 and LoRaWAN to the MQTT broker of their AFN. Furthermore, the IoT devices using LTE Cat M1 can act as Gateway (GW) and translate the traffic to different low-power protocols like MQTT-SN [139], SDNWise [6], or RPL [5]. These GWs enable the integration of low-power WSN nodes without a direct Internet connection.

We developed a LTE Cat M1 gateway, a LoRaWAN-based sensor, and a WSN sensor as low-power IoT devices for our testbed and designed and implemented the Veritaa IoT client to connect them to their AFN. In this testbed, all IoT devices are based on the EFR32MG21 microcontroller and custom circuit boards. The EFR32MG21 is a system on a chip with up to 96k RAM, up to 1 MB Flash memory, a multi-protocol radio module, and a security processor that enables the SecureVault technology of Silicon Labs [140]. To connect the IoT devices with the AFNs and application servers, we use the u-Blox SARA-R510 radio module for LTE Cat-M1 and the Microchip WLR089U0 radio module for LoRaWAN. For the communication between the sensor nodes of a WSN, the EFR32MG provides a multi-protocol 2.4GHz module that supports 802.15.4 and Bluetooth Low Energy.

4.3.2 Identity of Microcontrollers

Veritaa, as a DPKI, manages the distributed certification of identity claims' authenticity. The security of a private key is not in the scope of the Veritaa framework but in the scope of the applications that use the Veritaa framework to publish, authenticate, and certify identity claims. In this chapter, we use the EFR32MG21 microcontrollers that contain SecureVault, a dedicated security CPU that isolates cryptographic functions and data from the host processor core [141].

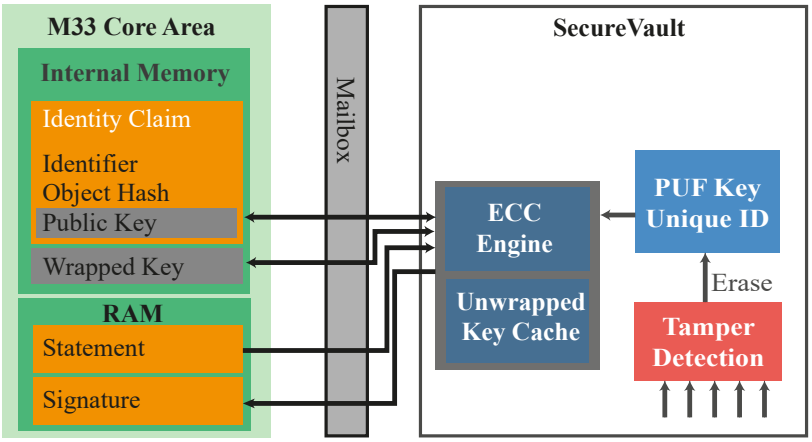


FIGURE 4.10: Identity claim using Silicon Labs SecureVault [136]

Figure 4.10 shows how we use Silicon Labs SecureVault to build the identity claim of an IoT device. All cryptographic functions are executed on the dedicated security CPU of SecureVault. The elliptic curve cryptography key pair is generated in the SecureVault, and only an encrypted wrapped key is stored in the internal flash memory of the microcontroller. Therefore, it is not possible to access the private key from the IoT device's firmware. However, the public key can be accessed by loading the wrapped key into SecureVault and exporting the public key. This public key can then be used to build the identity claim of the IoT device. The wrapped key is loaded from the flash memory into SecureVault over the mailbox when the private key is needed. Messages that must be signed are sent over the mailbox to the elliptic curve cryptography engine, and the signature is then returned over the mailbox.

The tamper detection system of SecureVault monitors several system parameters. When the tamper detection system detects a tamper attempt and a certain

threshold is exceeded, the tamper detection erases the Physical Unclonable Function (PUF) key. When the PUF key is erased, the wrapped keys cannot be encrypted anymore, and, therefore, the private keys are not usable anymore. Consequently, SecureVault's tamper detection can be used to prevent manipulated IoT devices from signing further transactions. SecureVault also provides Secure Boot with Root of Trust and Secure Loader to ensure that the firmware is not exchanged with malicious code.

Note that SecureVault is one way to implement identity claim management on an IoT device. The key pair could also be created using the arm TrustZone, and the trusted execution environment [142] or just plain keys. When the transactions are valid and the statements correctly signed, third-party auditors can't decide how the private keys for the signature are managed. However, the manufacturer can sign a "made" declaration towards each IoT device it has made. Therefore, all who validate a signed message can see which manufacturer has created a device and use the manufacturer's reputation to reason about the quality of its security implementation. A responsible manufacturer can also transparently sign a declaration towards devices with specific security vulnerabilities. This transparency provides information that can be used to decide if a specific IoT device should be trusted or not.

4.3.3 Pairing

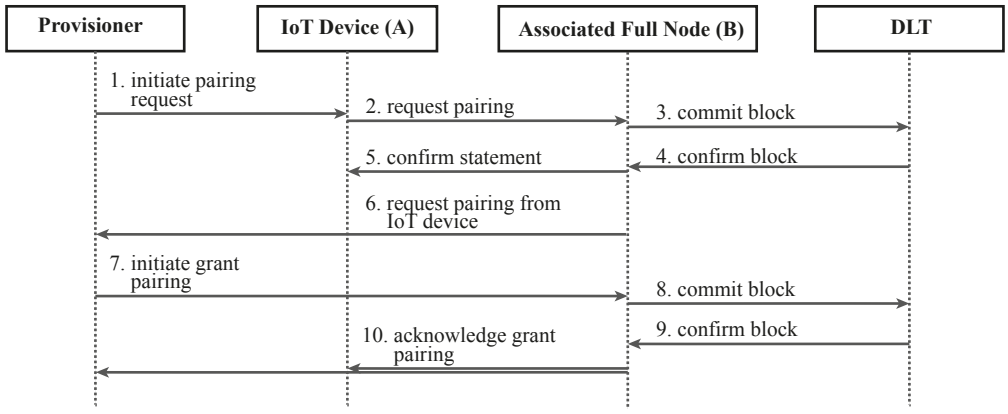


FIGURE 4.11: The sequence of pairing an IoT device with its AFN [136] © 2022 IFIP

Figure 4.11 shows a sequence diagram of the pairing process of an IoT device and its AFN. The pairing is conducted by a provisioner, an entity that is able to

initiate signatures of the IoT client and an identity claim managed on the IoT client's AFN. Typically, the provisioner is the owner of the IoT device or the operator of the IoT device's infrastructure. The provisioner can use a smartphone application to pair an IoT device with its AFN. The following actions are performed to pair the IoT device (A) with its AFN (B):

1. The provisioner forwards the fingerprint and public key of B's identity claim to A and initiates the pairing request.
2. A creates a request pairing declaration pointing towards the fingerprint of B provided by the provisioner, signs it in a statement, and forwards it to B.
3. B wraps the statement in block b_1 and commits b_1 to the DLT.
4. When B receives valid blocks from its DLT peers that reference b_1, then the block is confirmed.
5. B confirms to A that the statement was successfully written to the DLT.
6. B reports to the provisioner that it has an open incoming request pairing declaration.
7. The provisioner validates that A signed the open pairing request and, if so, forwards A's fingerprint and public key to B and initiates the grant pairing.
8. B creates a grant pairing declaration towards A's identity claim, wraps it as a statement in block b_2, and commits b_2 to the DLT.
9. When B receives valid blocks from its DLT peers that confirm b_2, then the block is confirmed.
10. B sends acknowledgments to A and the provisioner to notify them that the pairing was successful.

After pairing, the IoT device and the AFN have exchanged their public keys and, therefore, can validate each other's signatures and exchange encrypted messages. Furthermore, the pairing is declared on the GoT, and all Veritaa

members can see that the IoT device is paired with the AFN and, therefore, reason about the authenticity data signed by the IoT device.

4.4 Evaluation

4.4.1 Security Analysis

Attackers might try to manipulate, repudiate, or forge measurement values. In this security analysis, we assume that the used cryptographic functions (hash and signature) and the microcontroller's tamper detection system are secure. Furthermore, we assume that the DLT is secure, and therefore, all blocks committed to the network are immutably stored.

An adversary might try to manipulate, repudiate, or forge sensor values on the application servers after they have been measured. Suppose an attacker has manipulated or forged a sensor value on an application server. Then a validator can calculate the hash of the data and check on the GoT if the alleged IoT device has signed the data. If a signed declaration exists on the GoT, then data authenticity and integrity are proven.

An adversary that operates an application server might try to delete some data of an IoT device. Since the DLT only secures the data hashes, it can only prove that a data point existed at a given time, but the DLT cannot recreate it. However, if entities that do not trust each other collaborate on an application, all can store the data in their own infrastructure, and in case of a dispute, the DLT can be used to resolve the conflict. Assume an entity deletes a data point of its copy of the data set and claims that the data point never existed. Then, all members of the corresponding Veritaa network can see the signed declaration towards a hash with no related data point. With this information, they can prove that the data point was deleted. Furthermore, let us assume that another entity also has a copy of the data set. Therefore, it is able to build the hashes of all data points and select the data point that corresponds with the disputed signed declaration on the GoT. Since the hash function is secure, two data points with the same hash are equal. Consequently, the solution is able to detect missing data, and the original data can be restored if non-trusting parties make their own copy of the data.

An adversary might physically manipulate the IoT device and abuse its identity to forge signatures. Let us assume that the attacker has manipulated the hardware or firmware of the sensor node. Then the tamper detection of the microcontroller will detect the manipulation. As discussed in Section 4.3.2, the detection of this manipulation erases the PUF key, and therefore, the identity of the IoT device is destroyed. Consequently, it is impossible to manipulate an IoT device and use its identity to sign forged declarations.

An adversary might try to manipulate a statement before it is written to the DLT. Let us assume that an adversary has inserted, removed, or manipulated a transaction in a statement. Then the hash of the transaction changes, and since the following transaction references the hash of the manipulated transaction, the hash of the following transactions would also change. Therefore, also the hash of the last transaction changes. Therefore, the signature of the statement, which signs the hash of the last transaction, is invalid after the manipulation. The DLT only accepts blocks that only contain valid statements. Consequently, the statement cannot be manipulated. Furthermore, assume an adversary dropped a statement before it was written to the DLT. Then the first transaction of the following statement would reference the transaction hash of the last transaction of the dropped statement. The AFN would recognize that the first transaction is referencing a non-existing transaction and can send a NACK to initiate retransmission. Consequently, it is not possible to drop a statement unrecognized.

4.4.2 Key Size

Depending on the security requirement of the Veritaa network, different elliptic curves with different key sizes can be used. In this experiment, we evaluated the performance of the distinct Veritaa functions using different key sizes. Figure 4.12 shows the performance of the different Veritaa functions using hardware acceleration. Since the different operations mainly depend on the underlying cryptographic functions, the time needed depends on the key size. Verification is slightly slower than signing.

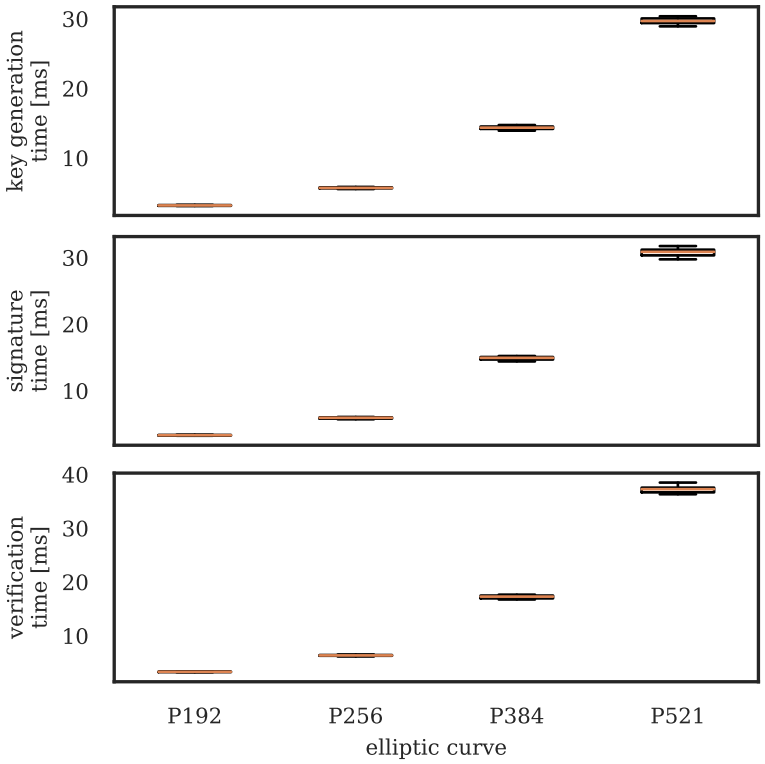


FIGURE 4.12: Performance of different elliptic curves

4.4.3 Hardware Accelerated Cryptography

The proposed architecture supports the public key management of various IoT devices. These devices come with distinguished capabilities. Some device types might provide hardware-accelerated cryptography, while others might not. Our testbed mainly uses the EFR32MG21 from Silicon Labs, which supports hardware-accelerated cryptography. However, to evaluate if it is also possible to integrate sensors without hardware-accelerated cryptography into Veritaa, we have implemented the Veritaa IoT client for Arduino Mega 2560. To implement the Arduino Veritaa IoT client, we use the Arduino Cryptography Library [143] for SHA256 and the micro-ecc library [144] for the elliptic curve cryptography. This experiment evaluated the time required to perform the different cryptographic functions on a sensor with and a sensor without hardware acceleration for cryptographic functions. For this experiment, we used the NIST P256 elliptic curve.

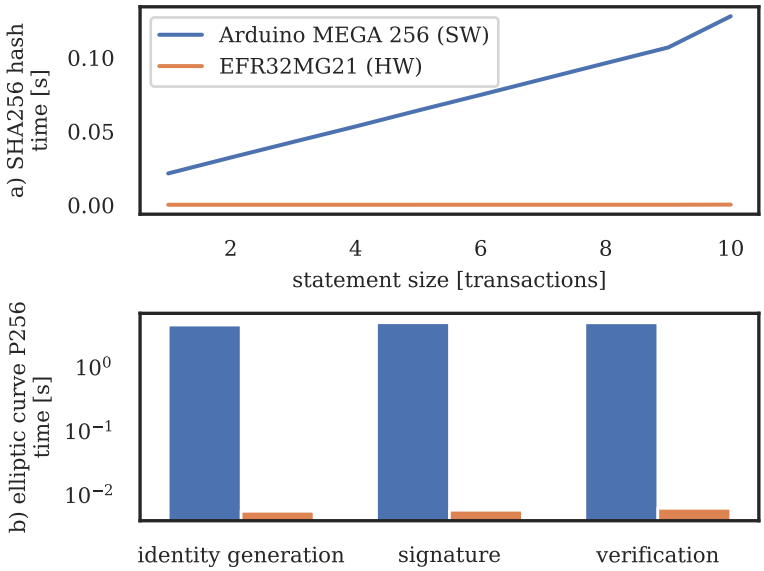


FIGURE 4.13: Performance with and without hardware acceleration

Figure 4.13a shows the time required to build the hashes for statements containing different numbers of transactions. Figure 4.13b shows the time required to generate an identity claim, sign a statement, and verify a statement. Note that most of the required time is used for the related cryptographic functions. This experiment shows that it is possible to run a Veritaa IoT client on sensor nodes with and without hardware-accelerated cryptographic functions. However, hardware acceleration has a significant impact on the application's performance. This improved performance and the fact that these cryptographic features become more and more common in today's microcontrollers might boost the acceptance of DLT applications in the IoT.

4.4.4 Transmission Time to Distributed Ledger Technology

Between an event that initiates a new transaction and the time the transaction is written immutably on a distributed ledger, time elapses. In IoT applications, this time depends on the underlying network architecture and technology.

This experiment evaluated the time until a statement's transactions have immutably been written from an IoT device to the DLT. In this experiment, we connected the IoT devices over LTE Cat-M1 and LoRaWAN. Note that the radio might longer be active than the time required to commit a transaction on

the DLT. This is because the radio remains active after the transmission and waits for ACKs.

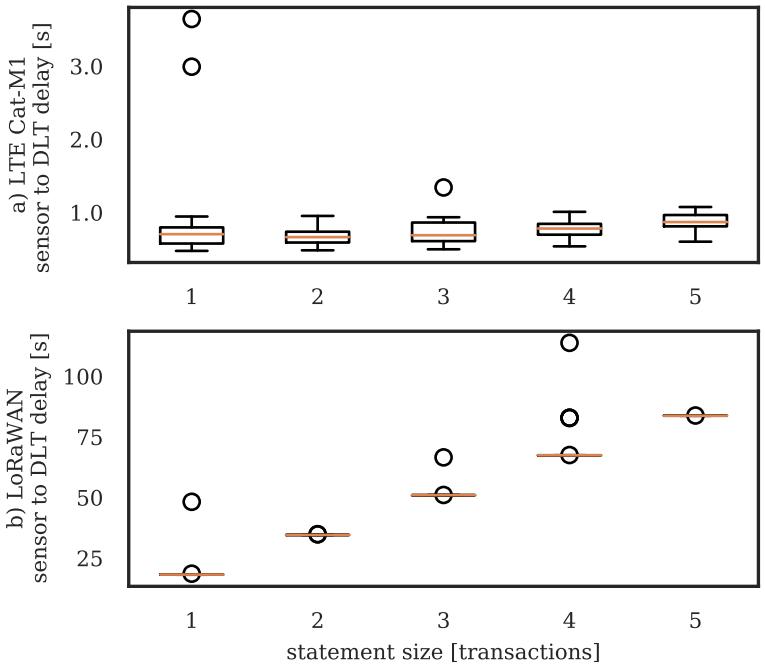


FIGURE 4.14: Required time to send a statement to the DLT. [136]

Figure 4.14a shows the time required to commit a new statement from the IoT devices to the DLT. The experiment shows that for LTE Cat-M1, times below 1 s are feasible. Note that these times include the whole process, especially the wake-up times of the microcontroller and the SARA-R510 radio module. The measurements show a few outliers that required more time. In low-power networks, protocols are designed to be resilient to packet loss. Therefore, lost packets are retransmitted. Furthermore, the radio channel is not always free, and the transmission is delayed by a backoff time. These retransmissions and backoff times can result in a few packets that require significantly more time for transmission than the average that is sent successfully on the first try. Therefore, a few outliers are expected.

Figure 4.14b shows the performance when the statements are transmitted over LoRaWAN. Compared to LTE Cat-M1, the LoRaWAN network has a lower bandwidth and a smaller MTU size to reach larger distances. The hashes of transactions and the signature of the statement require additional space that

might exceed the MTU when multiple transactions are sent. When the payload is larger than the MTU, the statement is split, and each transaction is sent separately to the AFN. When the last transaction has been sent, the signature for the complete statement is transmitted. The AFN validates the received transactions and signatures and assembles the parts as a statement. While the MTU of LTE Cat M1 is large enough to send 5 transactions, the MTU of LoRaWAN only enables 1. Consequently, more packets must be sent with LoRaWAN. Additionally, after transmitting a packet, the sensor node must remain silent to meet the maximum duty cycle requirement. With the given LoRa setup, it is possible to send up to approximately 200 uplink messages per hour (depending on the payload size) [145]. Note that the fair access policy of some LoRaWAN networks might even have lower limits. This experiment shows that it requires approximately 20 s to commit a transaction to the DLT. The results include the sleep time to meet the maximum duty cycle between each packet. The LoRaWAN experiment also reveals a few outliers for the same reason as the LTE Cat M1 experiment. Since all packets have the same size in this experiment and since LoRaWAN does not need to establish a connection, the successful LoRaWAN packets mostly require the same time for the transmission. Consequently, the confidence interval of this experiment is smaller.

4.4.5 Energy Consumption

The statement comprises hashes, creator id, and creator signature to ensure the transmitted data's immutability, integrity, and authenticity. We have measured the current and energy consumption required to transmit statements and plain messages to evaluate the statement overhead.

Figure 4.15 shows the current consumption when the message is transmitted over the LoRaWAN using spreading factor 7 and 125 kHz bandwidth. With this configuration, the bit rate is 5.47 kb/s. In this experiment, we transmitted a 32 bytes plain message, which resulted in a 170 bytes signed statement. Due to the relatively low data rate of LoRa, the transmission of data consumes most of the energy.

