TICKET TO RIDE: AN INVESTIGATION INTO THE USE OF BLOCKCHAIN TECHNOLOGY IN THE RAIL INDUSTRY

by

JOSEPH DAVID PREECE

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Birmingham Centre for Railway Research and Education

School of Engineering

College of Engineering and Physical Sciences

University of Birmingham

September 2020

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

In memoriam.

Eric Raymond Preece

1928-2015

"No human is limited."

- Eliud Kipchoge

Abstract

The rail industry in Great Britain (GB) is undergoing a renaissance. Rising passenger numbers, a steady freight industry, and numerous planned projects make this an exciting time to be involved. Nonetheless, the network relies upon many legacy systems that impede progress. This stifling of progress is noticeable in the digital domain in particular, where a cluster of impractical formats and systems leave swathes of data with untapped potential. The search for interoperability in the industry is not a new one; many have conducted investigations and projects, to no avail.

Blockchains are an exciting new avenue of technology and are beginning to disrupt various industries. Despite this, few investigations exist into the potential use of the technology in the rail industry. From a purely technical perspective, the distributed nature of the technology has the potential to overcome the issues of data centralisation and the lack of trust amongst stakeholders. Nevertheless, as a new technology, it is not yet fully understood by those in the industry. This lack of understanding is a barrier to adoption and is as essential to consider as the technical implications.

This thesis proposes a new model to aid the decision-making process of those seeking to use blockchain. We validate this model by utilising it for two rail-specific use cases. The first is to build a marketisable data-sharing platform for rail industry data. Within this project, we investigate both classical and post-quantum cryptographic approaches to the platform. The second is a brand new approach digital ticketing for the GB rail network, to initiate the process of replacing the legacy ticketing systems still in operation. We use blockchain technology as the core data store to achieve this. We demonstrate the viability of both use cases, supporting the appropriate deployment of blockchain technology in the rail industry.

Acknowledgements

The PhD is the ultimate academic adventure, and a testing time of life for each brave soul who embarks upon one. To dedicate so many years to a single idea is a great challenge, and requires not only resolve and spirit of oneself, but the encouragement and patience of those who share the journey with you.

With these words in mind, I have many to thank for their unwavering support as I have traversed this path myself. First of all, I give my utmost thanks to my girlfriend India, my friends, and my family. Together we have celebrated the successes and have sustained through the tougher times, of which I am truly grateful. Next, I would like to pay homage to countless former teachers and coaches of mine, who have crafted me into an individual of tenacity and resilience. Finally, I would like to thank my long-suffering supervisors Dr John Easton and Prof. Clive Roberts for their support, without which I could not have completed this thesis. Academic freedom is something I hold in high regard, and you have provided me the opportunities to explore ideas and learn the many tricks of this trade.

Contents

List of Figures **xvii**

List of Tables **xxiii**

List of Acronyms **xxv**

I Topic Establishment **1**

1 Introduction **3**

1.1 Overview 3

1.2 Thesis Structure 6

1.3 Publications 7

2 Literature Review **9**

2.1 Introduction 9

2.2 Blockchain Technology 14

2.2.1 Mechanics of Blockchain Technology 14

2.2.2 State-of-the-Art 22

2.2.3 Blockchain and Web 3.0 35

2.3 Cryptographic Prerequisites 41

2.3.1 Hash Functions 41

2.3.2 Symmetric Cryptography 45

2.3.3 Asymmetric Cryptography 47

2.4 Data and the Rail Industry 60
2.4.1 Ubiquitous Data 61
2.4.2 Data Platforms 68
2.4.3 Ticketing and the Rail Settlement Plan 69
2.5 Conclusion 73

3 Problem Statement **75**
3.1 A Disruptive Technology or Marketing Buzzword? 75
3.2 The Relationship Between Data and the Rail Industry 76
3.3 Research Questions 77

II Novel Research **79**

4 An Improved Model for Identifying Blockchain Suitability **81**
4.1 Introduction 82
4.2 Methodology 83
4.2.1 The Prepare Phase and Layer 1 85
4.2.2 The Provide Response Phase and Layer 2 95
4.2.3 The Obtain Feedback Phase and Layer 3 97
4.3 Results 99
4.3.1 Model Scope 100
4.3.2 Use Case Evaluation and Validation 101
4.3.3 Software Performance 106
4.4 Discussion 107
4.4.1 Advantages 107
4.4.2 Disadvantages 108
4.4.3 Impact 109
4.5 Conclusion 109

5 Towards a Distributed Data Platform for Rail Industrial Data **111**

5.1 Introduction . . . 112

5.2 Methodology . . . 115

5.2.1 Suitability of Blockchain Technology . . . 115

5.2.2 Data Collaboration Protocols . . . 117

5.2.3 Integration of Quantum-Resistant Cryptography . . . 127

5.3 Results . . . 130

5.3.1 Quantum Resistant Encryption . . . 130

5.3.2 RIDE Classical Approach . . . 133

5.3.3 RIDE Quantum Approach . . . 136

5.4 Discussion . . . 136

5.4.1 Data Sharing Protocols . . . 137

5.4.2 Quantum Resistant Approach and Security . . . 139

5.4.3 Impact . . . 141

5.5 Conclusion . . . 142

6 Towards a Distributed Rail Ticketing Platform **145**

6.1 Introduction . . . 146

6.2 Methodology . . . 147

6.2.1 Suitability and Preliminary Decisions . . . 148

6.2.2 Development of Platform Architecture . . . 151

6.2.3 Development of Platform Protocols and Mechanics . . . 153

6.3 Results . . . 154

6.3.1 Administrative Organisations . . . 155

6.3.2 Ticket Vending Organisations . . . 157

6.3.3 Passengers . . . 160

6.4 Discussion . . . 162

6.4.1 Benefits . . . 162

6.4.2 Limitations . . . 164

6.4.3 Impact 164
6.5 Conclusion 165

III Concluding Remarks and Appendices 167

7 Conclusion 169
7.1 Response to Research Questions 169
7.2 Impact 172
7.3 Future Research 172

A Chapter 2 Supplements 175
A.1 Advanced Encryption Standard 175
A.2 Rail Data Process Maps 181