Figure 4.16 shows the current consumption when the message is transmitted

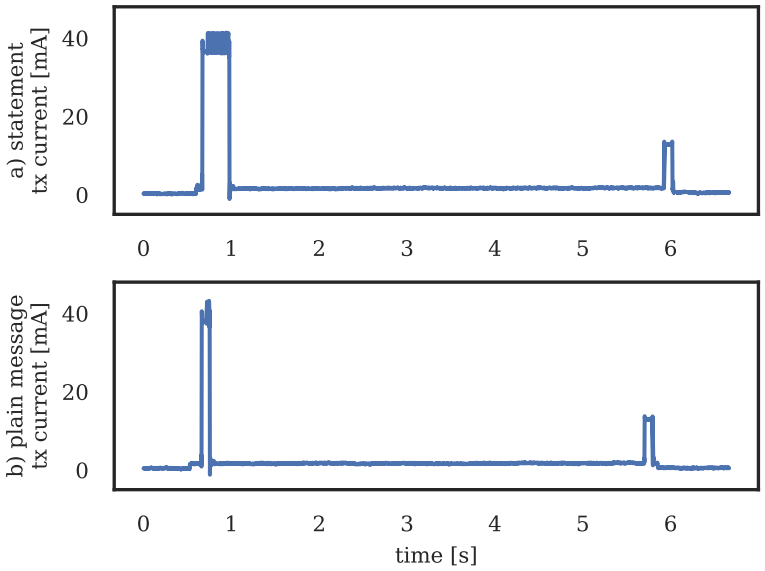


FIGURE 4.15: LoRaWAN tx current consumption [136]
© 2022 IFIP

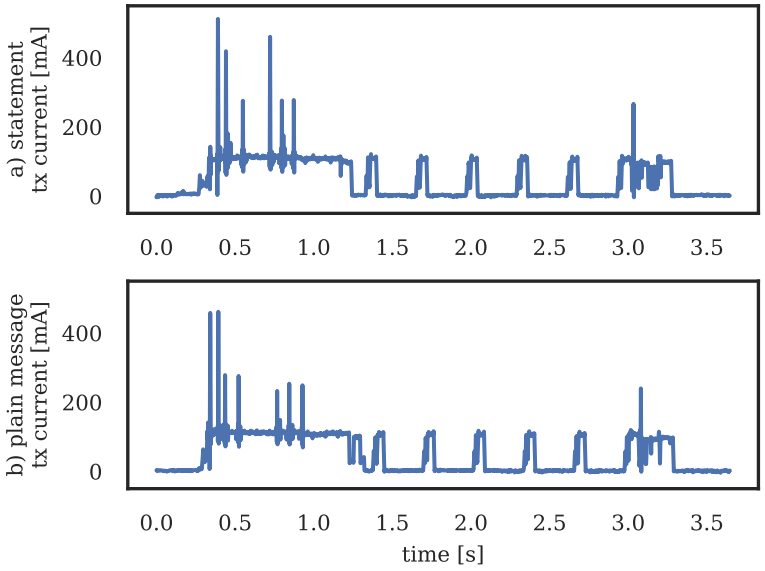


FIGURE 4.16: LTE Cat-M1 tx current consumption [136]
© 2022 IFIP

over LTE Cat-M1. LTE Cat-M1 has an uplink bit rate of up to 1.2 Mb/s. Therefore, the air time of the data is around 200 µs for the plain message and 1.1 ms for the statement. However, after the SARA-R510 radio module has transmitted a packet, it remains awake for around 3 s and listens to incoming messages and ACKs. Consequently, the size of the transmitted information does not

contribute much to the overall energy consumption.

We calculated the energy required to transmit a statement and plain message based on the current consumption. Figure 4.17 shows the energy required to transmit a statement and plain messages with LoRaWAN and LTE Cat-M1.

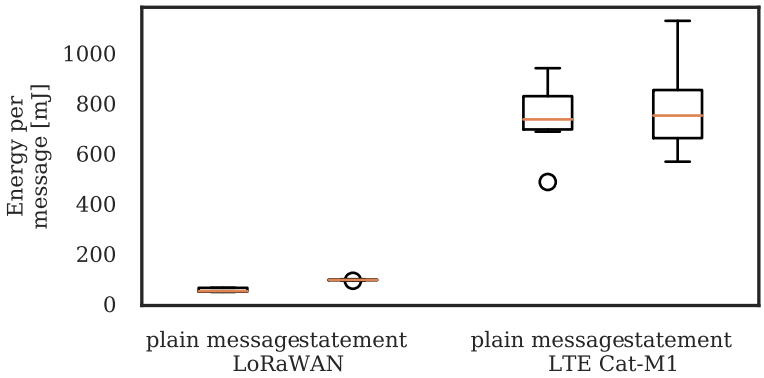


FIGURE 4.17: Required energy to transmit a message [136] © 2022 IFIP

$$overhead = \frac{c(statement) - c(message)}{c(message)} \cdot 100 \qquad [\%] \quad (4.1)$$

Securing the measurement data of IoT devices requires additional energy. We calculate the corresponding overhead to evaluate the impact of securing the data on energy consumption. Equation 4.1 shows how overhead is calculated. The overhead sets the direct costs in relation to the indirect costs, where $c(x)$ is a cost function.

	LoRaWAN		LTE Cat-M1	
	statement	plain message	statement	plain message
Data [bytes]	170	32	170	32
Overhead [%]	***431.25***		***431.25***	
Energy [mJ]	95.67 ±1.66	55.30 ±7.35	784.65 ±160.69	746.12 ±116.23
Overhead [%]	**73.00**		**5.16**	

TABLE 4.2: Signature Overhead [136] © 2022 IFIP

Table 4.2 shows the overhead introduced by securing the data with the proposed solution. Different cost functions $c(x)$ can be used to evaluate the overhead. The first cost function shown in this table uses the required amount of

bytes. Securing the data requires space for signatures and hashes, which introduces 431.25% overhead. However, this metric does not consider that the IoT devices must establish a connection before they can send data and that the energy required depends on the communication technology used. Therefore, the energy required to send a packet, including communication overhead, is a more meaningful metric. The experiment shows that securing measurements with our proposed solution requires 73% more energy with LoRaWAN and 5.16% more energy with LTE Cat M-1. Since LoRaWAN is more efficient in establishing and maintaining a connection than LTE Cat M1, the overhead introduced by securing the data is higher for LoRaWAN than for LTE Cat M1.

Nevertheless, this overhead is affordable for applications that require authenticity, integrity, and immutability of sensor data. If the sensor depends on a power supply like a solar cell, the solar cell must be dimensioned accordingly, and if the sensor depends only on batteries, they need to be replaced more frequently. Note that the quality-relevant sensors that benefit from this additional security also require frequent maintenance like calibrations.

4.5 Conclusion

In this chapter, we extended the Veritaa framework to support low-power IoT devices. Therefore, we extended the GoT with new pairing capabilities. The ABCG we extended with statements to offload DLT functions to an AFN and enable low-power IoT devices. Furthermore, we built a real-world testbed to evaluate these extensions. The testbed comprises custom-made low-power IoT devices using LoRaWAN and LTE Cat-M1 for communication.

As a proof-of-concept, we used the testbed to perform several experiments. We evaluated the delay until messages are secured on the DLT and the overhead introduced by securing the messages on the DLT. Our evaluation shows that securing the information introduces some overhead. However, there is always a tradeoff between energy consumption and security in low-power applications. The additional energy is affordable for many applications requiring this level of security. To evaluate the capabilities of low-power microcontrollers to calculate the required cryptographic functions, we measured the execution time of these functions. The evaluation showed that the hardware acceleration

of recent microcontrollers enables all required functions and makes the system feasible. Furthermore, the security analysis showed that the proposed architecture could ensure authenticity, integrity, and immutability of the data of IoT devices.

The proposed system enables auditors to verify the authenticity and integrity of data. However, the system does not make quality-related maintenance information like calibration of the IoT devices auditable. Therefore, third-party consumers of IoT data cannot verify the quality of received data. Consequently, a distributed audit trail should also be able to record quality-related information. To overcome this limitation, we introduce a DCCI that enables metrological traceability for the IoT in Chapter 5.

Chapter 5

Distributed Calibration Certificate Infrastructure

5.1 Introduction

When data consumers receive IoT data from third parties, they can use PKIs to verify the origin of the data. However, certificates managed by PKI certify the authenticity of identity claims but do not comprise quality information like calibration and maintenance data. Therefore, it is hard for third-party consumers to assess the quality of received data. To overcome this limitation, we propose a DCCI that is able to link quality information like DCCs to identity claims, addressing the research question 3: *How to securely manage quality-related metadata, like calibrations and other maintenance information, of IoT devices?*

We proposed a DLT-based Distributed Calibration Certificate Infrastructure (DCCI) [68] [1]. With the DCCI, calibration certificates can be recorded on the distributed audit trail. This makes this quality-related metainformation auditable for third-party data consumers. By building the DCCI based on the GoT, it comprises an expressive DPKI that meets the specific requirements of metrology.

The main contributions of this chapter are:

- Design and implementation of a DLT-based DCCI using the GoT

[1] Partially reproduced in this chapter - © 2022 IEEE

- Model to represent metrological traceability using DCCs and Accreditation Certificates (ACs)
- Design of a GoT-based multi-signature scheme
- Evaluation of the system in terms of scalability and speed
- Evaluation of sidechains to enable scalability and privacy

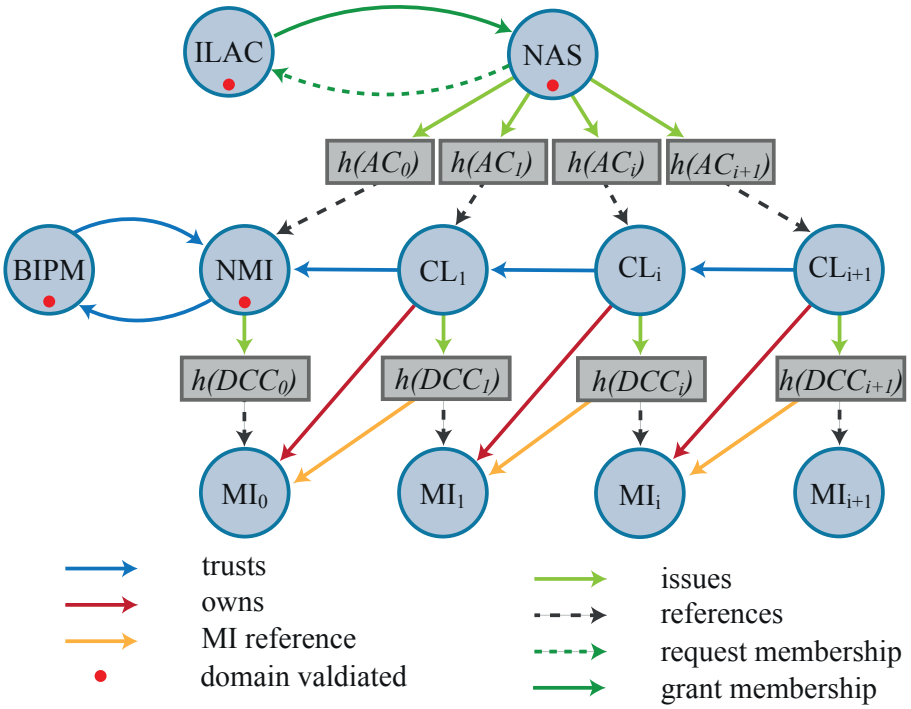
5.2 System Model

Metrology plays a key role in assuring the quality of measurement data. A calibration certificate certifies the accuracy of the corresponding Measurement Instrument (MI). MIs are devices for measuring physical parameters. Note that Sensor Nodes (SN) and IoT devices with sensors are a subset of MIs. In general, the MIs also include analog devices not connected to the IoT. While also DCCs that document the calibration of analog MIs can be signed and secured on our proposed solution, these devices obviously can not secure their measurements on the GoT. However, for simplicity, we focus on IoT devices and assume that all MIs can be connected to the IoT. Consequently, we use the terms IoT device, Sensor Node (SN), and MI interchangeably. A Distributed Calibration Certificate Infrastructure (DCCI) is a distributed system that secures the signatures of calibration certificates and makes the valuable quality information of the DCC accessible to the consumer of the calibrated MI's data. The DCCI comprises a PKI that certifies the authenticity of public keys, stores DCCs, is able to secure DCC signatures, and enables metrological traceability.

5.2.1 Distributed Public Key Infrastructure for Metrology

In Chapter 3, we introduced the GoT, a DPKI that uses a distributed model to derive the certification of identity claims from trust votes. The field of metrology as an application has some characteristics that fit the properties of the GoT well. First, organizations like the NASs accredit calibration laboratories. This accreditation is documented with an AC that can be secured on the GoT. Since this accreditation is stored on the GoT, it can be considered a highly qualified trust vote of the NAS towards the accredited calibration laboratory. Similarly,

the DCCs can be considered a highly qualified trust vote of the calibration laboratory towards the identity claim of a MI. Consequently, these certificates also certify the authenticity of the corresponding identity claims. Second, some organizations in the domain of metrology, like the NASs, NMIs, BIPM, and the ILAC, are nationally or even internationally well-known and can act as trust anchors in the DPKI. In a distributed certification model, these trust anchors help to bootstrap reputation. An auditor can use domain scoring that maps a priori trust to the identity claims these entities. Consequently, with these properties, the model of the GoT suits a DCCI well.



BIPM	Bureau International des Poids et Mesures
NMI	National Metrology Institute
CL	Calibration Laboratory
MI	Measure Instrument
NAS	National Accreditation Service
ILAC	International Laboratory Accreditation Cooperation
AC	Accreditation Certificate
DCC	Digital Calibration Certificate

FIGURE 5.1: Metrological traceability on the GoT [68]

Figure 5.1 shows an example GoT with entities and their signed relations. This

example shows a NAS that has accredited several calibration laboratories and signed the accreditation certificates with declarations on the GoT. Furthermore, in this example, each calibration laboratory owns a MI calibrated by a higher-level calibration laboratory. The DCCs that document these calibrations were also signed with declarations on the GoT. Each DCC references the MI that was used for its calibration. Therefore, our proposed solution secures metrological traceability. Note that the referenced MIs are the standards used for the calibration. The list of these MIs is contained in the DCC.

5.2.2 Digital Calibration Certificate Signatures

The Digital Calibration Certificate (DCC) is a digitally signable XML-based document that comprises all required information of a calibration certificate [71]. On the GoT, an entity can sign any digital document by signing a declaration towards the hash of a document. Each declaration has a type, such as issues, approves, or measures. The type of declaration indicates what property the signer aims to attribute to the document. Documents like calibration and accreditation certificates found in metrology often attest certain properties to another entity. We propose extending declarations with a references attribute to represent these kinds of documents on the GoT. The references attribute contains a list of identity claims and data hashes referenced by the signed document. This new reference relation enables calibration and accreditation certificates to certify MIs and entities.

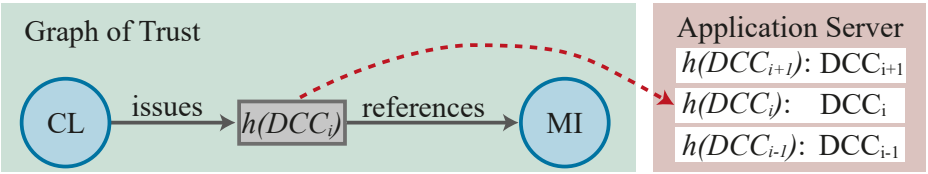


FIGURE 5.2: Digital Calibration Certificate Signature [68]

Figure 5.2 shows an example of a DCC signature on the GoT. The calibration laboratory *CL* has signed an issue declaration towards the *DCC* that references the calibrated *MI*. For privacy and scalability reasons, only this metadata is stored on the GoT but not the DCC itself. However, the declaration that issues the DCC can contain a link to the application server where the DCC is stored.

A hash map is used to find the corresponding DCC on this application server. The application server could be the calibration certificate database of the CL that performed the calibration. This architecture has the advantage that the application server can use its own access management, and the DCCI can be integrated into existing infrastructures.

5.2.3 Accreditation and Audits

Besides the DCC signatures, the expressive declarations of the GoT can also be used to represent accreditations and audits. To our knowledge, we are the first DLT based DCCI to support this kind of declaration. An audit is a professional service delivered by experts in response to economic and regulatory demand [146]. For example, companies that have calibrated their measuring equipment by external calibration laboratories might audit the external laboratories to reduce risk and ensure quality. Either the company that owns the calibrated measurement equipment or a trusted third party can perform an audit. A certificate usually documents the audit and comprises the capabilities of the audited calibration laboratory. Furthermore, accreditation is a special form of audit performed by the national accreditation service.

On the GoT, the audit and accreditation certificates are handled similarly to the DCC. The auditing or accrediting entity signs a declaration towards the hash of an audit or accreditation certificate and adds a reference attribute to the declaration that references the identity claim of the audited or accredited entity. With this information, the auditors that read the GoT can verify if a certain entity is capable of performing a calibration. A link to the certificate can be added to the declaration as it is done with the DCCs.

5.2.4 Metrological Traceability

Each calibration in the chain of calibrations contributes to the uncertainty of the final MI. Metrological traceability is inherently given when all MIs, references, and standards in the chain of calibrations are calibrated. Calibration laboratories accredited by a NAS comply with the corresponding standards, and, therefore, it can be expected that all references used for calibration were

calibrated. Therefore, when an accredited CL calibrates a device, it is not required to verify all upstream calibration certificates. However, the links to the upstream measurement equipment can be provided in the DCC. These links are the fingerprint of the device's identity. If a auditor has access to a calibration certificate, it can search the fingerprints of the direct upstream MIs on the GoT and browse its incoming calibration declarations. This calibration declaration contains a link where the auditor can access the corresponding calibration certificate. Note that only auditors authorized to access the link provided in the certificate signature can access the upstream certificates. The auditor can repeat this process until it has reached the national standard.

5.2.5 Multi-Signatures

A multi-signature is a signature scheme where an entity delegates its signing capabilities to subordinate entities. By delegating the signing capabilities, the entity also defines the minimum number of subordinate entities that have to sign the same declaration to account for this signature. To delegate signing capabilities, the subordinate entity signs a *request subordinate signing* declaration towards the entity's identity claim. After authenticating the request, the entity can sign a *grant subordinate signing* declaration towards the subordinate's entity identity claim to delegate signing to subordinate entities. With the granted declaration, the entity also defines how many subordinate entities have to sign the same declaration for a signature to be valid.

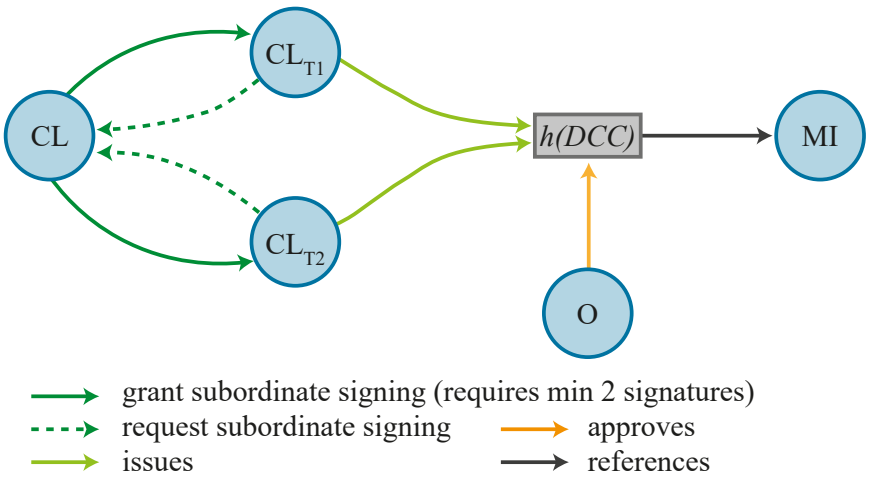


FIGURE 5.3: Multi-Signatures [68] © 2022 IEEE

Figure 5.3 shows an example where the calibration laboratory CL has granted the entities CL_{T1}, CL_{T2} subordinate signing capabilities. However, both subordinate entities have the requirement for a multi-signature of at least two. Therefore, their declarations account only for CL if both have signed the same declaration towards a hash.

In the above discussed multi-signature scheme, an entity defines on the GoT the requirements that need to be met so that the signatures of subordinate entities account for the entity. These requirements are stored on the DLT and binding for all validators. However, the business process of some validating organizations might require some additional rules for a certificate to be valid. For example, some companies that use external calibration laboratories to calibrate their measurement equipment check if the certificate and the measured accuracy fulfill their requirements and approve the calibration certificate before a MI is released. An additional declaration can represent the approval of a certificate. In Figure 5.3, the owner O of the MI has signed an *approves* declaration towards the hash of the digital calibration certificate DCC.

5.2.6 Measurement Values

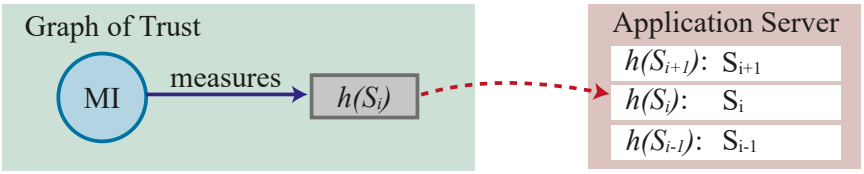


FIGURE 5.4: Measurement Signatures

Our proposed DCCI also enables securing measurement values. To securely link each measurement to its MI, the MI can sign a measurement declaration towards the sample's hash on the GoT. Figure 5.4 shows an example of a measurement declaration. The GoT only stores the samples' hash values $h(S_i)$ but not the samples themselves. However, the identity claim of the MI or the measurement declaration can contain a link to an application server where the measurements of the corresponding MI are stored. A hash map is used to find the corresponding sample S_i on these application servers.

5.2.7 Digital Calibration Certificate Retraction

The DCC retraction is the revocation of an issues declaration that referenced the hash of a DCC. Calibration certificate retraction is an important feature of calibration certificate infrastructures. When a laboratory issues a faulty certificate, it must be able to retract the faulty certificate. While certificate retraction is almost impossible with paper calibration certificates and is difficult with digital certificates containing the signature, it is straightforward with the GoT. To retract a DCC on the GoT, the entity that issued the certificate signs a revocation declaration toward the certificate's hash. The revocation declaration also contains the transaction id of the revoked declaration to assert that the correct signature is revoked if a digital calibration certificate was signed multiple times. Figure 5.5 shows an example of a DCC retraction. The target of a referenced declaration is contained in the same transaction as the declaration. Consequently, the complete DCC is retracted.

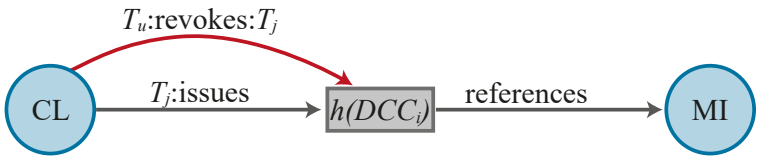


FIGURE 5.5: Digital Calibration Certificate Retraction

5.2.8 Validation

The validation of calibration certificates is an important task of auditors. The auditor first calculates the hash of the certificate and then searches the GoT for this hash to validate a DCC signature. If the hash was found, the validator traverses the most recent incoming issues declaration to find the identity claim which has issued these declarations. If multi-signatures are used (refer to Section 5.2.5), the validator checks if enough subordinate entities have signed the same declaration.