B Chapter 4 Supplements 185
B.1 Clarification on Notation 185
B.2 Extended Results 185

Bibliography 189

List of Figures

2.1 Centralised, decentralised, and distributed (Peer-to-Peer (P2P)) networks [5]. 10

2.2 A generalised overview of a blockchain platform [25]. 15

2.3 A newly proposed model of encapsulating blockchain components. . . . 16

2.4 The blockchain illustrated: the green block is the genesis block; the black blocks are those on the consensus chain; the purple blocks are the rejected orphan blocks [141]. 20

2.5 The suitability model of Lo et al. [84]. 25

2.6 The suitability model of Wust and Gervais [150]. 26

2.7 The suitability model of Maull et al. [86]. 27

2.8 Asserted information in a simple ontology. 37

2.9 Asserted and inferred information in a simple ontology. 38

2.10 Abstraction of a hash function. An input x of arbitrary length l passes through a hash function H to form a hash digest y of a pre-determined length n. 42

2.11 Illustration of symmetric key cryptography [130]. 46

2.12 Illustration of asymmetric key cryptography [130]. 47

2.13 The point addition of $P + Q = -R$ over an arbitrary elliptic curve [35]. . 49

2.14 The point addition of $P + Q = -R$ over an arbitrary elliptic curve in a finite field $\mathbb{F}_p$ [35]. 50

2.15 Abstraction of a key exchange protocol. The paint mixture represents mathematical operations [147]. 53

2.16 High-level overview of the Elliptic Curve Integrated Encryption Scheme (ECIES) [91]. 55
2.17 The Intelligent Integration of Railway Systems (InteGRail) methodology [80]. 63
2.18 The benefits of utilising a universal exchange format [113]. 64
2.19 The Railway Core Ontology (RaCoOn) architecture [143]. 65
2.20 Candidate architecture for ontology deployment within railway operations [142]. 67
2.21 Process map showing the collection of passenger rail transport administrative data [114]. 71

4.1 The hierarchical layers of the Core of the Model of Blockchain Suitability (MoBS) and the Interactive Process. Note that higher layers are dependent on lower layers. 84
4.2 Abstracted illustration of Layer 1 of the Core. There are four major concepts: the layer (blue); properties (red); statements (yellow); and scores (grey). 85

5.1 The process of the RIDE One-Key Approach (RIDE-1). Dashed lines and solid lines are synonymous, and only different to clarify crossover points. 118
5.2 The process of the RIDE Two-Key Approach (RIDE-2). Dashed lines and solid lines are synonymous, and only different to clarify crossover points. 120

6.1 The System for Ticketing Ubiquity with Blockchain (STUB) architecture. The different colours represent different organisations. 152
6.2 The process flow of the platform. 154

A.1 The process of the Advanced Encryption Standard (AES). 177
A.2 An illustration of the `AddRoundKey` step of AES [39]. 178
A.3 An illustration of the `SubBytes` step of AES [39]. 179
A.4 An illustration of the `ShiftRows` step of AES [39]. 179

A.5 An illustration of the `MixColumns` step of AES [39]. 180
A.6 Process map showing the collection of freight rail transport administrative data [114]. 181
A.7 Process map showing the flow of communication between data suppliers to output producers [114]. 181
A.8 Process map showing the passenger rail quality assurance checks by data suppliers [114]. 182
A.9 Process map showing the freight rail quality assurance checks by data suppliers [114]. 182
A.10 Process map showing the quality assurance method from the Office for National Statistics (ONS) to users [114]. 182
A.11 Process map showing the overall flow of rail transport statistics from suppliers to the ONS [114]. 183

List of Tables

2.1 The genesis transaction of Bytecoin, *Transaction 0*. Alice, Bob, and Charlie each receive 10 bytecoins. 11

2.2 The genesis block of Bytecoin, *Block 0*. The block includes Transaction 0. 11

2.3 The state of the Bytecoin network after the genesis block. 11

2.4 *Transaction 1*. Alice transacts 5B to Bob. 12

2.5 *Transaction 2*. Bob transacts 3B to Alice, and 7B to Charlie. 12

2.6 Charlie's mined block, *Block 1*. This block includes Transactions 1 and 2, and a reward transaction. 13

2.7 The state of the Bytecoin network after the first mined block. 13

2.8 Classification of public, private, and federated blockchain platforms [25]. 23

3.1 The locations of where this thesis answers the research questions. 78

4.1 A summary of the scores each category may take. Henceforth, the positive score is highlighted in blue, and the negative score is highlighted in red. 86

4.2 The attributes and properties of the MoBS Layer 1 implementation used in this thesis. The ordering within the table is retained, thus the identity convention is straightforward; for example, *A2P3* means the third property (P3) of the second attribute (A2). 87

4.3 The statements for the Participants property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 88

4.4 The statements for the Writers property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 89

4.5 The statements for the Trust property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 90

4.6 The statements for the Consensus property, and the relevant suitability scores for non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 90

4.7 The statements for the State property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 91

4.8 The statements for the Transparency property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 92

4.9 The statements for the Integrity property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 92

4.10 The statements for the Latency property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 93

4.11 The statements for the Scale property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 94

4.12 The statements for the Redundancy property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2. 94

4.13 The suitability level values and their respective indications for a score x. 98

4.14 The potential platforms for business and industry use cases, with a particular insight into energy and sustainability, supply chains, and transportation. 99

4.15 The potential platforms for data security and management use cases, with a particular insight into data distribution, and secure storage. . . . 100

4.16 The potential platforms for governance use cases, with a particular insight into eVoting, and Decentralized Autonomous Organizations (DAOs). . . 100

4.17 The potential platforms for integrity verification use cases, with a particular insight into intellectual property, and insurance. 101

4.18 The potential platforms for Internet of Things (IoT) use cases, with a particular insight into device management, and data capture and distribution. 101

4.19 The distribution of scores for every possible Individual Score Array (ISA) in the MoBS. 102

4.20 The distribution of scores for every possible Domain Score Array (DSA) in the MoBS. 102

4.21 An alternative distribution of the results of each ISA in the MoBS, with a focus on the suggested platforms. 103

4.22 An example response to MoBS. The simplified string identity is "1111111111". 104

4.23 Frank's response to the MoBS for his approach to currency. The simplified string identity is "1111111321". 104

4.24 Susan's response to the MoBS for her approach to currency. The simplified string identity is "2322121321". 105

4.25 Frank's response to the MoBS for his approach to supply chains. The simplified string identity is "2222121221". 105

4.26 Susan's response to the MoBS for her approach to supply chains. The simplified string identity is "2233121221". 105

4.27 Frank's response to the MoBS for his approach to data storage. The simplified string identity is "4344222212". 106

4.28 Susan's response to the MoBS for her approach to data storage. The simplified string identity is "2322222332". 106

4.29 Performance results for the MoBS. 106

5.1 The ISA and DSA of the response to the MoBS for the Rail Industry Data Environment (RIDE) use case. 116

5.2 Public key sizes, private key sizes, and signature sizes for different approaches of the Supersingular Isogeny Digital Signature Algorithm (SIDSA). 131

5.3 Performance of the signing and verification algorithms. 132

5.4 Performance of the Non Interactive Zero Knowledge Proof (NIZKP) rounds of the signing algorithm. 133

5.5 Key sizes and signature sizes of other post-quantum Digital Signature Schemes (DSSs). 140

5.6 Clock cycles of signing and verifying algorithms of other post-quantum DSSs. 140

6.1 The ISA and DSA of the response to the MoBS for the STUB use case. . . 149

A.1 Parameters of number of columns for the state array Nb, number of columns for the key array Nk, and number of rounds Nr 175

A.2 Illustration of how a 128-bit key k expands and splits into eleven round keys [39]. 176

B.1 Frank's response to the MoBS for his approach to currency. The simplified string identity is "1111111321". 186
B.2 Susan's response to the MoBS for her approach to currency. The simplified string identity is "2322121321". 186
B.3 Frank's response to the MoBS for his approach to supply chain. The simplified string identity is "2222121221". 187
B.4 Susan's response to the MoBS for his approach to supply chain. The simplified string identity is "2233121221". 187
B.5 Frank's response to the MoBS for his approach to data storage. The simplified string identity is "4344222212". 188
B.6 Susan's response to the MoBS for his approach to data storage. The simplified string identity is "2322222332". 188