To validate the calibration laboratory, the validator searches the GoT for an issues-references path between the NAS and the examined CL that contains an accreditation certificate. Graph databases like neo4j are optimized for these kinds of traversal algorithms and are fast, as we show in section 5.4.

5.3 Implementation

As a proof-of-concept, we have implemented a functional prototype of the DCCI. Figure 5.6 shows an overview of the prototype architecture. The prototype is written in python and comprises the GoT component that manages the GoT, neo4j as Graph Database to store the GoT, a smart contract that immutably stores the transactions that build the GoT, and the DLT component that connects the nodes and secures the transactions that build the GoT. The DCCI functionality is implemented in the application layer. Furthermore, the prototype comprises a web application to control the node and a REST API to connect further applications.

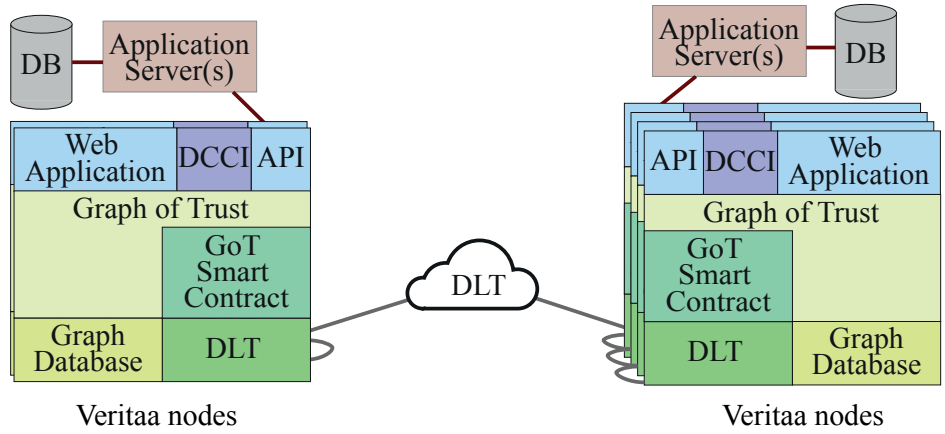


FIGURE 5.6: DCCI architecture

Since our proposed solution aims to secure complete metrological traceability, it must be possible to secure measurement data. With a fast-growing IoT, throughput and storage space form a major challenge if all nodes have to store a replica of the complete data. It has been proposed to use sidechains to solve these challenges [147]. A sidechain is a separate DLT that is attached to a parent DLT, also known as the main chain. While the main chain secures the state of the sidechain, it must not secure all transactions individually and, therefore, needs fewer resources. A sidechain builds consensus on the system state among a few peers and then secures its state on the main chain. Since the GoT basically only consists of identity claims that sign declarations towards hashes, splitting the GoT into several sidechains is trivial. To split the GoT into sidechains, each sidechain stores a subset of the identity claims and their declarations that are submitted to this sidechain. Since the declarations always

point to data hashes or hashes of identity claims, it is always possible to build a subgraph of the GoT out of the transactions of a sidechain. For example, suppose a declaration signed on sidechain A points to an object managed on sidechain B. This declaration points to a hash in that subgraph until the referenced object managed on sidechain B is loaded. However, when the node subscribes to all relevant sidechains and applies all transactions on its GoT, it results in a subgraph with all relevant information for further processing. Since the hash of identity claims and declarations is the same on all sidechains, identity claims and declarations can also be submitted to multiple sidechains to sign declarations on multiple sidechains. Transactions with the same hash are only added once to the GoT. Therefore, an object submitted to two sidechains is only added once to the GoT when both sidechains are loaded.

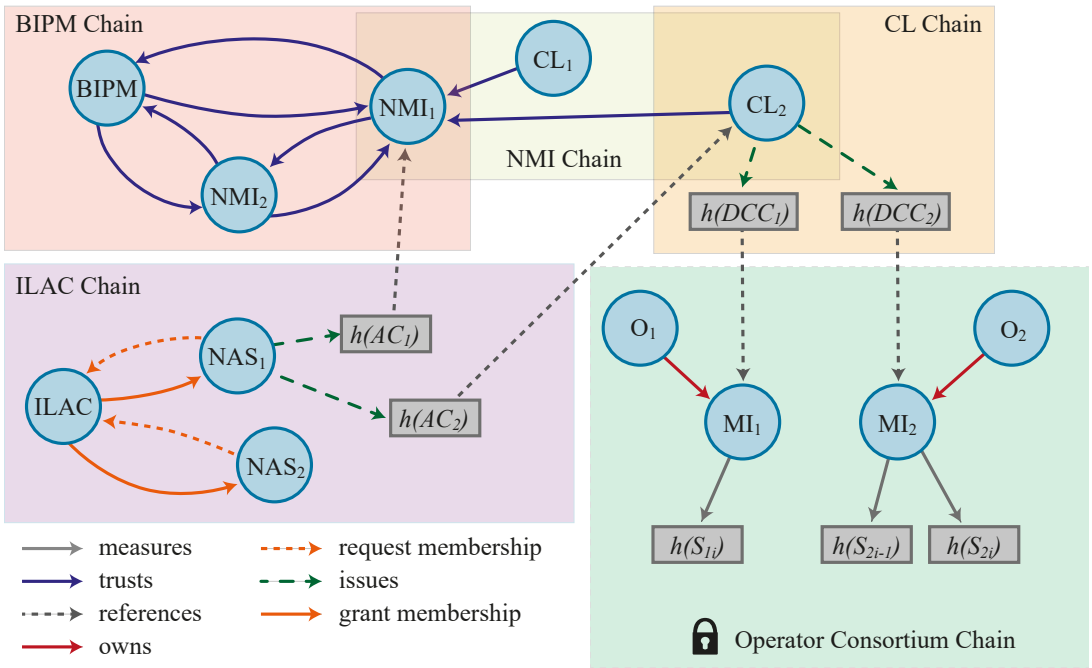


FIGURE 5.7: Sidechains

Figure 5.7 shows how the DCCI could be split into several sidechains. Some natural clusters like the NASs or the NMIs appear obvious as sidechains in metrology. An advantage of this architecture is that it enables higher granularity for permissions. For example, the ILAC could build a sidechain with all its members and manage the accreditation certificates. This NAS sidechain could be a permissioned but public sidechain where only ILAC members have

permission to contribute transactions, and the public has permission to read transactions. As another example, a consortium of IoT operators might want to observe some physical parameters in privacy and build a permissioned private sidechain. A further advantage is that each node only has to secure the data of the sidechains it subscribes. Therefore, data-intensive applications like sensor networks must only be secured by the nodes that are interested in the data and its quality.

5.4 Evaluation

5.4.1 Scalablilty

The immutability of DLTs results from many nodes with copies of their data and consensus about its state. However, when each transaction is stored on all network nodes, the required disk space increases with each new node and transaction. Sidechains can reduce the required disk space and enable scalability. To evaluate the scalability of our proposed solution, we simulated the effect of the sidechains on the required storage. Therefore, we simulate how many times a certain transaction is stored in the network. In contrast to a single-chain setup, a transaction is only stored on the sidechain nodes and not in the whole network of a sidechain setup. The simulation considers that not all sidechains have the same size, and not all network nodes create the same amount of transactions. We use Zipf's distribution to model the sidechain sizes and node activities.

Figure 5.8 a) shows how many times a transaction is stored with different network and sidechain sizes. The simulation shows that already using a few sidechains decreases the redundantly stored transactions significantly.

Since the size of the clusters and their activity can vary, the standard deviation is important to evaluate the effectiveness of sidechains. Figure 5.8 b) shows the standard deviation of the redundantly stored transactions. The evaluation of the standard deviation shows that the number of sidechains should be increased when the network has more participants. Since the DCCI would grow by adding new calibration laboratories, and each CL has its sidechain, it is

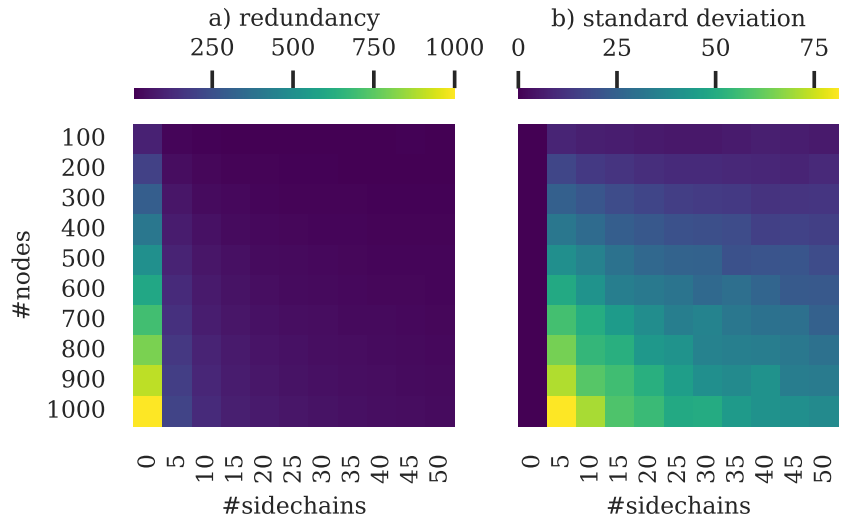


FIGURE 5.8: Redundantly stored transactions [68] © 2022 IEEE

expected that the number of sidechains will also increase together with the number of participating nodes.

5.4.2 Validation Performance

For the validation of certificates, graph traversal algorithms are required. To evaluate the validation performance, we have generated a random GoT with 10^7 identity claims that follows the Watts-Strogatz generation model [148]. We extended this random graph with accreditation and calibration certificate declarations. The experiment was conducted on a virtual machine with 8GB RAM and 4vCPUs.

Figure 5.9 shows the time required to validate the different objects of the DCCI. The time needed for the validation of a specific object depends on how many declarations must be verified. Only one declaration must be validated to validate the direct DCC signature. Therefore, this validation is fast. The multi-signature with one required subordinate signature $MS1$ requires at least three declarations. Thus, the $MS1$ validation is slower than the DCC signature validation. When more subordinate signatures are required with the multi-signature scheme MSn, more declarations must be validated. Consequently, the time required for the validation is increasing. To validate a CL, it must be validated that the CL has a valid accreditation certificate of a NAS which is a member of the ILAC. Therefore, three declarations must be validated.

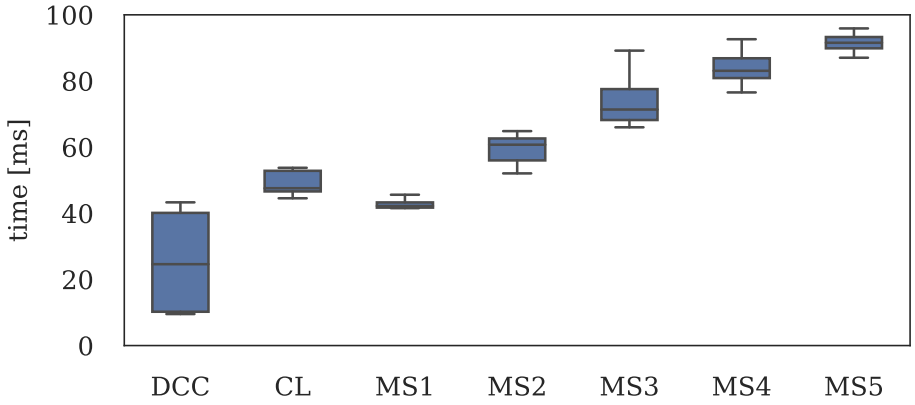


FIGURE 5.9: Validation times [68] © 2022 IEEE

Most time is required to look up the object's hash that must be verified. When this object is found, the traversal API of the graph databases are fast. Therefore, the validation of the DCC has the highest variance, even if only one declaration must be verified. This variance is the result of how queil the search algorithm finds the hash. This experiment shows that the validation of DCCs, MSns, and CLs is fast for a large GoT.

5.5 Conclusion

When sensor data is shared between the infrastructures of multiple organizations ensuring data quality, integrity and authenticity become important. In this chapter, we proposed a DCCI. The DCCI comprises a DPKI and declarations signed by the keys managed by the DPKI. These declarations enable metrological traceability starting from the measurement sample over its calibrated MI to the national standards.

Our proposed solution relies on the few core transactions that build the GoT. These transactions are expressive and enable the representation of the real-world relations between entities relevant to metrological traceability. Therefore, our solution can manage calibration and accreditation certificates. Integrating national accreditation services into the DCCI has the advantage that the NASs act as trust anchors in the DPKI. Furthermore, our model provides a flexible signature scheme that enables multi-signatures and certificate approval processes. Additionally, we discussed how sidechains could enable

privacy and scalability. The evaluation shows that our solution is scalable and fast.

A limitation of the proposed DCCI is that, in contrast to the DCC, no digital accreditation certificate has been proposed yet. To fully integrate the NAS into the DCCI, such a digital accreditation certificate would be required. However, proposing this standard is out of the scope of this thesis.

Chapter 6

Auditable Positioning System

6.1 Introduction

The position must also be auditable for applications where the integrity depends on the position of IoT devices. To make positions auditable, an anchor point must perform a PoL to prove that the asset tag is within its communication range. Furthermore, the anchor point must secure this PoL on an audit trail to make the position auditable. In this chapter, we propose APS, an auditable positioning system based on AoA, PoL, and the GoT addressing research question 4: *How to make positions auditable with a distributed audit trail?*

We proposed an APS that comprises a PoL, a distributed audit trail for positions based on the GoT, and a DPKI for the authentication of the tracked asset tags [149] [1]. Using the GoT for the audit trail has the advantage that it enables the description of relations between data. Therefore, this novel APS cannot only secure the positions but also link the position to the raw data used to estimate these positions. These links make the positioning algorithm transparent and enable data provenance. We implemented a real-world testbed to evaluate the performance of the proposed system. For the evaluation, we measure the additional energy required to secure the position claims, the impact of the position advertising frequency on the system's energy consumption, and the impact of the position sampling frequency on the required system storage data rate.

[1] Partially reproduced in this chapter - © 2023 IEEE

The main contributions of this chapter are:

- Design and implementation of an APS based on the GoT that uses a BLE-based AoA positioning system
- Design of a model that makes data provenance of positions visible
- Mathematical definition of auditability for positioning in terms of sampling rate
- Evaluation of the proposed system in terms of energy consumption and disk storage requirements

6.2 System Model

An auditable positioning system must be able to prove, to a third-party auditor, that stored position information of an asset tag is trustworthy by making it impossible for an attacker to manipulate position information. The APS proposed in this chapter is based on a standard AoA-based positioning system as we introduced it in Section 2.5. This standard positioning system is composed of asset tags, locators, and a centralized positioner.

An asset tag is an energy-constrained wireless device attached to an asset to monitor physical parameters like the asset's position. A locator is an anchor point at a fixed and known position that estimates the relative direction of asset tags using BLE-based AoA direction finding. A positioner is a multi-angulation algorithm run on a server, which estimates the position of an asset tag using the relative angles measured by locators.

6.2.1 Authentication and Proof of Location

We assume that the asset tag is paired (refer to Section 4.3.3) with the positioner. The provisioning consists of publishing the identity claim of the asset tag on the GoT and exchanging the public keys of the positioner and the asset tag. With the public keys exchanged and the identity claim made accessible on the GoT, the locators and the positioner can validate the signatures of the asset tag. Furthermore, the asset tag can validate the positioner's signatures. Additionally, asset tags, locators, and the positioner can use the exchanged public

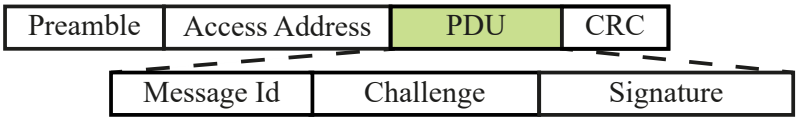


FIGURE 6.1: Challenge packet [149] © 2023 IEEE

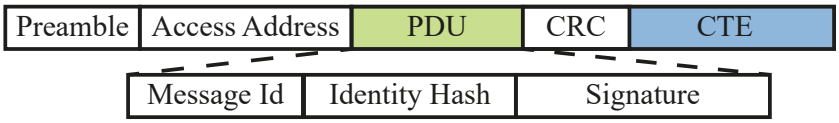


FIGURE 6.2: Response packet [149] © 2023 IEEE

keys to encrypt privacy-relevant payloads like sensor readings. With the public keys exchanged, the locators and the positioner can authenticate asset tags with a cryptographic challenge.

In our proposed system, every locator authenticates each message received from the asset tag through a challenge-response mechanism, which is needed to guarantee the message sender's identity. We define the challenge packet as a BLE extended advertisement frame broadcasted by the locators containing a message id, a challenge payload (i.e., a random number), and the message's signature for authentication (Figure 6.1). We define the response packet as a BLE frame transmitted by the asset tag, containing a message id, the asset tag's identity hash, and the message's signature for authentication (Figure 6.2). The message id of the challenge and response packets is used as a replay counter, and duplicate packets with the same message id are dropped.

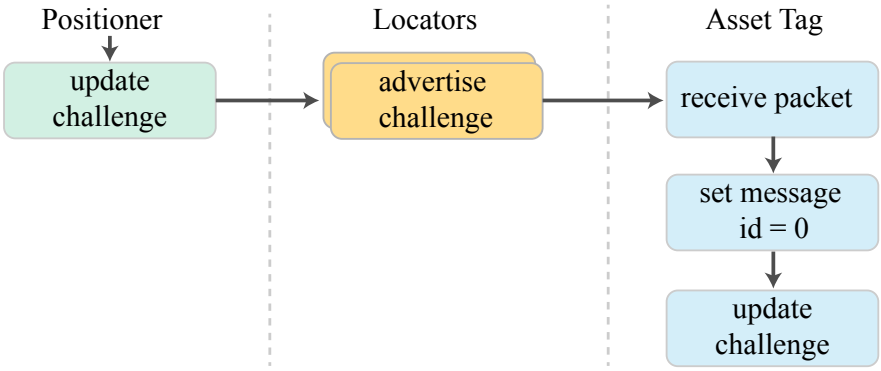


FIGURE 6.3: Challenge update

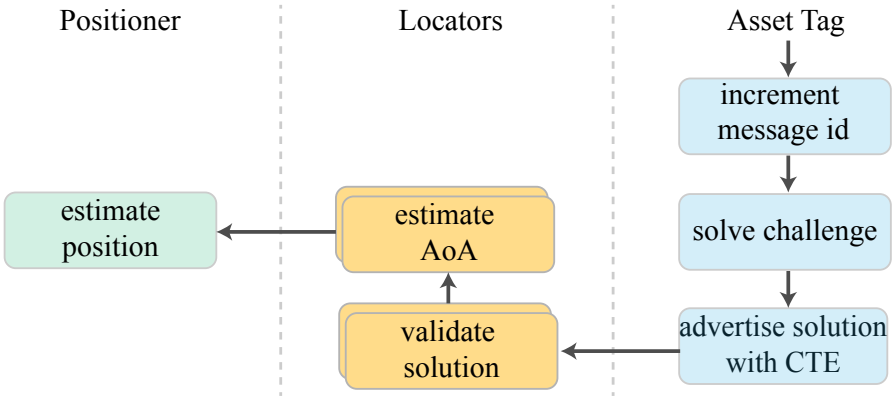


FIGURE 6.4: Asset tag position estimation

In the proposed system, the positioner orchestrates the challenge packet generation, the locators broadcast the challenge packets, and the asset tags reply with response packets as detailed hereafter. To prevent replay attacks, the positioner updates the challenge payload encapsulated in challenge packets every τ_c seconds and disseminates the new challenge throughout the system via the locators. Figure 6.3 shows the process of updating the challenge. To update the challenge, the positioner first creates a challenge packet payload by generating a random number (i.e., the challenge payload) and message id, then signs them with its private key, encapsulates the payload in a challenge packet and forwards it to the locators. The locators, after receiving a valid challenge packet from the positioner, validate that the positioner has signed it and start advertising the challenge packet periodically. This procedure ensures that all locators broadcast the same challenge. The asset tag, after receiving a challenge packet, verifies that a paired positioner has signed the challenge packet, sets the message id to zero, and updates the challenge payload in its internal memory. Finally, when the asset tag has updated the challenge internally, it can use the new challenge payload for authentication until the positioner has generated the next challenge.

With each position advertisement, the asset tag solves the challenge in order to authenticate itself. Figure 6.4 shows the process of advertising a position. First, to authenticate itself and as proof that it is in the range of the locators, the asset tag solves the challenge by signing and advertising the incremented message id, its identity hash, and the random number of the challenge. Message id and signature are updated for each broadcast. Then, when the locators receive the

response packet, they verify the signature of the asset tag, check that the message id of the solution is larger than the message id of the previous solution, and estimate the relative angle to the asset tag. Finally, the locators forward the estimated angles to the positioner, which estimates a position from all received angles with the same solution message id.

This challenge and response approach proves that the asset tag is in the range of the locators with the uncertainty of the time required for updating the challenge on the asset tag. Note the PoL that certifies that the asset tag is close to the locators is always provided by the first response packet after a challenge update. For further and more fine-grained accuracy of the PoL the response of this challenge-response approach must be tied to the measurement of the physical parameter required for the direction estimation.

6.2.2 Secure Direction Finding

For estimating the asset tags' relative direction, the locators use BLE direction finding. The recent BLE 5.1 standard introduced the CTE which enables AoA positioning [86]. The CTE is a CW appended after the CRC of a BLE packet. The design of the BLE-CTE has the advantage that the locators not only measure a CW but also receive data in the same packet. This data can be used for the cryptographic authentication of the asset tag that has sent the CTE-enabled packet.

Figure 6.2 shows an CTE-enabled packet with a signed response in its payload. Since the CTE is directly sent after the CRC, it is hard for an attacker to manipulate the CTE, and therefore, the signature in the payload also secures the authenticity of the CTE up to a certain degree. However, a limitation is that since the CTE is a physical analog signal, cryptographic functions cannot directly protect the signal used for positioning. In Chapter 7, we show how we can overcome this limitation by extending the APS with an LVS that can detect positions that were manipulated on the physical layer.

6.2.3 Data Provenance

In an auditable positioning system, the auditor must be able to verify that an asset tag is associated with a claimed position within a certain tolerance. Typically, positioning systems require two steps to calculate a position. First, locators estimate the relative distance or angle towards the tracked asset tag, and second, a positioner uses the estimates of multiple locators to estimate the position. The proposed APS enables securing the raw data and the position on the GoT. Furthermore, the GoT links the position to the raw data used to estimate this position. These links make the positioning algorithm transparent and enable data provenance.

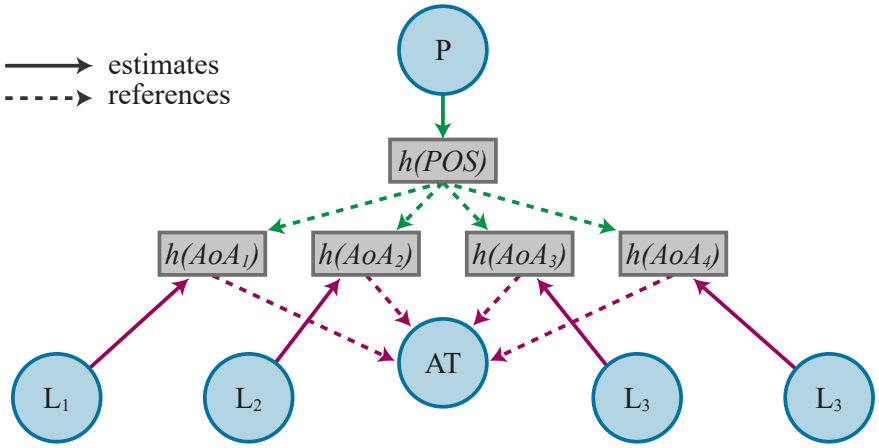


FIGURE 6.5: GoT position trace

Figure 6.5 shows an example GoT subgraph with a position trace of a position estimated with an PS with four locators L_i and one positioner P. Each locator L_i has signed an estimates declaration towards the hash of the estimated angle that references the asset tag AT. Furthermore, the positioner signed an estimates declaration towards the hash of the estimated position that references all hashes of the angles used to estimate this position.

This data structure enables auditors to verify the provenance of data. To verify the provenance of a position, the auditor calculates the position's hash and searches this hash on the GoT. Afterwards, the auditor can traverse the GoT by the references relations to find the hashes of the angles used to estimate this position. Finally, starting from these hashes, the auditor finds the locator that

measured this angle by following the estimates declaration and the tracked asset tag by following the references relation.

Consequently, this data structure makes the positioning transparent to the auditor and, therefore, auditable. Note that this data structure also applies to other multi-level positioning systems like UWB ranging and multilateration.

6.2.4 Low Power Scan

Since unsecured asset tags only need to broadcast a CTE enabled extended advertisement, AoA based PS are energy-efficient. Nevertheless, APS aims to make the position of low power IoT devices auditable. Therefore, these devices require to exchange additional data and perform cryptographic operations. Recent low-power microcontrollers like the EFR32BG22 provide all required cryptographic operations like elliptic curve signatures at high speed and with low energy requirements, as we have evaluated in Chapter 4. However, to authenticate and secure the asset tag, information must be exchanged between the locators and the asset tags, which requires additional energy. This additional energy consumption occurs either when scanning for advertisement packets or for maintaining the BLE connection. We assume that locators are placed at fixed positions and have an unlimited energy supply, in contrast to the asset tags that have a constrained energy budget. Therefore, the energy consumption of most deployments can be optimized for the asset tag.

The asset tag must not constantly scan for updates to optimize the energy consumption of the asset tag. Consequently, the asset tags scan window t_{sw} should be minimized and the scan period τ_s maximized. The t_{sw} is the time the BLE module is listening to the channel to receive packets, and τ_{sp} is the time between two scan windows. To ensure that an asset tag receives advertisements, the advertisement period τ_{al} of the locator must be shorter than t_{sw}.

Figure 6.6 shows an example of a low-power scan. The locators L_i broadcast the challenge with the advertisement period τ_{al}. The asset tag (AT) periodically listens for the advertisements from the locators for t_{sw} seconds, and updates its response after receiving a new challenge. The locator broadcasts the challenge periodically until the positioner sends a new challenge. Locators only accept AoA packets with a message id greater than the message id of the last packet

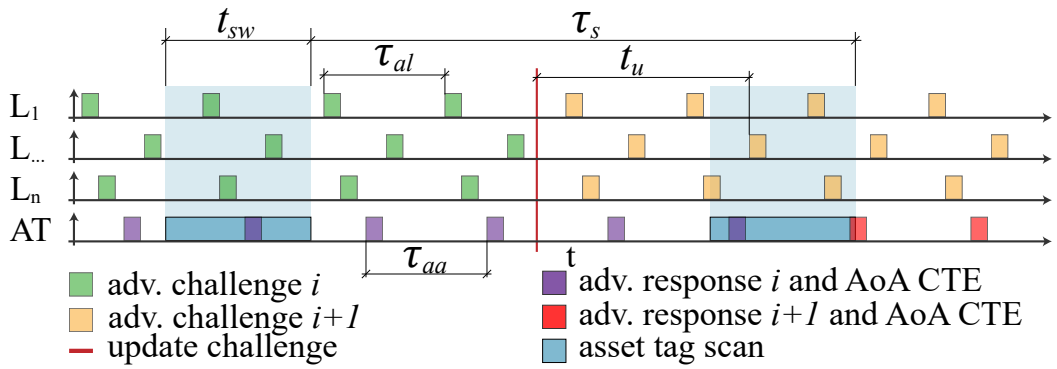


FIGURE 6.6: Low power listening for advertisement updates [149] © 2023 IEEE

to prevent replay attacks. Since the update delay t_u depends on τ_s, the locator accepts responses to the challenge i and $i-1$ to enable low-power scanning on the asset tags.

6.2.5 Sampling and Storage Requirements for Auditability

A positioning system that monitors asset tags' locations produces a continuous data stream. Already storing the data stream of the locators and positioner without securing it can consume several GB of disk space per day depending on the sampling rate. Securing the data on a DLT requires additional hashes and signatures, further increasing the required disk space. Furthermore, the immutability of the DLT is assured by storing replicas on independent nodes of the network that multiplies the required disk space by the number of nodes. Consequently, the data rate is an important challenge for auditable positioning, and applications must be designed to optimize the required disk space.

Let us define the sampling period τ as the time between two consecutive broadcasts of a CTE-enabled advertisement by the asset tag, which is equivalent to the time between two consecutive estimations of the asset tag's position by the system. Let us define the asset tag's true position at time $t \in \mathbb{R}$ as $p(t) \in \mathbb{R}^3$, and the asset tag's true position sampled at time $k\tau \in \mathbb{R}$ as $p(k\tau) \in \mathbb{R}^3$. We assume that the asset tag moves at speed upper-bounded by $v_{\max} = \max_{t\in\mathbb{R}} \|\dot{p}(t)\| \in \mathbb{R}_+$. Under these assumptions, we can say that the

sampling error is bounded by $\frac{1}{2}\tau v_{\max}$, i.e.,

$$\max_{t\in\left[k\tau-\frac{\tau}{2},k\tau+\frac{\tau}{2}\right)} \|p(t)-p(k\tau)\| \leq \frac{1}{2}\tau v_{\max}, \quad \forall k \in \mathbb{Z} \tag{6.1}$$

Let us define the estimated position $s(k\tau) \in \mathbb{R}^3$ of the asset tag generated by the positioning system at time $k\tau \in \mathbb{R}$, where $k \in \mathbb{Z}$ is the sample index and $\tau \in \mathbb{R}$ is the sampling period. The estimated position $s(k\tau) = p(k\tau) + \varepsilon$ is the sum of the sampled asset tag position and a random error vector $\varepsilon \in \mathbb{R}^3$. We assume that the random error vector ε can follow any distribution but its norm is bounded by $\varepsilon_{\max}$, i.e. $\|\varepsilon\| \in [0, \varepsilon_{\max}]$. It is worth noting that the function $p(t)$ is unknown to the system, whereas the function $s(k\tau)$ represents the actual system measurements.

For positions to be auditable, the positioning system should be designed in a way that guarantees that the tracked asset tag cannot physically be located further than a tolerance distance δ from any claimed location, where δ is an application-specific parameter. We can now formally define auditability as in Definition 1.

Definition 1. *A positioning system with sampling period τ is auditable for a tolerance distance $\delta \in \mathbb{R}_+$ if and only if $\max_{t\in\left[k\tau-\frac{\tau}{2},k\tau+\frac{\tau}{2}\right)} \|p(t)-s(k\tau)\| < \delta, \quad \forall k \in \mathbb{Z}$.*

Theorem 1 (Strong Auditability, Sufficient Condition)**.** *Given a tolerance distance δ, a maximum asset speed v_{max}, and a random error vector whose norm is bounded by ε_{max}, a positioning system is auditable if its sampling speed τ is $\tau < 2\frac{\delta-\varepsilon_{max}}{v_{max}}$.*

Proof. We can combine the triangle inequality rule and Equation 6.1 to write the chain of inequalities:
$\|p(t)-s(k\tau)\| = \|p(t)-p(k\tau)-\varepsilon\| \leq \|p(t)-p(k\tau)\| + \varepsilon_{\max} \leq \frac{1}{2}\tau v_{\max} + \varepsilon_{\max}, \forall t \in \left[k\tau-\frac{\tau}{2}, k\tau+\frac{\tau}{2}\right), \forall k \in \mathbb{Z}$. Therefore, $\frac{1}{2}\tau v_{\max} + \varepsilon_{\max} < \delta \implies \max_{t\in\left[k\tau-\frac{\tau}{2},k\tau+\frac{\tau}{2}\right)} \|p(t)-s(k\tau)\| < \delta, \quad \forall k \in \mathbb{Z}$ and therefore $\tau < 2\frac{\delta-\varepsilon_{\max}}{v_{\max}}$ implies auditability.

□

It is worth noting how, given the fixed system parameters δ and $v_{\max}$, an inaccurate positioning system with a high $\varepsilon_{\max}$ reduces the maximum acceptable

sampling period. If the system positioning error ε_{max} is greater than the tolerance distance δ, there is no sampling period τ that enables the system to provide auditability.

6.3 Evaluation

6.3.1 Testbed

To evaluate the proposed system, we have built a real-world testbed comprising an asset tag, one positioner, four locators (i.e., the anchors), and a server that executes the Veritaa framework. The asset tag is a Silicon Labs EFR32BG22 development board. All four locators are also Silicon Labs EFR32BG22 development boards provided with a PCB4185A antenna array. The positioner and the locators use the Silicon Labs' Real Time Locating Library [150] for direction finding and positioning. BLE direction finding is available for periodic advertisements, extended advertisements [150], and connection-based packets. Extended advertisements enable a Protocol Data Unit (PDU) of up to 191 B, which enables the exchange of cryptographic information, and they do not require maintaining a connection like connection-based characteristics [150].

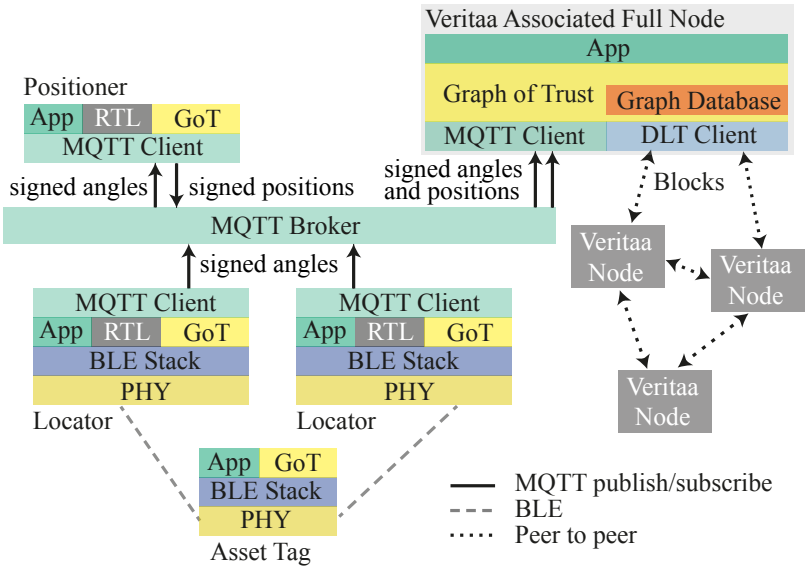


FIGURE 6.7: APS logical architecture [149] © 2023 IEEE

Figure 6.7 shows an overview of the testbed. Asset tags and locators communicate over BLE, and locators, the positioner, and the Veritaa AFN communicate

over MQTT. All asset tags, locators, and the positioner comprise a GoT component that manages their identity and is able to create and sign statements. Data is exchanged over the Veritaa AFN to communicate with the Veritaa network and its DLT.

6.3.2 Energy Consumption

Authenticating the asset tags and securing the positions requires additional energy. The extended advertisements contain the response to the challenge in the PDU to secure the positions. With this response, more data must be broadcasted. Consequently, more energy is needed.

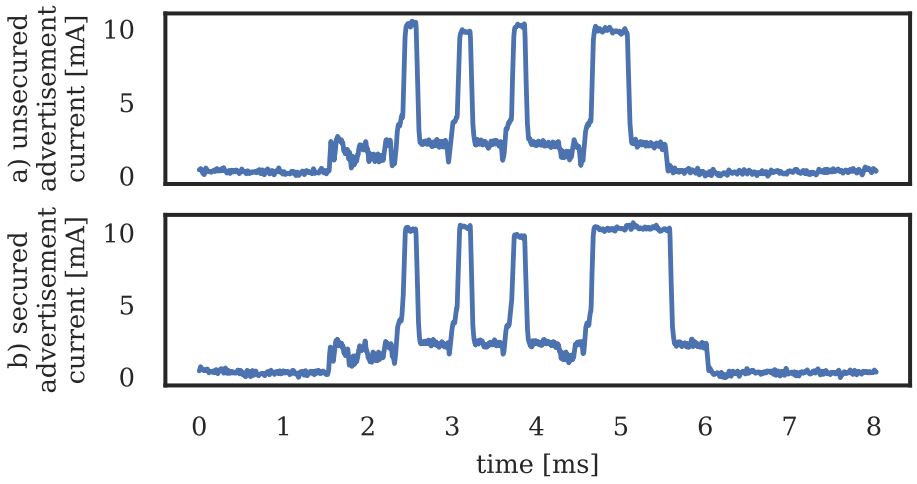


FIGURE 6.8: Current analysis

Figure 6.8 shows the current over time of a secured and of a non-secured extended advertisement with enabled CTE. First, the packet is advertised in the three advertisement channels of BLE, and afterwards, the data with the CTE is broadcasted. For the secured advertisement, we observe that the data broadcast requires more time and, therefore, more energy due to the additional data.

Figure 6.9 shows the energy required to broadcast an unsecured and a secured extended advertisement with enabled CTE. This experiment shows that the secured advertisements require 30% more energy than the not secured ones.

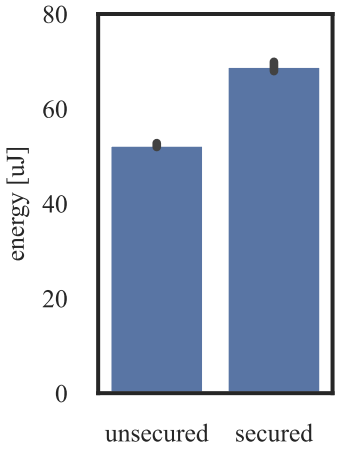


FIGURE 6.9: Energy consumption

Securing positions requires more energy for the CTE enabled extended advertisements and for receiving updates from the locators. The following experiment evaluates the average power consumption of asset tags for different advertisement periods τ_{aa}. In this experiment, we measured the average power for unsecured positioning, secured positioning with updates over a BLE characteristic, and secured positioning with updates over extended advertisements. For this experiment, the challenge update interval is set to 30 s. For the extended advertisement measurements, the scan interval was set to 30 s and the scan window to 100 ms. The locators broadcast the update with a 50 ms interval. The average power consumption over 120 s is calculated for each measurement.

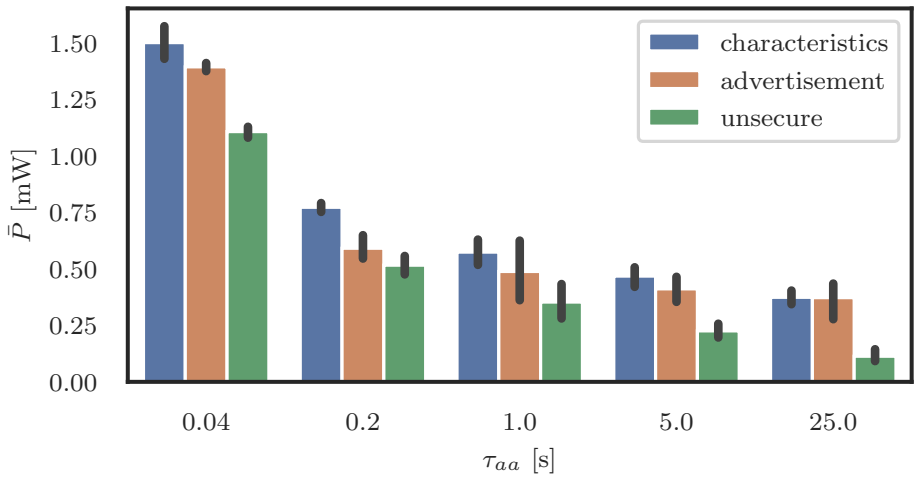


FIGURE 6.10: Average power consumption. Confidence intervals at level 99% for five experimental repetitions.

Figure 6.10 shows the average power consumption of the asset tag. The average power consumption mainly depends on the broadcast interval. However, the three challenge update methods have slightly different results. First, for short update periods scanning for advertisements accounts for most of the energy budget, but it only accounts for a little of the energy budget with long update periods. Second, updating the challenge over BLE characteristics requires a connection. This connection must be established and maintained, which requires energy and limits scalability because the asset tag and the locators must keep track of these connections. However, updating the characteristics does not require scanning on the asset tag's side, so the update performs better at longer sampling periods. Finally, since the asset tag only needs to broadcast the CTE enabled extended advertisement, the unsecured AoA positioning requires the least energy.

6.3.3 Sampling and Storage Rate

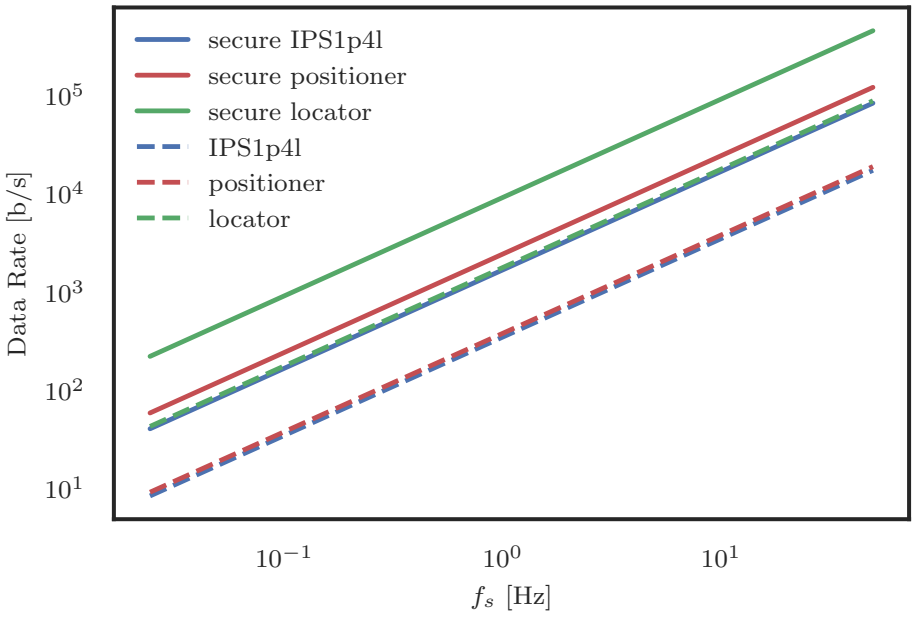


FIGURE 6.11: PS data rate at different sampling rates

Depending on the sampling rate, storing a data stream requires disk space. Additionally, securing and immutably storing the data increases the data rate and, therefore, the required disk space. To evaluate the impact of securing the positions on disk usage, we calculated the data rate of the different PS components. Figure 6.11 shows the data rate in function of the sampling rate for

secured and not secured data streams of a locator, the positioner, and a setup with one positioner and four locators (1P4L). Since signatures and data hashes are of constant length and the sample length does not change with the sampling rate, the data rate linearly depends on the sampling rate. This evaluation shows that storing the PS data with a high sampling rate leads to a high data rate. Securing this data requires roughly seven times more disk space than storing the data unsecured. Consequently, the sampling rate should be optimized with the data rate objective and the application's requirements constraint. Note, when the data is secured on a DLT, then the data is additionally replicated on multiple nodes. Sidechains can be used to scale data-intensive DLT applications, as discussed in Chapter 5.

In this chapter, we propose to adapt the storage rate to the actual speed of the tracked assets and the tolerances of the application to minimize the required disk space. To evaluate the effect of this adaptive storage rate, we calculate the average storage rate for three different movement profiles that can be found in a warehouse application shown in Table 6.1.