List of Acronyms

AES Advanced Encryption Standard

ABox asserted triples

API Application Programming Interface

BiTA Blockchain in Transport Alliance

BPQS Blockchained Post-Quantum Signatures

CSSI Computational Supersingular Isogeny

CSV Comma-Separated Values

DApp Decentralised Application

DAO Decentralized Autonomous Organization

dBFT Delegated Byzantine Fault Tolerance

DES Data Encryption Standard

DfT Department for Transport

DLT Distributed Ledger Technology

DHKE Diffie-Hellman Key Exchange

DRF Digital Rail Framework

DSA Domain Score Array

DSS Digital Signature Scheme

DSSP Decisional Supersingular Product

ECC Elliptic Curve Cryptography

ECDH Elliptic Curve Diffie Hellman Key Exchange

ECDSA Elliptic Curve Digital Signature Algorithm

ECIES Elliptic Curve Integrated Encryption Scheme

ERC20 Ethereum Request for Comments Proposal 20

ERTMS European Railway Traffic Management System

EVM Ethereum Virtual Machine

FCA Financial Conduct Authority

GCM Galois/Counter Mode

GB Great Britain

HKDF Hash-Based Message Authentication Code Key Derivation Function

HTTP Hypertext Transport Protocol

IEEE Institute of Electrical and Electronics Engineers

InteGRail Intelligent Integration of Railway Systems

IoT Internet of Things

IPFS InterPlanetary File System

IRS International Railway Standard

IT Information Technology

ITSO Integrated Transport Smartcard Organisation

IV Initialisation Vector

JSON JavaScript Object Notation

KDF Key Derivation Function

KEK Key Encryption Key

KES Key Exchange Scheme

LENNON Latest Earnings Network Nationally Overnight

MAC Message Authentication Code

MD Message-Digest

MDAG Merkle Directed Acyclic Graph

MoBS Model of Blockchain Suitability

MSP Membership Service Provider

NASDAQ National Association of Securities Dealers Automated Quotations

NFC Near Field Communication

NIST National Institute of Standards and Technology

NIZKP Non Interactive Zero Knowledge Proof

ODASE Ontology-Driven Architecture and Service Engineering

ONS Office for National Statistics

OOP Object-Oriented Programming

ORCATS Operational Research Computerised Allocation of Tickets to Services

OWL Web Ontology Language

P2P Peer-to-Peer

PoET Proof-of-Elapsed-Time

PoS Proof-of-Stake

PoW Proof-of-Work

QR Quick Response

RaCoOn Railway Core Ontology

RailML Railway Markup Language

RDF Resource Description Framework

RDG Rail Delivery Group

RFID Radio Frequency Identification

RIDE Rail Industry Data Environment

RIDE-1 RIDE One-Key Approach

RIDE-2 RIDE Two-Key Approach

RIDE-C RIDE Classical Approach

RIDE-Q RIDE Quantum Approach

RIDE-1C RIDE-1 Classical

RIDE-1Q RIDE-1 Quantum

RIDE-2C RIDE-2 Classical

RIDE-2Q RIDE-2 Quantum

RSA Rivest–Shamir–Adleman

ISA Individual Score Array

RSP Rail Settlement Plan

SHA Secure Hash Algorithm

SIC Supersingular Isogeny Cryptography

SIDH Supersingular Isogeny Diffie Hellman Key Exchange

SIDSA Supersingular Isogeny Digital Signature Algorithm

SIKE Supersingular Isogeny Key Exchange

STUB System for Ticketing Ubiquity with Blockchain

SWRL Semantic Web Rule Language

TBox inferred triples

TLS Transport Layer Security

TOC Train Operating Company

TVO Ticket Vending Organisation

UIC International Union of Railways

UK United Kingdom

μTP Micro Transport Protocol

VID VeChain Identity

W3C World Wide Web Consortium

WebRTC Web Real-Time-Communication

WWW World Wide Web

XML eXtensible Markup Language

XMSS eXtended Merkle Signature Scheme

ZKP Zero Knowledge Proof

Part I

Topic Establishment

Chapter 1

Introduction

This chapter introduces the topic of blockchain technology and the structure of this thesis. Section 1.1 provides a description of the evolution of blockchain technology from a non-technical perspective. Section 1.2 outlines the structure of this thesis, and provides relevant descriptions of the three parts and seven chapters that compose it. Section 1.3 presents a list of publications (published during the undertaking of this thesis) that form the groundwork of the research.

1.1 Overview

When Satoshi Nakamoto initiated *Bitcoin* in January 2009, few knew the disruptive impact this exciting new technology would create in the years to follow. Nakamoto created Bitcoin to solve an old problem in the financial industry; removing the need for intermediate parties to process payments [90]. Those reading this thesis will (most likely) have experienced a financial intermediary in one form or another; banks, building societies, and online applications such as PayPal all fall under this category. When we open a bank account, we entrust this bank to securely store our money, provide measures against theft, and allow seamless access to our funds. When we transact funds using a debit card, we entrust a card company (such as Visa or Mastercard) to process the payment from our bank account to the recipient's bank account. It is straightforward

to see that there might be more than one intermediary in certain situations, creating a complex process chain.

The use of financial intermediaries is not without benefit. For one, they provide convenient and seamless services to interact with our money, significantly reducing the risks involved with financial transactions. For example, assume oneself is purchasing a two-bedroom maisonette in Bermondsey. Given a choice between an electronic transfer or a cash-in-hand payment, one would most likely opt for the former of the two options. Few would want to carry £1,000,000 through the streets of south London; those who conduct themselves in such a way probably do so to support other nefarious activities, such as money laundering. This raises another advantage of intermediaries; the ability to trace the intention of a transaction and to aid the authorities in capturing illegal activities.

The financial domain is not the only sector that uses intermediaries. Supply chains rely upon central intermediaries to organise and distribute the swathes of data between the large number of stakeholders operating throughout complex chains of logistics. Companies rely on software that makes use of external servers to communicate and operate. Even we, as users of internet services such as Google and Apple, entrust corporations to store and handle our sensitive data and documents; a particularly interesting scenario, whereby they become intermediaries to data transactions between ourselves.

There is a key element which the entire process of using intermediaries relies upon; trust. Many intermediaries conduct under strict regulations, set by governing bodies. In the United Kingdom (UK), banks and building societies adhere to the rules of the Financial Conduct Authority (FCA). Because the government regulates the FCA, and the citizens of the UK trust their government[1], the intermediary inherits the trust. The intermediaries provide a common point of trust; it is far harder to establish trust between all individuals. The digital age exacerbates this establishment of trust, where

[1]Even those of opposing political dispositions are likely to trust the handling of the FCA.

fraudulent claims of identity are made easier due to the anonymous nature of the World Wide Web (WWW).

Intermediaries are all well and good for the financial industries of modern, democratic countries. Nevertheless, the scope of blockchain technology has expanded beyond the transaction of monetary currency. In many ways, data is the currency of the digital age, and data is far more intrusive to our daily lives. Data reveals our habits and our desires; information that many wish to keep to themselves. With the current monopolies of the WWW, we continue to entrust large corporations to store and use our sensitive data. Once more, the most appealing factor of this is that it is convenient. However, the ideology of controlling our own assets and data (as opposed to shifting the majority of the responsibility to intermediaries) is an enticing one; after all, the one person we can trust the most is ourself. Thus, it is crucial to find alternative solutions and, according to those evangelising blockchains, is where blockchain technology will come in to play.

The rail industry is no exception to these issues, especially the rail network of Great Britain (GB). A rail network should be a cohesive unit; an entity made of cooperative, interoperable components that form an efficient system. Instead, a cluster of outdated legacy systems and a restrictive privatised industry cause sleepless nights for those requiring access to essential data they do not own. Despite existing in the digital age, the potential of vast quantities of raw data remains untapped, trapped in database silos from several competing organisations unwilling to share their most valuable assets. This is not a healthy approach to operating a railway, as it stifles progress in regards to efficiency and safety. Nevertheless, with no concrete plans to reform the flawed system, the best solution appears to be an addition that makes data sharing a seamless process. Might blockchain provide the long-sought answer to the issues that plague the current systems? This thesis intends to find out.

1.2 Thesis Structure

This thesis presents itself in seven chapters, spread over three parts. Part I introduces the relevant concepts and technical information that underlie the novel research undertaken in the thesis. Part II presents this novel research, conducted into the application of blockchain technology within the rail industry, and the subsequent results and discussions. Finally, Part III provides concluding statements with regards to the findings of the thesis as a whole, and relevant appendices.