TABLE 6.1: Mobility profiles [149] © 2023 IEEE

Profile	Mobility	Tolerance	Remark
A	35% 0m/s 45% 1.5m/s 20% 3.6m/s	3m	16h/day in use (e.g., forklift)
B	99.95% 0m/s 0.015% 1.5m/s 0.031% 3.6m/s	1m	20min/month in motion (e.g., parcel moved by a forklift in a warehouse)
C	91.7% 0m/s 8.3% 1.5m/s	0.5m	2h/day in motion (e.g., asset moved by person)

Figure 6.12 shows the average data rate for the investigated mobility profiles. For the static and dynamic evaluation, the sampling period τ_s is set to the maximum value that complies with the tolerances. Setting the sampling to this static value already reduces the data rate below 10 kb/s for secured positions. With the adaptive storage model, the storage rate is adapted to the actual velocity of the tracked asset tag subject to strong auditability. Therefore, the system still estimates the positions every τ_s seconds but only secures the

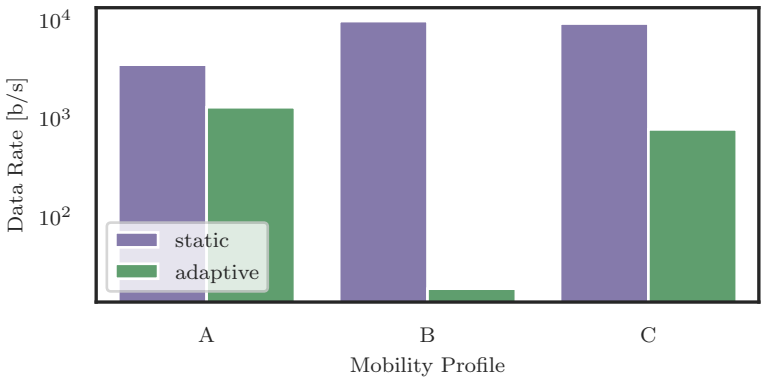


FIGURE 6.12: Data rate of static and dynamic sampling rates for different mobility profiles

position on the DLT when the tracked object moves more than the accepted tolerance. With this adaptive strategy, the storage rate adjusts to the actual movement of the tracked object. The evaluation shows that depending on the movement profile, this adaptive storage strategy can reduce the storage rate significantly.

6.3.4 Security Analysis

Adversaries might attempt to attack the APS in order to manipulate or forge a PoL, or a GoT transaction not yet stored on the DLT. An adversary can eavesdrop, modify and forge packets. In this chapter, we assume the used cryptographic functions and private keys are secure, and the transactions stored on the GoT are immutable and secure. Furthermore, we assume that the microcontrollers' cryptographic co-processors and tamper detection systems are secure.

An attacker might manipulate or forge a CTE-enabled advertisement packet to manipulate the PoL. Let us assume an attacker successfully manipulated or forged a CTE-enabled advertisement packet of an asset tag. Then the packet contains a valid signature of the attacked asset tag. This contradicts the assumption that the cryptographic functions are secure. Consequently, the attacker cannot forge or manipulate a packet.

An attacker might perform a replay attack to manipulate a position. Let us assume an attacker reads a CTE-enabled advertisement packet and rebroadcasts

it at a different location. Then the locators will receive two challenge solutions with the same message id. However, the locators ignore packets with the same message id for the same solution, so the replay attack fails.

An attacker might attempt to forge a location by pre-signing a couple of challenges and broadcasting them from another device. Let us assume that an attacker has signed some challenge and message ids of the future and later broadcasts it from a different device at a different location. Since the attacker could not use the random number of the challenge in the future, the pre-signed responses contain outdated challenges. Consequently, the locators can detect the attack and drop the packet.

An attacker might try to manipulate or forge a GoT transaction not yet stored on the DLT. Let us assume an adversary successfully changed or forged a GoT transaction. Then the statement containing this transaction comprises a valid signature of the creating locator or positioner. However, this contradicts the assumption that the cryptographic functions are secure.

Privacy is an important property of tracing applications. For an audit log, the entity that owns the infrastructure must be able to limit access to this audit log. The structure of the GoT enables private subgraphs that are only shared among permissioned peers. To build a private subgraph, the parties aiming to make a PS application auditable set up a new sidechain when they use the IOTA GoT client or a new private channel when they use the Hyperledger Fabric GoT client. All transactions of these private subgraphs are only exchanged with the peers of the IOTA sidechain or the Hyperledger Fabric channel and, therefore, only visible to the permissioned peers.

A limitation is that since the CTE is a simple CW, the CTE cannot be directly secured with cryptography. Therefore, the physical parameter (i.e., the CW) used for the angle estimation is vulnerable to a wormhole attack. A wormhole attack is an attack where the attacker records a packet and retransmits it at a different location without changing the packet itself. In this case, an attacker could listen to CTE-enabled advertisements and jam the channel from the start of the CRC to the end of the CTE. Since the CRC is jammed, the locators would drop the packet. Nevertheless, the attacker can still record the packet without verifying the CRC. After the jamming, the attacker can replay the recorded

packet at a different location with a newly calculated CRC. Since the data do not change with this attack, cryptography cannot detect it. And when the attacker rebroadcasts this packet before the asset tag sends the next advertisement with incremented message id, the locators will not detect a replay attack. Since cryptography cannot mitigate this attack, it must be handled with a LVS. In Chapter 7, we extend the APS with a LVS to overcome this limitation.

6.4 Conclusion

In this chapter, we proposed APS a system to make positioning auditable. The proposed solution uses PoL to assert that a tracked asset tag was at a claimed position and secures this PoL on the GoT. Furthermore, this work discussed how the sampling rate could be optimized to improve energy consumption. Moreover, we proposed an adaptive storage rate to reduce disk usage of applications with high sampling frequencies.

As a proof of concept, we built a testbed for the APS and used this testbed for evaluations. The evaluation shows that performing and securing PoL, but also that additional required energy is limited and that auditable positioning is feasible for low-power IoT applications. Furthermore, the evaluation shows that adopting the sampling rate to the maximum speed of the tracked asset tags saves energy for all evaluated positioning methods. Additionally, calculations show that adapting the storage rate to the effective speed of the asset tag can significantly reduce disk usage. Finally, the security evaluation showed that our proposed PoL is secure. However, the security evaluation revealed the limitation that the BLE-based AoA positioning is vulnerable to a certain wormhole attack. In Chapter 7, we will show how a LVS can overcome this limitation.

Chapter 7

SecureAoX: A Location Veritifaction System for AoA Positioning

7.1 Introduction

In Chapter 6, we introduced a system capable of making positioning systems auditable. However, while the evaluation showed that this system is secure on the cryptographic layer, we found a vulnerability on the physical layer of BLE-based AoA positioning. This vulnerability enables attackers to manipulate positions by performing a wormhole attack. To perform this attack, the attacker can use a reactive jammer to block the transmission of an CTE enabled BLE packet and replay it at a different location. Since this attack does not tamper with cryptography, cryptography cannot be used to protect against it. To overcome this issue, we built an LVS based on independent AoA and AoD measurements addressing research question 5: *How to build an LVS for BLE-based AoA IPSs using AoD as independent verification information?*

We proposed SecureAoX, an LVS for BLE-based AoA and AoD positioning systems [85][1]. SecureAoX uses AoA for the positioning and AoD measurements as independent position information for the verification and vice versa. Since the AoA is measured on the anchor points and the AoD on the asset tag, AoA and AoD are independent measurements. When a position has not been manipulated, the distance between the AoA and AoD positions should be within the expected error of the IPS. Positions larger than the expected error can be

[1] Partially reproduced in this chapter - ©2022 IFIP

classified as attacks. We designed the system and used Monte Carlo simulations to evaluate its performance. Furthermore, we evaluated the security of SecureAoX.

The main contributions of this chapter are:

- Design the architecture of SecureAoX, an LVS for BLE-based AoA and AoD positioning systems
- Implementation of a simulator to evaluate the proposed system
- Evaluation of the classifier's precision, recall, and accuracy for different system's accuracies
- Comparison of the classifier with the related work
- Real-world accuracy measurements of a BLE-based AoA positioning system to asses realistic classifier accuracies

7.2 System Model

7.2.1 Architecture

SecureAoX is an LVS for BLE-based AoX positioning systems. The system comprises asset tags, locators, and a positioner. The asset tags are the lower-power devices whose positions are estimated by an IPS and verified by SecureAoX. The locators are anchor points positioned at fixed reference points that estimate the AoA towards the tracked asset tag's packets. The locators are typically installed with a power supply and constant network connection at a fixed position. Finally, the positioner is a service that estimates positions based on the angles provided by the locators. The positioner service can run on any server or in the cloud. The SecureAoX location verification algorithm is running as a component on the positioner. Furthermore, the positioner orchestrates communication between the locators and the asset tags to avoid duplicate transmissions.

Figure 7.1 shows a schematic overview of the architecture. MQTT is used for the information exchange between the locators and the positioner. Besides

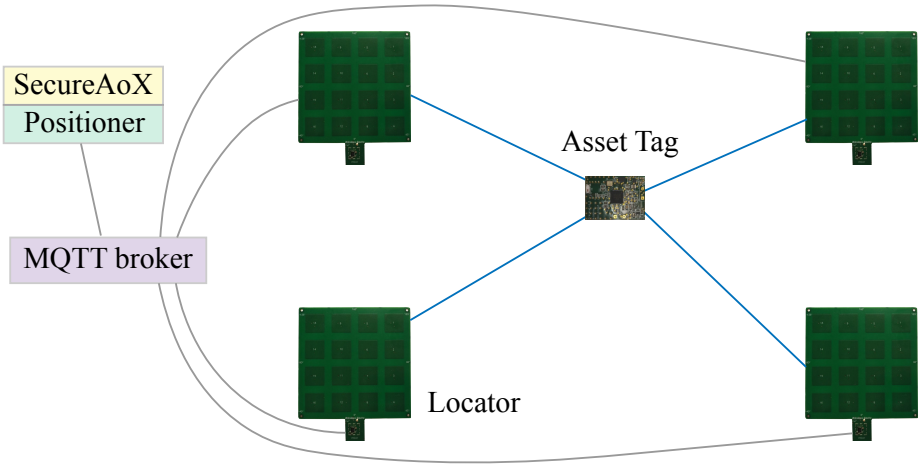


FIGURE 7.1: Architecture of SecureAoX [85] © 2022 IFIP

the AoX functionality, BLE is also used to exchange information between the locators and the asset tags.

7.2.2 Independent Location Verification Information

The Bluetooth CTE is appended after the CRC of BLE packets. Therefore, an attacker could replay packets with enabled CTE at different locations to manipulate positions without modification of data. Consequently, the asset tags must be authenticated to attribute each position to a specific asset tag. SecureAoX uses the same challenge-response scheme to authenticate the CTE enabled advertisements as used for the APS proposed in Chapter 6. However, in contrast to the APS in SecureAoX, all locators send the update packets with an enabled CTE for AoD positioning, and the asset tags measure the In-phase and Quadrature component (IQ)-samples of these CTEs. The in-phase and quadrature components of a complex signal are used to represent the amplitude and phase of a signal in a single parameter. The IQ-samples are practical to measure the phase shift between the antennas of an array.

Figure 7.2 shows the updated process for updating the challenge on asset tags and measuring the independent location verification information. When the asset tag has received an update packet and updated its challenge, it reads the IQ sample of the CTE of this update packet. The IQ-samples are forwarded to a designated locator using a BLE characteristic. The positioner elects for

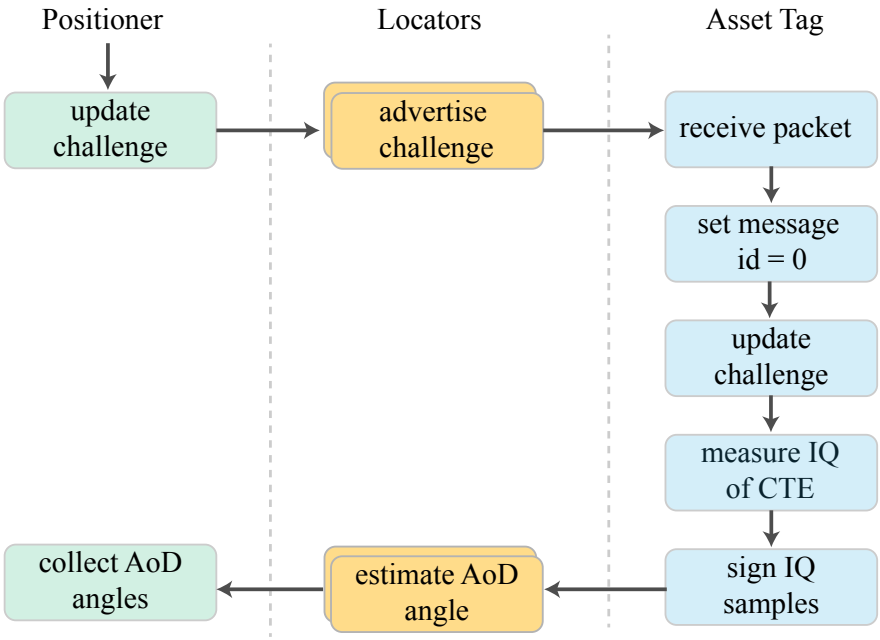


FIGURE 7.2: Process to update a challenge and measure independent location verification information [85] © 2022 IFIP

each asset tag a designated locator to limit connections and save energy. The designated locator estimates the angles based on the IQ parameters received from the asset tag. Calculating the angles on the locators has the advantage that it saves energy on the energy-constrained asset tags. Finally, the locator forwards the angles to the positioner, which uses the AoD angles to verify the AoA angles.

When the asset tag receives a new challenge, the asset tag starts broadcasting extended advertisement packets with enabled AoA CTE periodically. Figure 7.3 shows the process of broadcasting the AoA-enabled advertisement. For SecureAoX, the AoA position advertisements are similar to the position advertisements in the APS proposed in Chapter 6. However, when the positioner has estimated the position and the position is the first position after a challenge update, it compares the AoA position with the independently measured AoD location verification information to detect attacks.

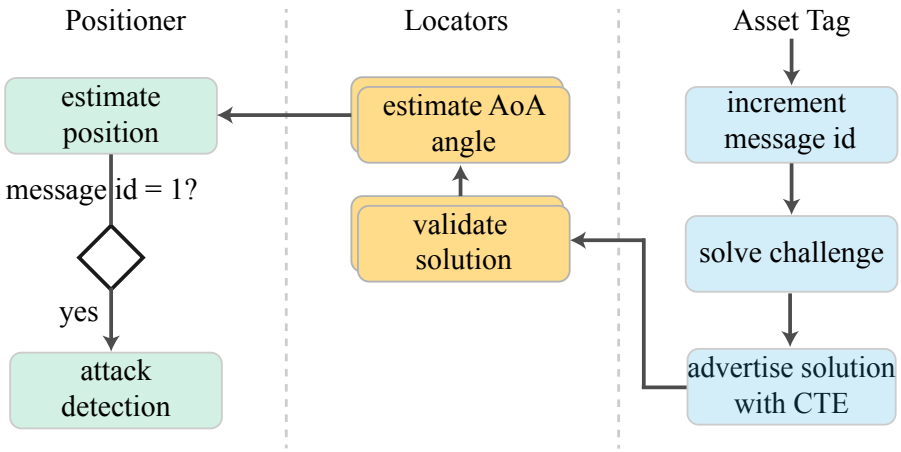


FIGURE 7.3: Process to advertise AoA and location verification [85] © 2022 IFIP

7.2.3 Detecting Manipulated Positions

SecureAoX uses two independently measured angle values to detect attacks. The AoAs measured by the locators are the claimed location information of the IPS that the LVS must verify. The asset tag measures the AoDs to verify these claims. The locators and asset tag signatures assert that the designated entities have measured a given angle and that the data was not manipulated.

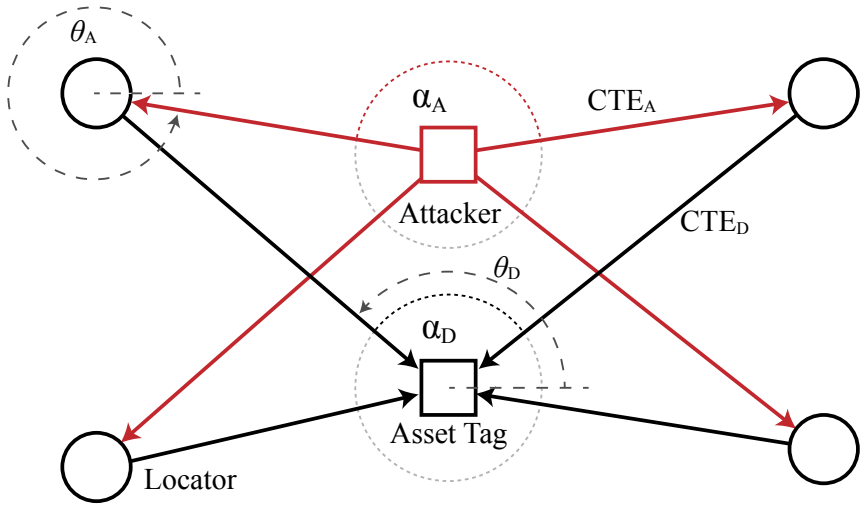


FIGURE 7.4: Contradiction in AoA and AoD angles [85] © 2022 IFIP

For a honest asset tag, the difference between the AoAs and AoDs should be within the positioning system's accuracy. However, when an attacker tries to

manipulate a honest asset tag's position by replaying its AoA-enabled advertisement packets from a location that is considerably far from the victim asset tag, then the difference between the AoAs and AoDs is larger than the system's tolerances. Figure 7.4 shows an example where the AoAs are different from the AoDs. The measured angles θ are relative to the device, and the angles α are the angles between the straight lines from the asset tag to the locators.

In contrast to the angles θ_{D}, the angles α_{D} are independent of the asset tags orientation. Since the source of all CTE-enabled packets is known, the angles α_{X} can be calculated from the angles θ_{X}. $\forall i \in \{1, \ldots, n-1\} : \alpha_D^i = |\theta_D^i - \theta_D^{i+1}|$ and $\alpha_A^i = |\theta_A^i - \theta_A^{i+1}|$ where n is the number of locators, $\boldsymbol{\alpha}_A = (\alpha_A^1, \ldots, \alpha_A^{n-1})$ and $\boldsymbol{\alpha}_D = (\alpha_D^1, \ldots, \alpha_D^{n-1})$. Finally, Equation 7.1 calculates the error between the claimed angles α_{A} and the verification angles α_{D}.

$$\varepsilon(\alpha_{\mathrm{A}}, \alpha_{\mathrm{D}}) = \frac{1}{n} \sum_{i \in \{1,\ldots,n\}} |\alpha_{\mathrm{A}}^i - \alpha_{\mathrm{D}}^i| \tag{7.1}$$

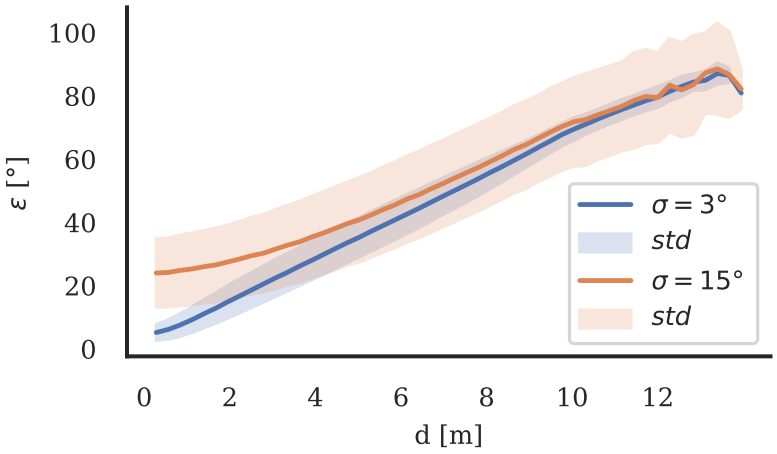


FIGURE 7.5: Error ε between the AoA computed by the anchors and the AoD provided by the tag (or the attacker) to the verifier. [85] © 2022 IFIP

The error between α_{A} and α_{D} increases with the distance d between the asset tag and the forged position. Figure 7.5 shows the results of a simulation

where the error in relation to distance d between attacker and asset tag is simulated for two assumed uncertainties of the positioning systems. The dependency on error and distance enables a threshold-based classifier to detect attacks. Equation 7.2 shows a classifier based on the sigmoid activation function $S(x) = \frac{1}{1+e^{-x}}$ and a threshold. This classifier applied on the result of Equation 7.1 is able to detect manipulated positions, as we will show in the evaluation.

$$th(x) = \begin{cases} 1, & \text{if } S(x) \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{7.2}$$

The optimal threshold can be found with a training set. Therefore, the Receiver Operating Characteristic (ROC) curve of the training set is calculated. And then, the Geometric Mean G-Mean $= \sqrt{precision \cdot recall}$ for all thresholds is calculated, and the threshold that results in the greatest G-Mean is chosen for the classification. With this approach, a threshold is selected that optimizes precision and recall.

7.3 Evaluation

7.3.1 Experiment Setup

We have implemented a simulator to evaluate the performance of our LVS. This simulator comprises the area of the IPS, the locators at fixed positions, and the tracked asset tags.

Figure 7.6 shows the scenario of an experiment. The simulator places one asset tag and one attacker with a uniform random distribution on a $10\,\text{m} \times 10\,\text{m}$ scenario containing $n = 4$ anchors placed at the corners. To simulate the AoX estimations, first, the ground truth angles ${\theta'}^i_\chi$ are calculated $\forall i \in \{1, \ldots, n\}$ for each anchor. Then for each true angle ${\theta'}^i_\chi$ the estimated angle $\theta^i_\chi = \mathcal{N}({\theta'}^i_\chi, \sigma^2)$ is simulated, where σ represents the measurement uncertainty of the locator. Applying multi-angulation on the simulated θ^i_χ returns a set of positions randomly distributed around the ground truth positions, as shown in the example in Figure 7.6.

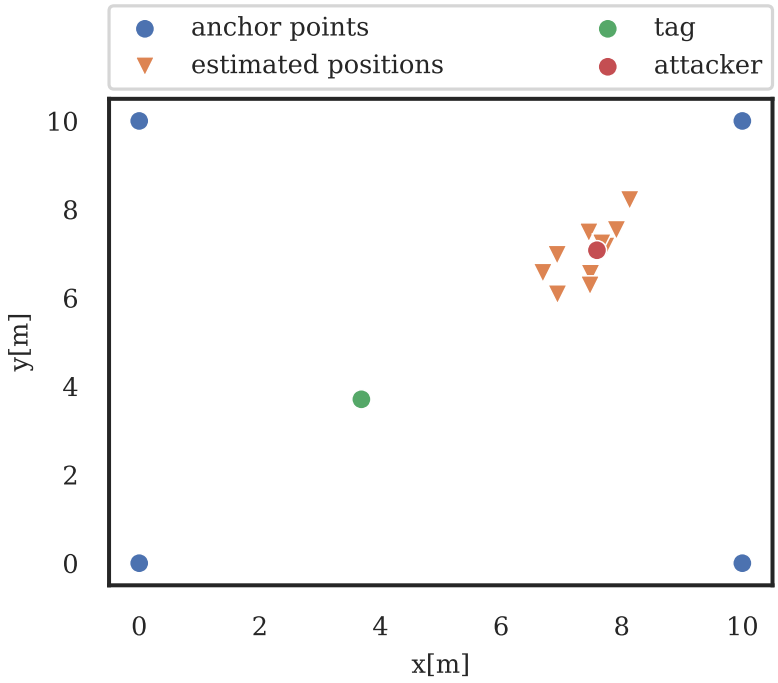


FIGURE 7.6: Example of simulated scenario with one attacker and 10 position estimations from the anchors. [85] © 2022 IFIP

The random distribution and the limited area of the scenario define the ground truth distance between asset tags and attackers. Figure 7.7 shows the distribution of the ground truth distance between the asset tag and the attacker. With the uniform distribution, attacker positions close to the asset tag are less likely due to the small area, and attacker positions far from the asset tag are less likely due to the limited scenario size. Nevertheless, assuming the attacker forges a position at a random position, this distribution also represents the likelihood that an attacker might forge a certain position.

The performance of an LVS is defined by how well it detects malicious positions. Since SecureAoX depends on the difference between the estimated AoA and the AoD, the classifier's performance depends on the accuracy of AoX estimation. Furthermore, this accuracy depends on the hardware used for direction-finding. For example, a positioning system with low-cost hardware and small antenna arrays could have lower accuracy than high-end systems with large antenna arrays. To build a meaningful model that can be used to derive statements for positioning systems with different accuracies, we evaluated

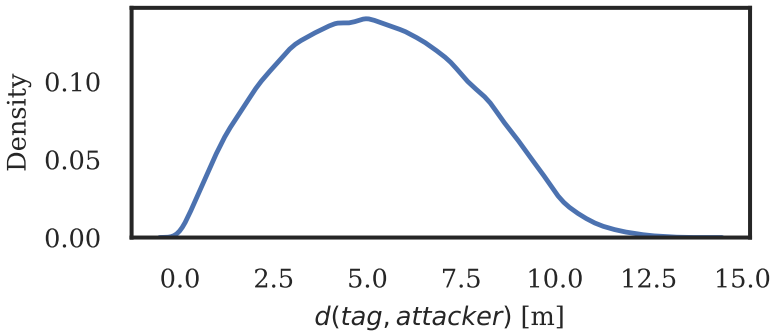


FIGURE 7.7: Attacker to asset tag distance distribution [85] © 2022 IFIP

SecureAoX assuming different degrees of accuracy. Therefore, we performed the simulation with varying standard deviation values σ, used to simulate the accuracy of direction-finding. For all evaluated values of σ, we conducted a Monte Carlo simulation with 10^6 draws. Half of the draws are attacked positions.

7.3.2 Performance Analysis

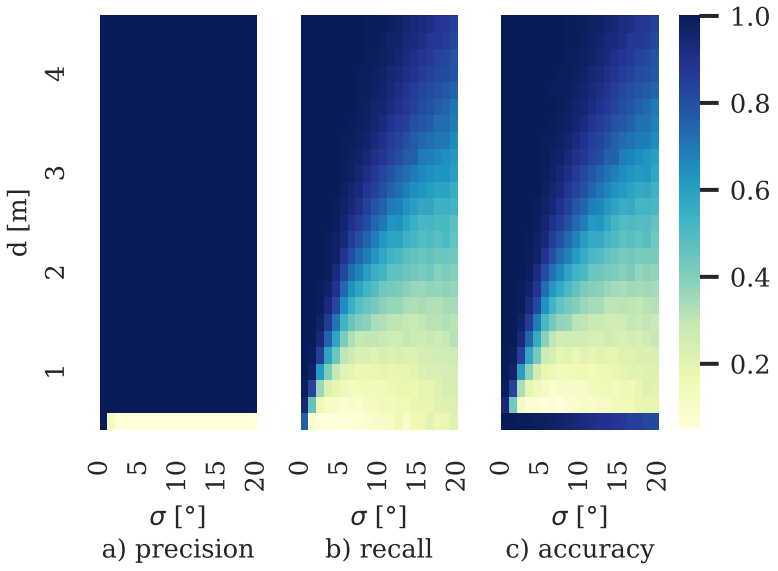


FIGURE 7.8: Results of the Monte Carlo simulation [85] © 2022 IFIP

The detection rate depends on the distance d between the asset tag and the attacker. When d is smaller than the accuracy of the positioning system, it is hard to detect attacks. Figure 7.8 shows the performance of SecureAoX according to precision, recall, and accuracy in relation to the distance and the positioning systems accuracy σ. Figure 7.8a shows the $precision = \frac{TP}{TP+FP}$ of the classifier. This experiment shows that attacks detected by SecureAoX are mostly true positive, except for attacks with small distances d. Figure 7.8b shows the $recall = \frac{TP}{TP+FN}$ of the classifier. This experiment shows that SecureAoX is able to detect most attacks, except for small distances d and high uncertainties σ of the positioning system. However, the area where SecureAoX has a lower detection rate is also within the uncertainty of the positioning system. Therefore, the attacker can only manipulate a position within the noise of the position measurements. Figure 7.8c shows the $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ of the classifier. This simulation shows that attacks close to the asset tag are not detected. Note that the accuracy for the distance $d = 0$ is high. This high accuracy is mainly the result of the distance distribution shown in Figure 7.7. Since 50% of the samples represent non-manipulated positions, and only a few attacks are launched close to the original position, the falsely negative classified attacks do not affect the accuracy close to $d = 0$ much. Rather, this shows that SecureAoX correctly classifies the not manipulated positions as non-attacks.

The Receiver Operating Characteristic (ROC) curve can be used to evaluate the performance of a classifier. Figure 7.9 shows the ROC curve for different σ values that characterize the accuracy of the angle measurements. The ROC curve shows that the classifier's performance depends on the accuracy of the used positioning system.

In this chapter, we built a model that supports evaluating diverse positioning systems with varying accuracies. Therefore, we simulated the angle measurements with multiple standard deviations $\sigma \in \{1, 2, ..., 20\}$. However, to assess these simulations in the context of reality, we measured the accuracy of a BLE-based AoA locator. For these measurements, we used the Silicon Labs PCB4185A Direction Finding Radio as a locator and a Silicon Labs Thunderboard as an asset tag. To measure the accuracy, we positioned the locator and measured the AoA to an asset tag at a well-defined angle and position. For this experiment, the asset tag and the locator were in the line of sight.

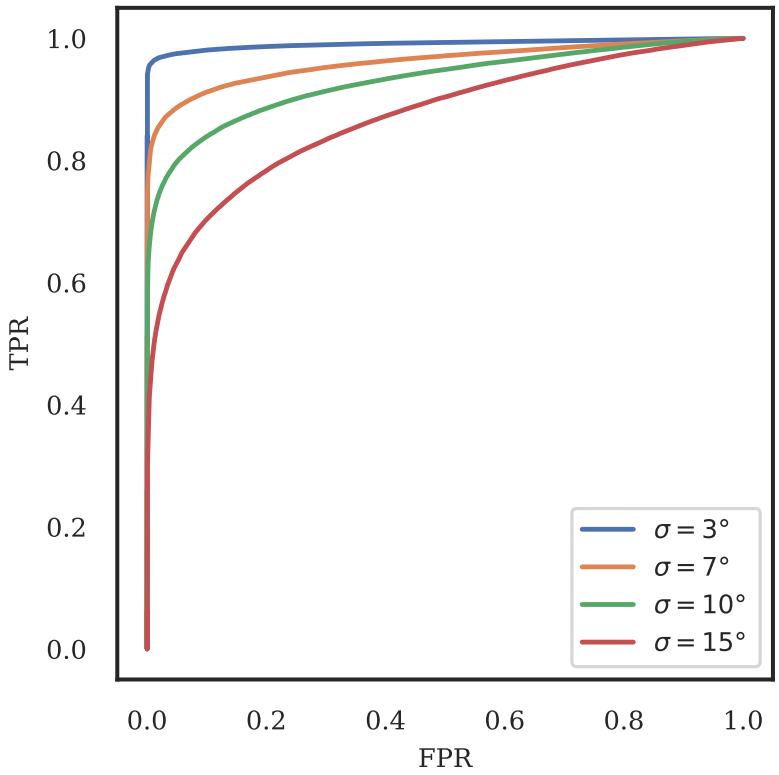


FIGURE 7.9: ROC curve for different accuracies [85] © 2022 IFIP

Figure 7.10 shows the angle error and the standard deviation for different distances after calibration of the locator. Overall, the standard deviation is $\sigma = 6.49°$. We used our model to assess the performance of a system with the same accuracy as we measured in this experiment. Figure 7.11 shows the accuracy in dependence of the distance using the realistic standard deviation $\sigma = 7°$.

Figure 7.12 shows the area under the ROC curve of different location verification algorithms. The area under the ROC curve can be used to evaluate the performance of different classifiers. The data for this analysis was measured out of the corresponding ROC curves found in the related work. Note that due to the different scenarios and architectures of the location verification systems, this comparison must be handled with caution. Nevertheless, the comparison shows how well the algorithms perform in their scenario.

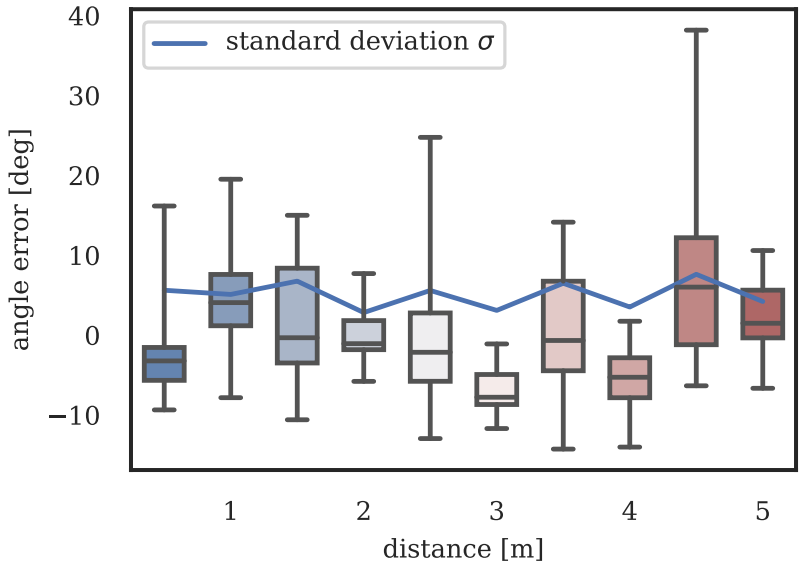


FIGURE 7.10: AoA Accuracy with LOS in relation to distance [85]

7.3.3 Security Analysis

The LVS should detect malicious positions and attacks aiming to forge or manipulate positions. This section analyzes the capabilities of SecureAoX to detect attacks.

We assume that all asset tags, locators, and positioners have secure, unclonable, and appropriate keys to create signatures. Furthermore, the locators and positioners have access to a PKI to validate signatures. Finally, for the security analysis, we also assume that the attack distance d is larger than the uncertainty of the used positioning system.

An adversary might try to forge, manipulate, or infer a position. An adversary could try to suppress a position by jamming the medium to interfere with the AoA or AoD-enabled advertisement packets. Since the medium cannot be protected from such unsolicited activities, it is impossible to prevent this attack. However, since each advertisement has a message id and the devices keep track of the latest ids, the devices can detect missing packets and can either request retransmission or raise an alert.

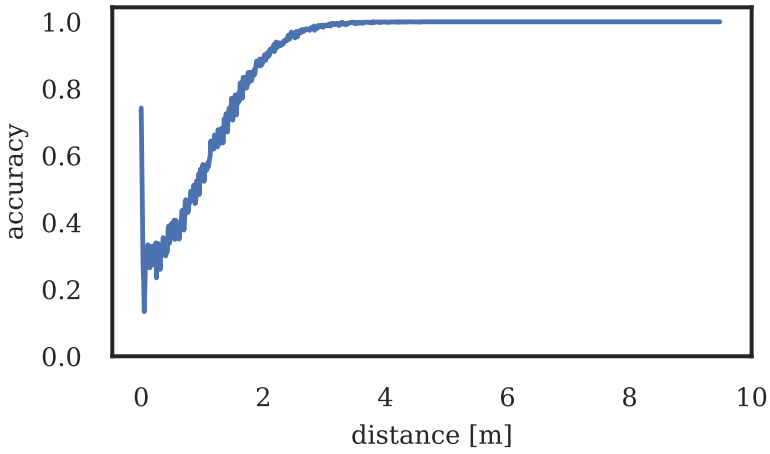


FIGURE 7.11: Accuracy over distance [85] © 2022 IFIP

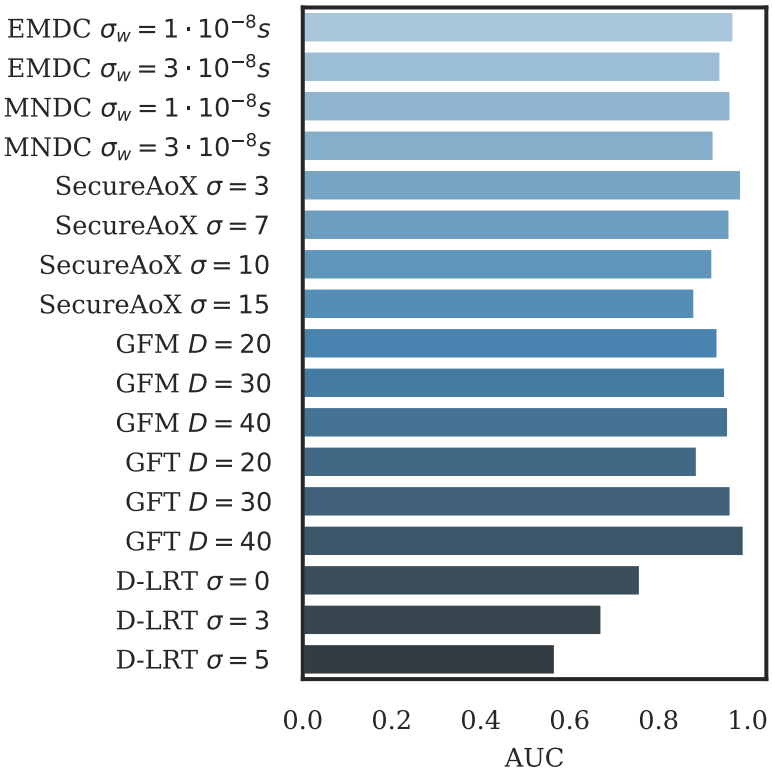


FIGURE 7.12: Area under the ROC curve [85] © 2022 IFIP

An attacker might perform a replay attack to manipulate a position. Let us assume an attacker reads a CTE-enabled advertisement packet and rebroadcasts it at a different location. Then the locators will receive two challenge solutions with the same message id. The locators ignore packets with the same message id for the same solution, and therefore, the replay attack fails.

An attacker might perform a wormhole attack to manipulate a position. Assume an attacker successfully jammed the CRC and CTE of the original transmission of an asset tag's CTE enabled advertisement and then rebroadcasted it at a different location. Then the locators would receive valid signed challenge responses with valid CTE signals of a different location. However, the asset tag receives the CTE-enabled advertisements from the locators, measures the IQ samples of the CW, signs, and sends the data with a notification to the subscribing locators. The locators validate the IQ samples' signature, calculate the AoD from this information, and forward it to the positioner, which uses the SecureAoX classifier to detect manipulations. Since the AoDs are measured between locators and asset tags and the AoAs are measured between attacker and locators, the AoAs are inconsistent with the AoDs. Consequently, the attack fails when the difference between the attacker and the asset tag is large enough to be detected by the classifier.

A limitation of SecrueAoX is that an attacker might succeed in cloning the scenario by not only jamming and replaying the AoA but also AoD packets. The attacker could clone the scenario at a different location with this attack. However, with SecureAoX, the α_{A} must match the α_{D}. Therefore, the cloned asset tag must have the same ratio to the cloned locators as the attacked asset tag to its locators. This requirement to preserve the ratio makes the attack complex and limits its use.

7.4 Conclusion

In this chapter, we proposed SecureAoX, a location verification system for Bluetooth AoA and AoD indoor positioning systems. SecureAoX uses the contradiction between AoA and AoD angles when a position was manipulated to verify positions.

We proposed a classifier that uses the error between the AoAs and AoDs to detect attacks against locations. For the evaluation of the classifier, we used a hybrid approach with a Monte Carlo analysis and real-world measurements. With a Monte Carlo simulation, we simulated the AoAs and AoDs for different asset tag and attacker locations with various assumptions about the measurement accuracy. We performed accuracy measurements in a real-world scenario to assess a realistic accuracy assumption. The evaluation shows that our classifier performs well under the condition of the AoA-based IPS accuracy. In the security analysis, we showed that the architecture of SecureAoX is secure and is able to respond to several attacks that try to manipulate positions.

The main limitation of the proposed LVS is that compared to the plain APS proposed in Chapter 6, the asset tag must also measure positions for position verification. These additional measurements will increase energy consumption. This elevated energy consumption should be considered when assessing the sufficiency of security provided by the APS or determining whether the incorporation of the LVS is necessary.

Chapter 8

Conclusions and Future Work

In this chapter, we first summarize the contributions of this thesis in Section 8.1. Then, we briefly discuss future research directions in Section 8.2.

8.1 Summary of Contributions

In this thesis, we investigated several aspects required to build a distributed audit trail for the IoT. Therefore, we mainly focused on how IoT data and meta-data of IoT devices can be attributed to their corresponding device identities and how this information can be made available to third parties in an auditable form. In this section, we summarize the contributions of this thesis in the order of their occurrence.

In Chapter 3, we proposed a novel DLT-based DPKI with a signature store. This DPKI comprises the GoT, to represent the trust declarations between the managed identity claims, and the ABCG to immutably store the transactions that build the GoT. For the distributed certification of the identity claims, we proposed to use reputation functions based on the trust declarations of the GoT, domain vetting, and domain scoring to estimate the reputation of identity claims. The evaluation showed that these reputation functions are able to attribute a higher reputation to honest identity claims than to attackers. Consequently, a reputation threshold can be used for the distributed certification of the authenticity of identity claims. As a DLT-based system, all permissioned nodes can contribute to the GoT by committing signed transactions to the ABCG. The ABCG assures that only valid transactions are secured, all

full nodes have a replica of the GoT, and that confirmed transactions cannot be altered or deleted. The ABCG is an application specific BlockDAG optimized for graph transactions. The evaluation of the ABCG showed that it is able to outperform general purpose DLTs in terms of transaction throughput. To enable distributed timestamps, we evaluated block discovery services based on the reputation provided by the DLT. These block discovery services can split the DLT into epochs. The GoT for each epoch can be recreated by loading all transactions until this epoch. Therefore, the ABCG enables audibility.

In Chapter 4, we investigated the applicability of the DPKI proposed in the previous chapter to manage the identities of IoT devices and how their data can be made auditable. To exchange IoT devices with the ABCG we have extended the data structure by statements. With the statements, low-power IoT devices can offload energy demanding DLT operations to an AFN. Furthermore, we designed a model to build the identity claims using hardware-accelerated cryptography of recent microcontrollers. Additionally, we introduced a pairing declaration to represent the relationship between IoT devices and their owning, respectively, operating entities. Finally, to evaluate the applicability of the proposed system, we implemented a real-world testbed. This real-world testbed comprises LoRaWAN and LTE Cat-M1-based custom IoT devices. The evaluation showed that recent off-the-shelf microcontrollers can handle all required cryptographic functions and can handle the statements required to handle DLT transactions. Furthermore, the evaluation showed that low-power networks are able to forward the statements comprising the signed transactions. Finally, the evaluation showed that the energy overhead introduced by securing the IoT device's identity and data exists but is affordable for quality-relevant applications requiring auditability.