Within Part I, there are three chapters. Chapter 1 - this current chapter - introduces blockchain technology and associated concepts inherent to the thesis in the form of a summary. Chapter 2 provides a detailed literature review and technical background of blockchain technology and related topics, such as the prerequisite cryptography and many use cases built upon blockchain platforms. Chapter 2 also details the current approaches to data ubiquity and distribution within the rail industry, and identifies the gaps in the literature that blockchain technology might fill. Chapter 3 reviews these gaps in the literature review, and derives several research questions to serve as the basis of Part II.

Within Part II, there are three chapters. Chapter 4 describes the Model of Blockchain Suitability (MoBS), an improved decision model to aid the decision process of whether users should use blockchain technology for their use case, and (in the cases where blockchain technology is suitable) which type of blockchain suits best, and which platforms are readily available to do so. Chapters 5 and 6 both utilise the MoBS to validate the model and identify appropriate use cases for the rail industry. Chapter 5 introduces the Rail Industry Data Environment (RIDE), a data hosting platform that uses a mixture of distributed web and blockchain technologies to achieve security and traceability, with a marketisable, rail industry context in mind. (Though the concepts are expandable to other sectors.) This platform is built in two cryptographic domains; classical (using Elliptic Curve Cryptography (ECC)) and post-quantum (using Super-

singular Isogeny Cryptography (SIC)). As part of the post-quantum construction, this chapter introduces subtle improvements to an existing Supersingular Isogeny Digital Signature Algorithm (SIDSA) to bolster the efficiency of the RIDE and explore the usage of SIC in blockchains. Chapter 6 presents the System for Ticketing Ubiquity with Blockchain (STUB) platform. The STUB is a Hyperledger Fabric based platform designed to host a distributed rail ticketing system for the Great Britain (GB) rail network, to overcome key flaws of current approaches.

Within Part III, there is one chapter and two appendices. Chapter 7 presents a collective summary of the individual findings from Part II, and discusses the implication of these results within the rail industry. Furthermore, Chapter 7 presents avenues for future research conductible beyond the scope of this thesis. Appendices A and B provide supplementary materials for Chapters 2 and 4 respectively.

1.3 Publications

The author has presented the research conducted for this thesis in several publications before submission.

- Blockchain Technology as a Mechanism for Digital Ticketing [106]
  - Conference Proceedings of the Fourth International Conference on Railway Technology
  - Published September 2018
- A Review of Prospective Applications of Blockchain Technology in the Railway Industry [107]
  - Preprint to the International Journal of Railway Technology
- Towards Encrypting Industrial Data on Public Distributed Networks [108]

  - Conference Proceedings of the 2018 Institute of Electrical and Electronics Engineers (IEEE) International Conference on Big Data
  - Published February 2019

- Blockchain Technology as a Mechanism for Digital Railway Ticketing [109]
  - Conference Proceedings of the 2019 IEEE International Conference on Big Data
  - Published February 2020

Chapter 2

Literature Review

This chapter provides a literature review and technical background of the technologies and concepts used in the scope of this thesis. Section 2.1 presents a description of a blockchain using terminology suited for a non-technical introduction to the technology. Section 2.2 discusses the mechanics and applications of blockchain technology, and the state-of-the-art of existing platforms. Section 2.3 presents the cryptographic prerequisites required to understand the core of blockchain technology and the technical work conducted in this thesis. Section 2.4 identifies the current approach to data solutions within the rail industry of Great Britain (GB) and discusses the pitfalls of these approaches. Section 2.5 summarises the findings of this literature review.

2.1 Introduction

Fundamentally, a blockchain platform is a Peer-to-Peer (P2P) network that maintains a ledger of transactions between the platform users. A P2P network is a distributed network whereby the nodes connect directly to one another, as opposed to a connection via a centralised node. Figure 2.1 illustrates this. The transactions are agreed upon by the users of the network instead of an intermediary. Users collectively agree upon which transactions are valid via an amalgamation of cryptographic protocols. (We will cover these protocols in-depth in Section 2.3.) To provide an example, assume Alice,