In Chapter 5, we designed a DCCI to represent quality-related metadata on the GoT, and, therefore, making this information accessible for third-party auditors. With this information available, the auditors can audit the quality of IoT data, even when the data source is not one of their own devices. For quality-related applications, this quality information is as valuable as the authenticity of the data. To represent this quality-related information, we designed a GoT based model to represent signatures on DCCs and ACs. To enable these certificate signatures, we extended the existing declarations towards document

hashes with optional references to link the certified identity claims. In contrast to other calibration certificate infrastructures that only support the management of DCCs, the distributed approach of the GoT enables the integration of NASs, NMIs, and ACs. This has the advantage that auditors can also verify the signatures of ACs and the calibration laboratories responsible for these signatures. Furthermore, the integration of the NASs and NMIs in the DPKI has the advantage that these organizations typically are well known and have a high reputation. Therefore, these organizations act as natural trust anchors and can help bootstrap reputation in the DPKI. Additionally, we designed a multi-signature and approval model for the GoT. To evaluate the proposed extensions of the GoT, we implemented the changes in our real-world testbed. The evaluation showed that the validation of signatures on calibration certificates, calibration laboratories, and multi-signatures is fast, even when the GoT is large. Furthermore, the evaluation showed that the GoT could be split into multiple side-chains enabling scalability and privacy.

In Chapter 6, we discussed how positions could be secured in a distributed audit trail. Therefore, we designed and evaluated the APS. To make the positions auditable, the APS performs a PoL and secures these location proofs on the GoT. Our proposed data model to secure the PoL on the GoT includes the raw data from the asset tags, the intermediate angle or distance estimations of the locators (i.e., anchors), and the final position estimated by the positioner. This linking between raw and processed data enables data provenance and, combined with secure and attributed storage, auditability. Furthermore, the auditability of a position depends on the tracked asset's maximum velocity and the positions' sampling rate. Therefore, we built a mathematical model to define the conditions required to achieve auditability. Finally, we implemented a real-world testbed to evaluate the proposed APS. This real-world testbed is built on a BLE-based AoA positioning system. In the evaluation, we investigated the impact of adding auditability on energy consumption and disk storage requirements. The evaluation showed that auditable positioning requires additional energy to perform the PoL on the asset tag. However, the evaluation also showed that adjusting the sampling rate to the minimum frequency that meets the auditability requirements optimizes the required energy. Consequently, the additional energy required to enable auditability is affordable

for quality-relevant applications.

In Chapter 7, we addressed a vulnerability on the physical layer that enables attackers to manipulate positions in BLE-based AoA positioning systems and cannot be prevented by cryptographic systems like the APS. To overcome this issue, we designed SecureAoX a LVS that uses AoD as independent information to verify the position claims of AoA positioning. Two position estimations can be done by measuring the AoAs on the locators and the AoDs on the asset tag. SecureAoX uses the distance between these two independently measured positions for manipulation detection. When this distance is larger than the accuracy of the used PS, manipulation of the position is likely. We performed a Monte Carlo simulation to evaluate the proposed LVS. Since the different PSs have various accuracies, we performed the simulation for a range of accuracies. Furthermore, to assess SecureAoX in a realistic scenario, we measured the accuracy of a real-world BLE-based AoA positioning system. The simulations showed that the system could detect manipulated positions further apart from the original position than the accuracy of the used PS. Consequently, a positioner can perform this attack detection for each position and drop a position when manipulation is detected.

The distributed audit trail comprises several elements introduced in the different chapters of this thesis. As an overview, Figure 8.1 provides an example GoT containing the most important elements of our proposed distributed audit trail.

In the distributed audit trail, the transactions represent the audit records. Each transaction has a timestamp that enables the auditor to recreate a replica of the GoT's state at any time in history. This recreation of the GoT's state enables the auditor to scrutinize the state of the monitored system at any time in the past. Signatures on each transaction and a DPKI attribute the audit records to the entity that has created the record. The DPKI uses a combined model to certify the authenticity of identity claims. A distributed model based on trust declarations, domain vetting, scoring, and reputation is used to certify the authenticity of organizations' identity claims. Furthermore, a CA like certification model using pairing declarations can be used to certify the authenticity of identity claims of departments and IoT devices.

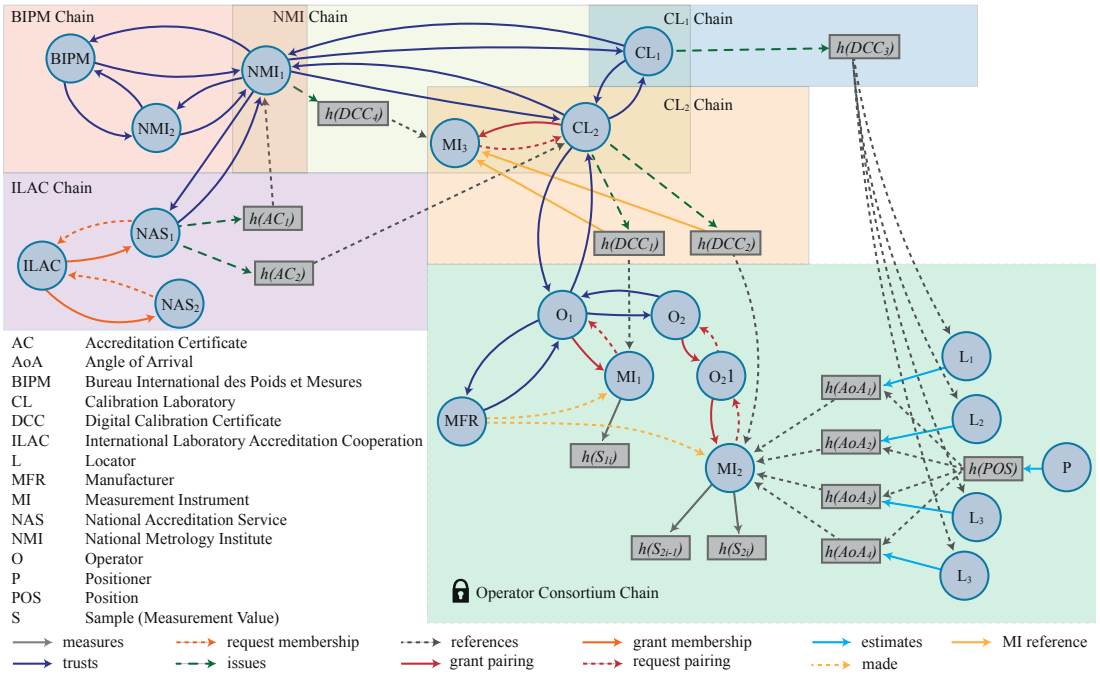


FIGURE 8.1: Example GoT with the most important elements

Accreditation and calibration certificates enable auditable metadata about the quality of the referenced CLs and MIs. Additionally, the MIs managed on the GoT can directly secure measurements using measures declarations. Finally, a position trace can directly be linked to MIs for applications that require auditable positions. This enables the auditor to verify when audit records signed by the MI were created.

With these components, the distributed audit trial can securely document all required details of an audit record, such as who created the record, what action or observation is documented with the record, when the record was created, and where the documented action or observation took place.

In summary, the results of this thesis cover the key components required to build a distributed audit trail for the IoT. The Veritaa framework stands out from related work primarily due to the combination of DPKI and audit records in the GoT. In the context of the IoT, both the DPKI and the audit trail benefit from this combination. In the IoT, a certified public key does not provide any information about the accuracy of the corresponding IoT device. Therefore, the DPKI for IoT devices benefits when third-party data consumers can access

auditable quality-related meta-data like calibrations and use this information to reason about the quality of the received data. Furthermore, since each audit record requires the signature of the entity that created the record, the audit trail benefits from the integration of a DPKI. Integrating the audit trail in the GoT has some key advantages. First, the graph model represents the relations between entities well and enables the use of well-researched and performant reputation functions for the distributed certification of a DPKI. Next, recent graph databases enable graph traversal algorithms mainly used to verify GoT objects. These traversal algorithms are typically fast since the traverser can follow the edges of the graph. Finally, the graph model makes the audit trail easily readable and explorable for auditors. This readability is mainly due to the representation of the real-world relationships between objects. In conclusion, the graph-based model has several advantages as a distributed audit trail and, therefore, is worth considering in future work.

8.2 Future Work

In this thesis, we investigated the components required to build a DLT-based distributed audit trail. Auditors can use this audit trail to verify if certain IoT applications comply with relevant guidelines, standards, or individual agreements between entities. Currently, these auditors are typically humans interpreting the audit trail. However, with machine-readable audit trails, like the Veritaa framework presented in this thesis, available, any oracle with the ability to decide if the events recorded in the audit trail comply with certain standards could act as an auditor. Therefore, also artificial intelligence could be trained as an auditor. Building an artificial intelligence-based auditor raises several research questions and opens new research opportunities.

The distributed audit trail designed and evaluated in this thesis does not only enable auditors to check the compliance of IoT applications. It also enables third-party data consumers to verify the received data's authenticity, integrity, and quality. A future direction of the Veritaa framework could be the integration into emerging data marketplaces or open-data platforms to make the quality of real-time sensor data auditable and data provenance visible. Verifiable

data provenance could facilitate federated and shared IoT infrastructure improving the sustainability and the total cost of the infrastructure because several stakeholders are not building the same infrastructure next to each other. Future research should accompany this integration process and monitor the benefits of auditability in this field.

This thesis mainly discussed a static view of the GoT, the distributed audit trail, at a given time for the reputation calculation. However, the Veritaa framework's ability to recreate the GoT's state at any time in history would also enable dynamic analysis of the GoT. For example, future research could investigate how the evolution of trust declarations towards an identity claim influences its reputation and if the development of the reputation over time could be used to improve distributed certification of identity claims.

This thesis showed how trust, secured round trip time measurements, and time synchronization declarations can be used to create auditable timestamps for the IoT. However, with a high enough sensor density, multiple sensors might observe the same event. Distributed sensing could be used to build consensus about measurement values and timestamps, similar to how the block discovery services built consensus about time on the ABCG. However, since the sensing range is not congruent with the radio transmission range, it is hard to tell if two sensors have measured the same or two independent events. Therefore, future work could focus on such distributed sensing and how to represent it on distributed audit trails like the one presented in this thesis.

Bibliography

[1] E. M. Navarro, A. N. Ramos Álvarez, and F. I. Soler Anguiano, "A new telesurgery generation supported by 5G technology: Benefits and future trends," in *3rd International Conference on Industry 4.0 and Smart Manufacturing (ISM)*, vol. 200, Jan. 2022, pp. 31–38. DOI: 10.1016/j.procs.2022.01.202.

[2] *Coke machine history*, 1990. [Online]. Available: http://www.cs.cmu.edu/~coke/coke.history.txt (visited on 12/04/2022).

[3] J. Teicher, *The little-known story of the first IoT device*, Feb. 2018. [Online]. Available: https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/ (visited on 12/04/2022).

[4] W. E. Zhang, Q. Z. Sheng, A. Mahmood, *et al.*, "The 10 Research Topics in the Internet of Things," in *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, Dec. 2020, pp. 34–43. DOI: 10.1109/CIC50333.2020.00015.

[5] T. Winter, P. Thubert, A. Brandt, *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.," *rfc*, vol. 6550, pp. 1–157, 2012.

[6] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, ISSN: 0743-166X, Apr. 2015, pp. 513–521. DOI: 10.1109/INFOCOM.2015.7218418.

[7] J. Schaerer, Z. Zhao, J. Carrera, S. Zumbrunn, and T. Braun, "SDNWisebed: A Software-Defined WSN Testbed," in *Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW)*, M. R. Palattella, S. Scanzio, and S. Coleri Ergen, Eds., Series Title: Lecture Notes in Computer Science, vol. 11803, Cham: Springer International Publishing, 2019, pp. 317–329,

ISBN: 978-3-030-31830-7 978-3-030-31831-4. DOI: 10.1007/978-3-030-31831-4_22.

[8] J. Schaerer, Z. Zhao, and T. Braun, "DTARP: A Dynamic Traffic Aware Routing Protocol for Wireless Sensor Networks," in *Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks (RealWSN)*, Shenzhen China: ACM, Nov. 2018, pp. 49–54. DOI: 10.1145/3277883.3277885.

[9] "Ericsson Mobility Report November 2022," Tech. Rep., 2022. [Online]. Available: https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf (visited on 12/04/2022).

[10] Lionel Sujay Vailshery, *IoT connected devices worldwide 2019-2030*, Jul. 2022. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (visited on 11/13/2022).

[11] *The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge*. [Online]. Available: http://opendefinition.org/ (visited on 12/15/2022).

[12] *Opendata.swiss*. [Online]. Available: https://opendata.swiss/de/ (visited on 12/12/2022).

[13] *The official portal for European data | data.europa.eu*. [Online]. Available: https://data.europa.eu/en (visited on 12/31/2022).

[14] *Open Government*. [Online]. Available: https://www.data.gov/open-gov/ (visited on 12/31/2022).

[15] *Data Marketplace | IOTA Wiki*. [Online]. Available: https://wiki.iota.org/blueprints/data-marketplace/overview (visited on 08/05/2022).

[16] *Free flow of non-personal data | Shaping Europe's digital future*. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/non-personal-data (visited on 12/31/2022).

[17] Bank für Internationalen Zahlungsausgleich, *A glossary of terms used in payments and settlement systems*. BIS, Committee on Payment and Settlement Systems, 2001, ISBN: 92-9197-133-2. [Online]. Available: https://www.bis.org/cpmi/glossary_030301.pdf (visited on 11/13/2022).

[18] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Tech. Rep., 2009, . Accessed November 24, 2019. [Online]. Available: https://bitcoin.org/bitcoin.pdf (visited on 11/24/2019).

[19] D. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Tech. Rep., 2014. [Online]. Available: http://gavwood.com/Paper.pdf (visited on 11/25/2019).

[20] E. Androulaki, A. Barger, V. Bortnikov, *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys Conference (EuroSys)*, Porto Portugal: ACM, Apr. 2018, pp. 1–15, ISBN: 978-1-4503-5584-1. DOI: 10.1145/3190508.3190538.

[21] S. Popov, "The Tangle," Tech. Rep., Apr. 2019. [Online]. Available: https://iota.org/IOTAWhitepaper.pdf (visited on 11/18/2019).

[22] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling Blockchain via Full Sharding," in *Proceedings of the 2018 ACM Conference on Computer and Communications Security (SIGSAC)*, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 931–948, ISBN: 978-1-4503-5693-0. DOI: 10.1145/3243734.3243853.

[23] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019. DOI: 10.1109/ACCESS.2019.2925010.

[24] A. de Vries, "Renewable Energy Will Not Solve Bitcoin's Sustainability Problem," *Joule*, vol. 3, no. 4, pp. 893–898, Apr. 2019, ISSN: 2542-4351. DOI: 10.1016/j.joule.2019.02.007.

[25] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion in Environmental Sustainability*, vol. 28, pp. 1–9, Oct. 2017, ISSN: 18773435. DOI: 10.1016/j.cosust.2017.04.011.

[26] D. Malone and K. O'Dwyer, "Bitcoin Mining and its Energy Footprint," in *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland*

International Conference on Information and Communities Technologies (ISSC/CIICT), Limerick, Ireland, 2014, pp. 280–285, ISBN: 978-1-84919-924-7. DOI: 10.1049/cp.2014.0699.

[27] C. Stoll, L. Klaaßen, and U. Gallersdörfer, "The Carbon Footprint of Bitcoin," *Joule*, vol. 3, no. 7, pp. 1647–1661, Jul. 2019, ISSN: 25424351. DOI: 10.1016/j.joule.2019.05.012.

[28] K. Croman, C. Decker, I. Eyal, *et al.*, "On Scaling Decentralized Blockchains," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds., Berlin, Heidelberg: Springer, 2016, pp. 106–125, ISBN: 978-3-662-53357-4. DOI: 10.1007/978-3-662-53357-4_8.

[29] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," vol. 19, no. 1, 2012.

[30] *Proof-of-stake (PoS)*, Jan. 2023. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/ (visited on 01/23/2023).

[31] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019. DOI: 10.1109/COMST.2018.2886932.

[32] S. Popov, H. Moog, D. Camargo, *et al.*, "The coordicide," Tech. Rep., Jan. 2020. [Online]. Available: https://files.iota.org/papers/20200120_Coordicide_WP.pdf.

[33] D. L. Baird, M. Harmon, and P. Madsen, "Hedera: A Public Hashgraph Network & Governing Council," Tech. Rep., Aug. 2020. [Online]. Available: https://hedera.com/hh_whitepaper_v2.1-20200815.pdf (visited on 01/23/2023).

[34] J. Neumann, J. Nordholz, M. Dohlus, and M. Nischwitz, *European Metrology Cloud – WP 1 Year 2: Node Rollout*.

[35] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on Hyperledger Fabric with secure multiparty computation," *IBM Journal of Research and Development*, vol. 63, no. 2/3, 3:1–3:8, Mar. 2019. DOI: 10.1147/JRD.2019.2913621.

[36] C. A. Hornbuckle, "Fractional-N synthesized chirp generator," pat. US7791415B2, Sep. 2010. [Online]. Available: https://patents.google.com/patent/US7791415B2/en (visited on 12/31/2022).

[37] O. B. A. Seller, "Wireless communication method," pat. US20160094269A1, Mar. 2016. [Online]. Available: https://patents.google.com/patent/US20160094269A1/en (visited on 12/31/2022).

[38] Semtech Corporation, "LoRa and LoRaWAN - A Tech Overview," Tech. Rep., Feb. 2020. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf (visited on 12/28/2022).

[39] LoRa Alliance, *The Things Network*. [Online]. Available: https://www.thethingsnetwork.org/ (visited on 12/15/2022).

[40] A. Haleem, A. Allen, A. Thompson, M. Nijdam, and R. Garg, "A Decentralized Wireless Network," Tech. Rep., Nov. 2018. [Online]. Available: http://whitepaper.helium.com/ (visited on 10/22/2022).

[41] *The Things Network Map*. [Online]. Available: https://www.thethingsnetwork.org/map (visited on 02/01/2023).

[42] *Helium Explorer*. [Online]. Available: https://explorer.helium.com (visited on 02/01/2023).

[43] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *2016 IEEE Wireless Communications and Networking Conference (WCNC)*, Doha, Qatar: IEEE, Apr. 2016, pp. 1–5. DOI: 10.1109/WCNC.2016.7564708.

[44] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, and A. Ghosh, "LTE-M Evolution Towards 5G Massive MTC," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Singapore: IEEE, Dec. 2017, pp. 1–6. DOI: 10.1109/GLOCOMW.2017.8269112.

[45] A. Hoglund, D. P. Van, T. Tirronen, O. Liberg, Y. Sui, and E. A. Yavuz, "3GPP Release 15 Early Data Transmission," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 90–96, Jun. 2018. DOI: 10.1109/MCOMSTD.2018.1800002.

[46] Z. Amjad, A. Sikora, J.-P. Lauffenburger, and B. Hilt, "Latency Reduction in Narrowband 4G LTE Networks," in *15th International Symposium on Wireless Communication Systems (ISWCS)*, Lisbon: IEEE, Aug. 2018, pp. 1–5. DOI: 10.1109/ISWCS.2018.8491085.

[47] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas, "Internet X.509 Public Key Infrastructure: Certification Path Building," RFC 4158, Sep. 2005, p. 81. [Online]. Available: https://www.rfc-editor.org/info/rfc4158 (visited on 01/31/2021).

[48] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, "A blockchain-based PKI management framework," in *2018 IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–6, ISBN: 978-1-5386-3416-5. DOI: 10.1109/NOMS.2018.8406325.

[49] P. Eckersley and J. Burns, *The EFF SSL Observatory*, Aug. 2010. [Online]. Available: https://www.eff.org/observatory (visited on 12/09/2019).

[50] *Cert Spotter - Timeline of PKI Security Failures*. [Online]. Available: https://sslmate.com/certspotter/failures (visited on 12/07/2019).

[51] J. R. Prins, "DigiNotar certificate authority breach 'Operation black tulip'," Tech. Rep., Sep. 2011.

[52] P. Zimmermann, *The official PGP user's guide*. Cambridge, Mass: MIT Press, 1995, ISBN: 978-0-262-74017-3.

[53] U. Maurer, "Modelling a public-key infrastructure," in *Computer Security (ESORICS)*, E. Bertino, H. Kurth, G. Martella, and E. Montolivo, Eds., Berlin, Heidelberg: Springer, 1996, pp. 325–350, ISBN: 978-3-540-70675-5. DOI: 10.1007/3-540-61770-1_45.

[54] G. Caronni, "Walking the Web of trust," in *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE)*, Gaithersburg, MD, USA: IEEE Computer Society, 2000, pp. 153–158, ISBN: 978-0-7695-0798-9. DOI: 10.1109/ENABL.2000.883720.

[55] T. Hepp, F. Spaeh, A. Schoenhals, P. Ehret, and B. Gipp, "Exploring Potentials and Challenges of Blockchain-based Public Key Infrastructures," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France: IEEE, Apr. 2019, pp. 847–852. DOI: 10.1109/INFCOMW.2019.8845169.

[56] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: A new PKI model with Certificate Transparency based on blockchain," *Computers & Security*, vol. 85, pp. 333–352, Aug. 2019, ISSN: 01674048. DOI: 10.1016/j.cose.2019.05.013.

[57] A. Papageorgiou, A. Mygiakis, K. Loupos, and T. Krousarlis, "DPKI: A Blockchain-Based Decentralized Public Key Infrastructure System," in *2020 Global Internet of Things Summit (GIoTS)*, Dublin, Ireland: IEEE, Jun. 2020, pp. 1–5, ISBN: 978-1-72816-728-2. DOI: 10.1109/GIOTS49054.2020.9119673.

[58] S. Yao, J. Chen, K. He, R. Du, T. Zhu, and X. Chen, "PBCert: Privacy-Preserving Blockchain-Based Certificate Status Validation Toward Mass Storage Management," *IEEE Access*, vol. 7, pp. 6117–6128, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2889898.

[59] D. Wilson and G. Ateniese, "From Pretty Good to Great: Enhancing PGP Using Bitcoin and the Blockchain," in *Network and System Security (NSS)*, M. Qiu, S. Xu, M. Yung, and H. Zhang, Eds., Cham: Springer International Publishing, 2015, pp. 368–375, ISBN: 978-3-319-25645-0. DOI: 10.1007/978-3-319-25645-0_25.

[60] M. Al-Bassam, "SCPKI: A Smart Contract-based PKI and Identity System," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (CCS)*, Abu Dhabi United Arab Emirates: ACM, Apr. 2017, pp. 35–40, ISBN: 978-1-4503-4974-1. DOI: 10.1145/3055518.3055530.

[61] K. O'Hara and W. Hall, "Trust on the Web: Some Web Science Research Challenges," *Web Science*, vol. UOC Papers, no. Iss. 7. UOC, 2008, ISSN: 1885-1541.

[62] A. Singla and E. Bertino, "Blockchain-Based PKI Solutions for IoT," in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, Philadelphia, PA: IEEE, Oct. 2018, pp. 9–15, ISBN: 978-1-5386-9502-9. DOI: 10.1109/CIC.2018.00-45.

[63] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications: Workshops (PerCom Workshops)*, Kona, HI, Mar. 2017, pp. 618–623. DOI: 10.1109/PERCOMW.2017.7917634.

[64] M. A. Bouras, Q. Lu, S. Dhelim, and H. Ning, "A Lightweight Blockchain-Based IoT Identity Management Approach," *Future Internet*, vol. 13, no. 2, p. 24, Jan. 2021. DOI: 10.3390/fi13020024.

[65] A. S. Omar and O. Basir, "Identity Management in IoT Networks Using Blockchain and 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)*, Jul. 2018, pp. 994–1000. DOI: 10.1109/Cybermatics_2018.2018.00187.

[66] H. G. Semerjian, "NIST – PROMOTING INNOVATION, COMPETITIVENESS AND FACILITATING TRADE," in *Simposio de Metrología*, 2006, p. 7.

[67] "The International System of Units (SI)," Bureau International des Poids et Mesures, Tech. Rep., 2019.

[68] J. Schaerer and T. Braun, "A Distributed Calibration Certificate Infrastructure," in *2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, Paris, France: IEEE, Sep. 2022, pp. 1–4. DOI: 10.1109/BRAINS55737.2022.9909437.

[69] M. S. Gadelrab and R. A. Abouhogail, "Towards a new generation of digital calibration certificate: Analysis and survey," *Measurement*, vol. 181, 2021. DOI: 10.1016/j.measurement.2021.109611.

[70] B. Acko, H. Weber, D. Hutzschenreuter, and I. Smith, "Communication and validation of metrological smart data in IoT-networks," *Advances in Production Engineering & Management*, vol. 15, no. 1, pp. 107–117, Mar. 2020. DOI: 10.14743/apem2020.1.353.

[71] S. G. Hackel, F. Härtig, J. Hornig, and T. Wiedenhöfer, "The Digital Calibration Certificate," *PTB-Mitteilungen. Volume 127*, vol. Issue 4, 2017. DOI: 10.7795/310.20170403.

[72] *DCC*. [Online]. Available: https://www.ptb.de/dcc/ (visited on 03/14/2023).

[73] *Examples of the Digital Calibration Certificate*. [Online]. Available: https://dccwiki.ptb.de/en/gp_home (visited on 04/23/2023).

[74] F. Thiel and J. Wetzlich, "The European Metrology Cloud: Impact of European Regulations on Data Protection and the Free Flow of Non-Personal Data," in *19th International Congress of Metrology (CIM2019)*, EDP Sciences, 2019. DOI: 10.1051/metrology/201901001.

[75] D. Peters, J. Wetzlich, F. Thiel, and J.-P. Seifert, "Blockchain applications for legal metrology," in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2018, pp. 1–6. DOI: 10.1109/I2MTC.2018.8409668.

[76] T. Takatsuji, H. Watanabe, and Y. Yamashita, "Blockchain technology to visualize the metrological traceability," *Precision Engineering*, vol. 58, 2019. DOI: 10.1016/j.precisioneng.2019.04.016.

[77] T. Mustapää, P. Nikander, D. Hutzschenreuter, and R. Viitala, "Metrological Challenges in Collaborative Sensing: Applicability of Digital Calibration Certificates," *Sensors*, vol. 20, no. 17, 2020. DOI: 10.3390/s20174730.

[78] R. Shah, M. McIntee, S. Nagaraja, S. Bhandary, P. Arote, and J. Kuri, "Secure Calibration for Safety-Critical IoT: Traceability for Safety Resilience," Feb. 2020, arXiv: 1908.00740. [Online]. Available: http://arxiv.org/abs/1908.00740 (visited on 12/31/2021).

[79] W. S. Melo, A. Bessani, and L. F. R. C. Carmo, "How blockchains can help legal metrology," in *18th International Middleware Conference (Middleware)*, Las Vegas Nevada: ACM, Dec. 2017, pp. 1–2. DOI: 10.1145/3152824.3152829.

[80] W. S. Melo, A. Bessani, N. Neves, A. O. Santin, and L. F. R. C. Carmo, "Using Blockchains to Implement Distributed Measuring Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 5, pp. 1503–1514, May 2019, Conference Name: IEEE Transactions on Instrumentation and Measurement. DOI: 10.1109/TIM.2019.2898013.

[81] W. Melo, R. C. S. Machado, D. Peters, and M. Moni, "Public-Key Infrastructure for Smart Meters using Blockchains," in *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*, Jun. 2020, pp. 429–434. DOI: 10.1109/MetroInd4.0IoT48571.2020.9138246.

[82] M. Peterek and B. Montavon, "Prototype for dual digital traceability of metrology data using X.509 and IOTA," *CIRP Annals*, vol. 69, no. 1, pp. 449–452, 2020. DOI: 10.1016/j.cirp.2020.04.104.

[83] A. Yassin, Y. Nasser, M. Awad, *et al.*, "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017. DOI: 10.1109/COMST.2016.2632427.

[84] F. Che, A. Ahmed, Q. Z. Ahmed, S. A. R. Zaidi, and M. Z. Shakir, "Machine Learning Based Approach for Indoor Localization Using Ultra-Wide Bandwidth (UWB) System for Industrial Internet of Things (IIoT)," in *2020 International Conference on UK-China Emerging Technologies (UCET)*, Aug. 2020, pp. 1–4. DOI: 10.1109/UCET51115.2020.9205352.

[85] J. Schaerer, A. Di Maio, and T. Braun, "SecureAoX: A Location Verification System," in *2022 14th IFIP Wireless and Mobile Networking Conference (WMNC)*, Sousse, Tunisia: IEEE, Oct. 2022, pp. 38–45. DOI: 10.23919/WMNC56391.2022.9954303.

[86] Martin Woolley, "Bluetooth Core Specification Version 5.1 Feature Overview," Tech. Rep., Jan. 2019. [Online]. Available: https://www.bluetooth.

com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/ (visited on 05/03/2022).

[87] *3 Ways for Android Pokemon Go Spoofing in 2023- Dr.Fone*. [Online]. Available: https://drfone.wondershare.com/virtual-location/pokemon-go-spoofing-android.html (visited on 01/22/2023).

[88] *Chinese GPS spoofing circles could hide Iran oil shipments*, Section: Featured Stories, Dec. 2019. [Online]. Available: https://www.gpsworld.com/chinese-gps-spoofing-circles-could-hide-iran-oil-shipments/ (visited on 01/22/2023).

[89] *Two years since the Tesla GPS hack*, Section: From the Magazine, Aug. 2021. [Online]. Available: https://www.gpsworld.com/two-years-since-the-tesla-gps-hack/ (visited on 01/22/2023).

[90] A. Zugenmaier, M. Kreutzer, and M. Kabatnik, "Enhancing applications with approved location stamps," in *IEEE Intelligent Network 2001 Workshop (IN)*, Boston, MA, USA: IEEE, 2001, pp. 140–147, ISBN: 978-0-7803-7047-0. DOI: 10.1109/INW.2001.915307.

[91] S. Saroiu and A. Wolman, "Enabling New Mobile Applications with Location Proofs," in *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile)*, event-place: Santa Cruz, California, New York, NY, USA: Association for Computing Machinery, 2009, ISBN: 978-1-60558-283-2. DOI: 10.1145/1514411.1514414.

[92] B. Pestourie, V. Beroulle, and N. Fourty, "Clock skew-based physical authentication protocol for 802.15.4 IR-UWB indoor positioning," in *Proceedings of the 10th International Conference on the Internet of Things (IoT)*, Malmö Sweden: ACM, Oct. 2020, pp. 1–8. DOI: 10.1145/3410992.3410994.

[93] M. Amoretti, G. Brambilla, F. Medioli, and F. Zanichelli, "Blockchain-Based Proof of Location," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018, pp. 146–153. DOI: 10.1109/QRS-C.2018.00038.

[94] F. Zafar, A. Khan, A. Anjum, C. Maple, and M. A. Shah, "Location Proof Systems for Smart Internet of Things: Requirements, Taxonomy, and Comparative Analysis," *Electronics*, vol. 9, no. 11, p. 1776, Oct. 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9111776.

[95] C. Javali, G. Revadigar, K. B. Rasmussen, W. Hu, and S. Jha, "I Am Alice, I Was in Wonderland: Secure Location Proof Generation and Verification Protocol," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, Dubai: IEEE, Nov. 2016, pp. 477–485, ISBN: 978-1-5090-2054-6. DOI: 10.1109/LCN.2016.126.

[96] D. Rjazanovs and E. Petersons, "Decentralized Byzantine Fault Tolerant Proof of Location," in *PoEM Workshops*, 2020, pp. 1–10.

[97] R. Khan, S. Zawoad, M. M. Haque, and R. Hasan, "'Who, When, and Where?' Location Proof Assertion for Mobile Devices," in *Data and Applications Security and Privacy XXVIII: 28th Annual IFIP WG 11.3 Working Conference (DBSec)*, D. Hutchison, T. Kanade, J. Kittler, *et al.*, Eds., vol. 8566, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 146–162.

[98] F. Boeira, M. Asplund, and M. P. Barcellos, "Vouch: A Secure Proof-of-Location Scheme for VANETs," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, Montreal QC Canada: ACM, Oct. 2018, pp. 241–248, ISBN: 978-1-4503-5960-3. DOI: 10.1145/3242102.3242125.

[99] F. Boeira, M. Asplund, and M. Barcellos, "Decentralized proof of location in vehicular Ad Hoc networks," *Computer Communications*, vol. 147, pp. 98–110, Nov. 2019, ISSN: 01403664. DOI: 10.1016/j.comcom.2019.07.024.

[100] S. Gambs, M.-O. Killijian, M. Roy, and M. Traore, "PROPS: A PRivacy-Preserving Location Proof System," in *2014 IEEE 33rd International Symposium on Reliable Distributed Systems (SRDS)*, Nara, Japan: IEEE, Oct. 2014, pp. 1–10, ISBN: 978-1-4799-5584-8. DOI: 10.1109/SRDS.2014.37.

[101] Z. Zhu and G. Cao, "Toward Privacy Preserving and Collusion Resistance in a Location Proof Updating System," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 51–64, Jan. 2013, ISSN: 1536-1233. DOI: 10.1109/TMC.2011.237.

[102] S. Lee, H.-W. Seok, K.-r. Lee, and H. P. In, "B-GPS: Blockchain-Based Global Positioning System for Improved Data Integrity and Reliability," *ISPRS International Journal of Geo-Information*, vol. 11, no. 3, p. 186, Mar. 2022, ISSN: 2220-9964. DOI: 10.3390/ijgi11030186.

[103] Foamspace Corp, "FOAM Whitepaper," Tech. Rep., Jan. 2018. [Online]. Available: https://foam.space/publicAssets/FOAM_Whitepaper.pdf (visited on 08/17/2022).

[104] H. Wang, Y. Wen, Y. Lu, D. Zhao, and C. Ji, "Secure Localization Algorithms in Wireless Sensor Networks: A Review," in *Advances in Computer Communication and Computational Sciences*, vol. 760, Singapore: Springer Singapore, 2019, pp. 543–553.

[105] Y. Wei and Y. Guan, "Lightweight Location Verification Algorithms for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 938–950, May 2013. DOI: 10.1109/TPDS.2012.42.

[106] M. E. Pivaro Monteiro, J. L. Rebelatto, and R. D. Souza, "Information-Theoretic Location Verification System With Directional Antennas for Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 93–103, Jan. 2016. DOI: 10.1109/TITS.2015.2460114.

[107] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," in *Network and Distributed System Security Symposium (NDSS)*, 2004, p. 11.

[108] S. Yan, G. W. Peters, I. Nevat, and R. Malaney, "Location Verification Systems Based on Received Signal Strength With Unknown Transmit Power," *IEEE Communications Letters*, vol. 22, no. 3, pp. 650–653, Mar. 2018. DOI: 10.1109/LCOMM.2017.2787129.

[109] U. Ihsan, Z. Wang, R. Malaney, A. Dempster, and S. Yan, "Artificial Intelligence and Location Verification in Vehicular Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9014171.

[110] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, "A Range-Based Secure Localization Algorithm for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 19, no. 2, pp. 785–796, Jan. 2019. DOI: 10.1109/JSEN.2018.2877306.

[111] D. Wu, Y. Liu, L. Ma, and P. Jing, "A Blocking Level based Location Verification Scheme in Wireless Sensor Networks," in *2018 25th International Conference on Telecommunications (ICT)*, St. Malo: IEEE, Jun. 2018, pp. 181–185, ISBN: 978-1-5386-2321-3. DOI: 10.1109/ICT.2018.8464841.

[112] M. Schäfer, C. Nogueira, J. B. Schmitt, and V. Lenders, "Secure Location Verification: Why You Want Your Verifiers to Be Mobile," in *Computer Security*, S. Katsikas, F. Cuppens, N. Cuppens, *et al.*, Eds., vol. 11980, Cham: Springer International Publishing, 2020, pp. 419–437.

[113] J. Schaerer, S. Zumbrunn, and T. Braun, "Veritaa - The Graph of Trust," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, Paris, France: IEEE, Sep. 2020, pp. 168–175. DOI: 10.1109/BRAINS49436.2020.9223289.

[114] J. Schaerer, S. Zumbrunn, and T. Braun, "Veritaa: A distributed public key infrastructure with signature store," *International Journal of Network Management*, vol. 32, no. 2, Mar. 2022. DOI: 10.1002/nem.2183.

[115] M. Nottingham, "Well-Known Uniform Resource Identifiers (URIs)," Tech. Rep. RFC8615, May 2019. DOI: 10.17487/RFC8615.

[116] *Trust*, 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/trust (visited on 11/29/2020).

[117] D. Gambetta, "Can we trust trust," *Trust: Making and breaking cooperative relations*, vol. 13, pp. 213–237, 2000.

[118] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the 1997 workshop on New security paradigms (NSPW)*, Langdale, Cumbria, United Kingdom: ACM Press, 1997, pp. 48–60, ISBN: 978-0-89791-986-9. DOI: 10.1145/283699.283739.

[119] *Reputation*, 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/reputation (visited on 12/31/2020).

[120] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999, Publisher: ACM New York, NY, USA.

[121] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, Jan. 1978, ISSN: 0378-8733. DOI: 10.1016/0378-8733(78)90021-7.

[122] P. Bonacich, "Power and centrality: A family of measures," *American journal of sociology*, vol. 92, no. 5, pp. 1170–1182, 1987, Publisher: University of Chicago Press.

[123] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the twelfth international conference on World Wide Web (WWW)*, Budapest, Hungary: ACM Press, 2003, p. 640, ISBN: 978-1-58113-680-7. DOI: 10.1145/775152.775242.

[124] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web.," Stanford InfoLab, Tech. Rep., 1999.

[125] *Top Websites Ranking - Most Visited Websites in February 2023*. [Online]. Available: https://www.similarweb.com/top-websites/ (visited on 03/11/2023).

[126] Y. Liu, W. Tome, L. Zhang, *et al.*, "An End-to-End Measurement of Certificate Revocation in the Web's PKI," in *Proceedings of the 2015 Internet Measurement Conference (IMC)*, Tokyo Japan: ACM, Oct. 2015, pp. 183–196, ISBN: 978-1-4503-3848-6. DOI: 10.1145/2815675.2815685.

[127] A.-M. Kermarrec and M. van Steen, "Gossiping in distributed systems," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 5, pp. 2–7, Oct. 2007, ISSN: 0163-5980. DOI: 10.1145/1317379.1317381.

[128] A. Bundy and L. Wallen, "Breadth-First Search," in *Catalogue of Artificial Intelligence Tools*, A. Bundy and L. Wallen, Eds., Berlin, Heidelberg: Springer, 1984, pp. 13–13, ISBN: 978-3-642-96868-6. DOI: 10.1007/978-3-642-96868-6_25.

[129] I. Robinson, J. Webber, and E. Eifrem, *Graph databases - new opportunities for connected data*. Sebastopol, CA: O'Reilly, 2015, ISBN: 978-1-4919-3200-1. [Online]. Available: https://neo4j.com/graph-databases-book/?ref=home (visited on 12/02/2019).

[130] M. Needham and A. E. Hodler, *Graph algorithms: practical examples in Apache Spark and Neo4j*, First edition. Sebastopol, California: O'Reilly Media, 2019, OCLC: on1066191517, ISBN: 978-1-4920-4768-1.

[131] Peter Abeles, *Efficient Java Matrix Library*, 2020. [Online]. Available: http://ejml.org (visited on 01/01/2021).

[132] M. Latapy, C. Magnien, and N. D. Vecchio, "Basic notions for the analysis of large two-mode networks," *Social Networks*, vol. 30, no. 1, pp. 31–48, Jan. 2008, ISSN: 03788733. DOI: 10.1016/j.socnet.2007.04.006.

[133] *A Primer on IOTA*, May 2017. [Online]. Available: http://blog.iota.org/a-primer-on-iota-with-presentation-e0a6eb2cc621 (visited on 02/07/2021).

[134] M. Schäffer, M. di Angelo, and G. Salzer, "Performance and Scalability of Private Ethereum Blockchains," in *Business Process Management: Blockchain and Central and Eastern Europe Forum*, C. Di Ciccio, R. Gabryelczyk, L. García-Bañuelos, *et al.*, Eds., vol. 361, Cham: Springer International Publishing, 2019, pp. 103–118, ISBN: 978-3-030-30428-7. DOI: 10.1007/978-3-030-30429-4_8.

[135] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 –15.

[136] J. Schaerer, S. Zumbrunn, and T. Braun, "Veritaa-IoT: A Distributed Public Key Infrastructure for the Internet of Things," in *2022 IFIP Networking Conference (IFIP Networking)*, Catania, Italy: IEEE, Jun. 2022, pp. 1–9. DOI: 10.23919/IFIPNetworking55013.2022.9829794.

[137] *Identity*. [Online]. Available: https://www.merriam-webster.com/dictionary/identity (visited on 08/25/2021).

[138] D. L. Mills, *Computer network time synchronization: the Network Time Protocol on Earth and in space*, 2nd ed. Boca Raton, FL: CRC Press, 2011, ISBN: 978-1-4398-1463-5.

[139] A. Stanford-Clark and H. L. Truong, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification," International Business Machines Corporation (IBM), Tech. Rep., 1999, p. 28.

[140] Silicon Laboratories, "EFR32xG21 Wireless Gecko Reference Manual," Tech. Rep., Sep. 2020. [Online]. Available: `https://www.silabs.com/documents/public/reference-manuals/efr32xg21-rm.pdf` (visited on 09/14/2021).

[141] "UG103.05: IoT Endpoint Security Fundamentals," Silicon Labs, Tech. Rep., p. 13. [Online]. Available: `https://www.silabs.com/documents/public/user-guides/ug103-05-fundamentals-security.pdf` (visited on 11/12/2021).

[142] A. Ltd, *TrustZone*. [Online]. Available: `https://developer.arm.com/ip-products/security-ip/trustzone` (visited on 09/14/2021).

[143] *Arduino Cryptography Library: Arduino Cryptography Library*. [Online]. Available: `https://rweather.github.io/arduinolibs/crypto.html` (visited on 09/10/2021).

[144] *Micro-ecc*, original-date: 2013-05-05T16:29:20Z, Sep. 2021. [Online]. Available: `https://github.com/kmackay/micro-ecc` (visited on 09/10/2021).

[145] *Airtime calculator for LoRaWAN*. [Online]. Available: `https://avbentem.github.io/airtime-calculator/ttn/eu868/90,12` (visited on 09/10/2021).

[146] W. R. Knechel, G. V. Krishnan, M. Pevzner, L. B. Shefchik, and U. K. Velury, "Audit Quality: Insights from the Academic Literature," *AUDITING: A Journal of Practice & Theory*, vol. 32, no. Supplement 1, pp. 385–421, May 2013. DOI: `10.2308/ajpt-50350`.

[147] T. Mustapää, J. Autiosalo, P. Nikander, J. E. Siegel, and R. Viitala, "Digital Metrology for the Internet of Things," in *2020 Global Internet of Things Summit (GIoTS)*, Jun. 2020, pp. 1–6. DOI: `10.1109/GIOTS49054.2020.9119603`.

[148] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998. DOI: 10.1038/30918.

[149] J. Schaerer, A. Di Maio, and T. Braun, "APS: An Auditable Positioning System Based on Angle-of-Arrival Proof of Location and Graph of Trust," in *7th International Workshop on Annotation of useR Data for UbiquitOUs Systems (ARDUOUS)*, Atlanta, GA, USA, Mar. 2023.

[150] Silicon Laboratories, "AN1296: Application Development with Silicon Labs' RTL Library," Tech. Rep., 2021. [Online]. Available: https://www.silabs.com/documents/public/application-notes/an1296-application-development-with-rtl-library.pdf (visited on 08/08/2022).