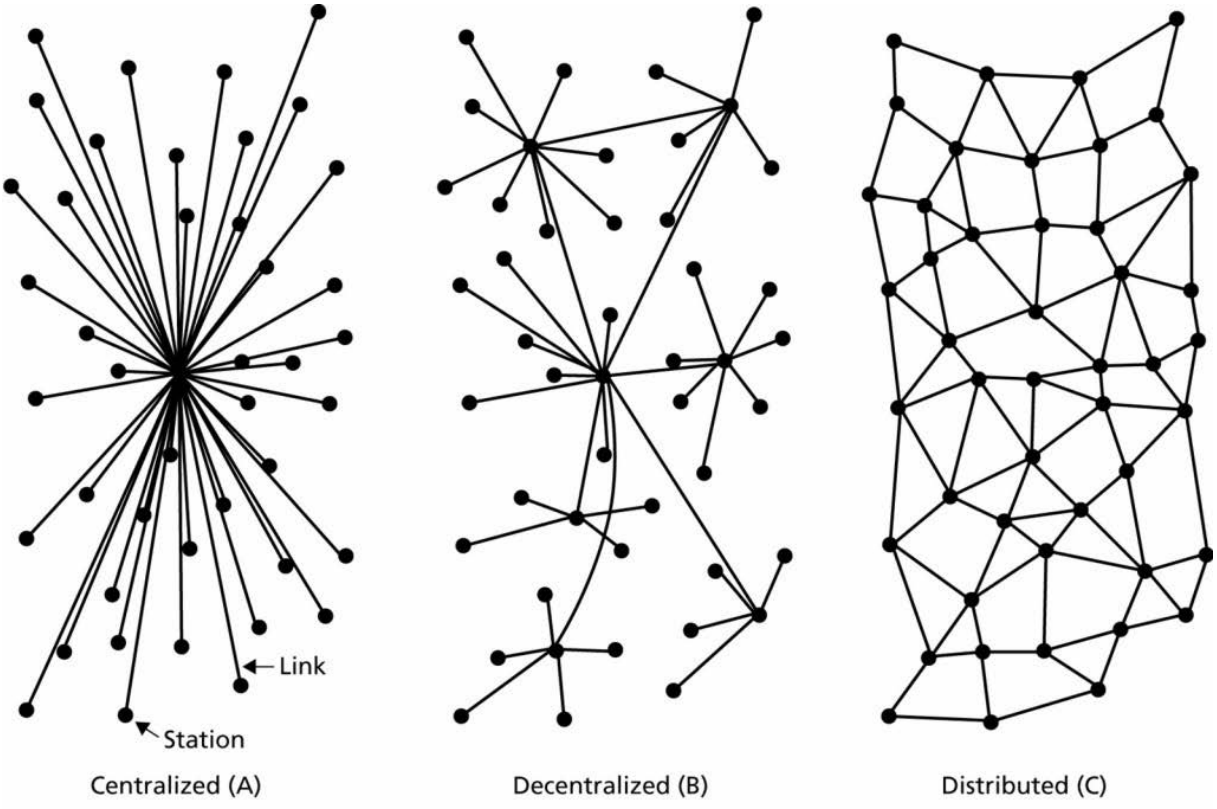


Figure 2.1: Centralised, decentralised, and distributed (P2P) networks [5].

Bob, and Charlie are three users on a hypothetical blockchain platform named *Bytecoin*. The goal is to transfer *bytecoins* amongst one another. To proceed, each user installs the Bytecoin software on their local system. Once installed, they each create an account formed of a cryptographic key pair; a private key only known to the individual, and a public key, known to everybody using Bytecoin. The software sets their system up as a node on the P2P Bytecoin network, to allow their device to communicate with the other nodes.

In order to begin transacting bytecoins between one another, they must initiate the blockchain data structure at the heart of the Bytecoin platform. Every blockchain has a genesis block; the first block in the chain. For Bytecoin, the genesis block tells the platform the initial number of minted bytecoins at each account. The three users agree to initiate the genesis block with 10B (ten bytecoins) each. A special type of transaction (known as a *Mint* transaction) makes up this block. Table 2.1 illustrates this transaction *Transaction 0*, and Table 2.2 illustrates the genesis block *Block 0* where Transaction 0 is

stored. Table 2.3 indicates the state of the Bytecoin network after initiating the genesis block.

Table 2.1: The genesis transaction of Bytecoin, *Transaction 0*. Alice, Bob, and Charlie each receive 10 bytecoins.

Transaction ID: 0 (Genesis) Transaction Type: Mint		
Recipient		
Name	Address	Coins
Alice	0x...a	10
Bob	0x...b	10
Charlie	0x...c	10

Table 2.2: The genesis block of Bytecoin, *Block 0*. The block includes Transaction 0.

Block ID: 0 Previous Block Hash: -				
Transaction				
ID	Type	Sender	Recipient	Coins
Transaction 0	Mint	-	Alice	10
		-	Bob	10
		-	Charlie	10

Table 2.3: The state of the Bytecoin network after the genesis block.

	Recipient		
Block ID	Alice	Bob	Charlie
Block 0	10	10	10

Alice wishes to transfer 5B to Bob. To achieve this, Alice uses Bytecoin software to initiate a transaction. She instructs the software to set up a *Spend* transaction to send 5B to Bob. Alice sets up this transaction, *Transaction 1*, by providing Bob's public key and signing the transaction with her private key to produce a digital signature S_1. Table 2.4 illustrates Transaction 1. After this, Bob proceeds to transact 3B back to Alice, and 7B to Charlie in a single transaction, *Transaction 2*, using the same process as Alice. Table 2.5 illustrates Transaction 2.

Table 2.4: *Transaction 1.* Alice transacts 5B to Bob.

Transaction ID: 1		
Transaction Type: Spend		
Sender: 0x...a (Alice)		
Signature: S_1		
Recipient		
Name	Address	Coins
Bob	0x...b	5

Table 2.5: *Transaction 2.* Bob transacts 3B to Alice, and 7B to Charlie.

Transaction ID: 2		
Transaction Type: Spend		
Sender: 0x...b (Bob)		
Signature: S_2		
Recipient		
Name	Address	Coins
Alice	0x...a	3
Charlie	0x...c	7

The software transmits Transactions 1 and 2 to all the users on the P2P network. However, the Bytecoin platform has not yet appended them to the blockchain; the users must determine whether the transactions are valid before its addition. This is known as consensus and requires a consensus mechanism. To verify the transactions, users on the network compete to provide consensus by sorting them into blocks of valid transactions and appending these blocks to the latest block. (In this case, the latest block is simply the genesis block.)

So why might Alice, Bob, or Charlie provide consensus? Furthermore, why might they trust one another with such a task? The answer to the former is that Bytecoin provides an incentive for them to provide honest consensus; should the network accept their block proposal, they will be rewarded with five newly minted Bytecoin. The answer to the latter is that the majority of platform users must accept the block in order to append it to the latest blockchain. Assume that Charlie attempts to mine a block that includes Transactions 1 and 2 (Tables 2.4 and Table 2.5). An additional Mint

transaction *Transaction 3* is the reward transaction that goes to Charlie's address, should he successfully mine the block. Once Charlie has assembled a candidate block, he adds the hash value of the previous block; a number that acts as a snapshot of the content of the block. Table 2.6 illustrates this block, *Block 1*. Both Alice and Bob confirm that the

Table 2.6: Charlie's mined block, *Block 1*. This block includes Transactions 1 and 2, and a reward transaction.

Block ID: 1
Previous Block Hash: 0x...0

Transaction				
ID	Type	Sender	Recipient	Coins
Transaction 1	Spend	Alice	Bob	5
Transaction 2	Spend	Bob	Alice	3
			Charlie	7
Transaction 3	Mint	-	Charlie	5

signatures of all transactions within the block are valid. In the case where any signature is invalid, they reject the block and proceed in seeking a new one. Furthermore, if the hash value of the previous block changes, Alice and Bob can instantly realise that Charlie has tampered with the data in a previous block, and can dismiss this new block. Table 2.7 indicates the state of the Bytecoin platform after Block 1 appends to the chain.

Table 2.7: The state of the Bytecoin network after the first mined block.

	Recipient		
Block ID	Alice	Bob	Charlie
Block 0	10	10	10
Block 1	8	5	22

The above example is a crude simplification of a blockchain platform and two blocks. In reality, there are far more aspects to consider, and Section 2.2 will cover these in detail.

2.2 Blockchain Technology

This section provides a detailed explanation of blockchain technology and subsequent use cases. Section 2.2.1 focuses on the construction of a blockchain platform and the application of cryptographic protocols. It identifies models that represent the complex formations of blockchain platforms in a simplified manner, providing visual aids to help illustrate the abstract concepts. Section 2.2.2 discusses the classifications and use cases of blockchain technology and existing platforms that aim to solve these problems. Finally, Section 2.2.3 discusses the integration of blockchain technology with other technologies and frameworks, with a particular focus on Web 3.0 and semantic data paradigms.

2.2.1 Mechanics of Blockchain Technology

Nakamoto [90] introduces the idea of a "peer-to-peer version of electronic cash" that utilises "digital signatures [to] provide... the solution". Nakamoto [90] proceeds to explain the protocols that govern this system, providing details on the Peer-to-Peer (P2P) network and the "blocks" that record the network transactions. The term *blockchain* was later coined by Brito and Castillo [19] who stated that "every transaction that occurs in the bitcoin economy is registered in a public, distributed ledger, which is called the block chain [sic]". Christidis and Devetsikiotis [32] expand on this definition by stating that the approach to a blockchain allows users to "verifiably interact with each without the need for a trusted authority", introducing the idea that blockchains exist to remove intermediaries from the processes. Since the publication of these documents, the research and development surrounding blockchain technology has flourished with the exploration of many new use cases and improved mechanics.

Blockchain is a common term and has come to encapsulate all components of the entire blockchain platform, including the network, data structures, and the associated protocols. It is important to distinguish these components from one another. Casino

et al. [25] present a generalisation of a blockchain platform with various layers of mechanisms: the *Transactions* and *Blocks* layer, defining the fundamental data structures at the heart of the platform; the *Consensus* layer, responsible for defining how new blocks append to the existing blockchain; the *Compute Interface* layer which eases the human interaction with the blockchain platform; and the *Governance* layer, which abstracts the decisions made on maintaining the platform. Figure 2.2 illustrates these layers. The

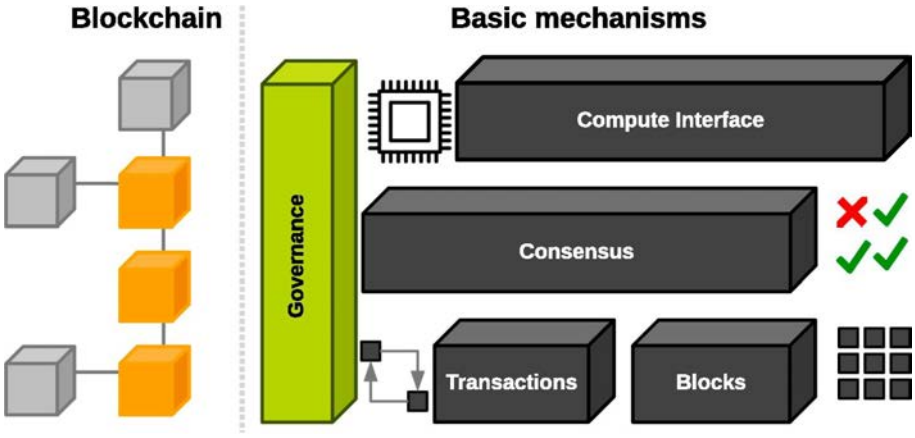


Figure 2.2: A generalised overview of a blockchain platform [25].

advantage of creating such a model is twofold. Firstly, it encapsulates the physical components, protocols, and interactivity in a hierarchical model to illustrate dependency and structure, which provides consistency to developers of various blockchain platforms. Secondly, the model educates those who are new to blockchain from a top-down perspective. Nonetheless, we believe the approach of Casino et al. [25] can simplify to a simple three-layer hierarchical model, proposed in Figure 2.3. The new model confines the layers (from the Casino et al. [25] model) of Transactions and Blocks, and the blockchain itself into one layer, named *Base Structures*. The second layer, *Network*, groups the P2P network - something not included in the Casino et al. [25] model - with the user nodes and the Consensus layer the users follow to interact with the Base Structures layer. Finally, the *Interactivity* layer groups all interactive process, including both the Governance and Compute Interface layers. This new model introduces concepts that the Casino et al. [25] model fails to address while wrapping it into a single simplified hierarchy. This section discusses these three mechanism layers and the associated



Figure 2.3: A newly proposed model of encapsulating blockchain components.

protocols with each layer. As with the model, the discussion generalises blockchain platforms by their core components and properties, with no particular reference to actual implementations of blockchains.

Base Structures

Casino et al. [25] define a blockchain as a "distributed append-only timestamped data structure". A series of blocks, each filled with several transactions, forms this data structure.

A Chain of Blocks The blockchain data structure is a hash linked list. Consider a regular linked list where each block of the list contains data and a link that refers to its preceding block. This link is a pointer and is the memory location of the preceding block. A hashed linked list is analogous to a regular linked list. However, instead of a pointer to a memory location, the hashed linked list uses a hash pointer. A hash pointer is the hash digest of the preceding block. This is significant because it allows instantaneous verification of the data within the preceding block to ensure adversaries have not tampered the data within the block. The use of hashing makes the entire

blockchain a tamper-evident log. Should an adversary attempt to change any data in any preceding block, each succeeding block's hash digest will change, thus breaking the chain [94].

Within each block are the hash pointer pointing to the previous block, the list of transactions in the form of a Merkle tree, and a timestamp. Becker [7] defines a Merkle tree as a "tree structure in which each leaf node is a hash of a block of data, and each non-leaf node is a hash of its children". The blocks use Merkle trees in order to make use of the average search complexity of $O(\log_2(n))$ [136] in order to verify the transactions stored within the tree efficiently.

Transactions The users of the platform initiate the transactions that make up the blocks described above. Each user has a keypair; a public key and a private key. The public key (or a hashed version of the public key) is the wallet *address* for the corresponding user. A transaction takes place when a user of one wallet sends some of a platform's native currency or asset to another wallet. For a simplified example, please refer to Section 2.1.

Smart Contracts An additional concept that the model of Casino et al. [25] does not illustrate is the smart contract. Szabo [134] defines a smart contract as "a computerised transaction protocol that executes the terms of a contract". This allows for replication of an analogue contract within a digital domain [135], which guarantees execution of the contract. This negates the potential disputes and errors that occur with traditional contracts. Within the blockchain context, a smart contract is a replication of such a contract embedded on the blockchain itself. Thus, the blockchain distributes the contract amongst network peers; this removes the need for a trusted authority between parties involved [32].

Network

A unique property of a blockchain platform is the method of storing the data. Instead of a centralised database maintaining the latest version of the blockchain, distributed

nodes on a P2P network store localised copies of the blockchain. Distributed networks have several properties that provide benefits over centralised networks. Firstly, they are fault-tolerant as they rely on several nodes which are unlikely to fail simultaneous to one another, as opposed to relying on one node prone to failure. Furthermore, they are attack resistant, as there is no sensitive central node. Finally, they are collusion resistant, as it is hard for adversaries to combine efforts to manipulate the network [24, 32].

Each node stores a copy of the blockchain and performs as a *passive node* or an *active node*. The purpose of a passive node is to utilise ad-hoc messaging (e.g. a gossip protocol) with neighbouring nodes to maintain the latest version of the blockchain. A passive node does not act in the consensus of adding new blocks to the blockchain. Instead, they wait for messages from neighbouring nodes, instructing them of new blocks in order to validate them and update the blockchain.

An active node participates in the creation of new blocks. Active nodes begin providing consensus on blocks full of new transactions in order to add them to the blockchain. Once an active node contributes and transmits a new block, passive nodes verify the transactions within the block by using the Digital Signature Scheme (DSS) verification algorithm with the transaction data and the public key. Anybody can validate a transaction, as the transaction data and public key are always accessible on the blockchain [94].

The consensus layer intends to define the consensus mechanism to support the processes executed by active nodes as described above [148, 32]. There are several consensus mechanisms that decide how nodes agree on new blocks; three common approaches are Proof-of-Work (PoW), Proof-of-Stake (PoS), and Proof-of-Elapsed-Time (PoET) [88]. Consensus mechanisms make it almost negligible for adversaries to attack the network using Sybil attacks. This is where a single user hosts multiple nodes in order to obtain the majority of nodes on the network, allowing them to gain control of the network by outvoting honest nodes [133]. Another similar attack is the 51% attack, whereby an adversary gains control of 51% of the network's computing power, enabling

them to double spend the currency by building new chains off older blocks; in practice, they can build their own blockchain containing transactions chosen by them, and have the network see this as the real chain, as they control the majority of the computing power. Again, consensus mechanisms greatly lower the chances of a successful 51% attack, as it is infeasible to control 51% of the computing power [94].

Proof-of-Work With a PoW mechanism, miners compete to find a valid block against one another. PoW makes use of the puzzle-friendliness property of hash functions that Section 2.3.1 describes. The incentive for miners to carry out mining work is a block reward (a quantity of the underlying platform currency) and transaction fees they receive for mining a new block. Mining can be particularly lucrative depending on the success of a platform; the reward for mining a Bitcoin block as of December 2017 was 25 BTC, worth approximately $500,000[1]. In order to mine a new block, a node must discover the nonce n that concatenates with the block's hashed data x to provide a new hashed value y below a certain difficulty threshold D, such that

$$H(n\|x) \in \{y|y \in \mathbb{R}_{>0}, y < D\} \tag{2.1}$$

holds true [3, 94]. This difficulty value is dynamic to keep the time interval between mining blocks the same size. An approximate difficulty equation is

$$D_n = \frac{D_{n-1}NT}{t_{n-1}} \tag{2.2}$$

where D_n is the new difficulty, D_{n-1} is the difficulty of the previous block, N is the number of blocks this new difficulty will last for, T is the approximate time between mined blocks, and t_{n-1} is the time it took to mine the last N blocks. The values of N and T remain constant in the protocol. For Bitcoin, these values are 2016 and 10 minutes, respectively. This allows the difficulty to adjust approximately every two weeks in order to keep the network mining blocks every ten minutes, no matter what the hash

[1] Value calculated from the website *CoinMarketCap*.

rate[2] of the network is.

Once a node discovers a suitable nonce, it transmits this block to the rest of the network in order for it to become accepted to the consensus chain. As a P2P network, this process is not instantaneous; it takes a certain amount of time for neighbouring nodes to validate the block and update their local copies of the blockchain, and then transmit this to their neighbours. Because of this latency, it is probable that on some occasions, multiple nodes mine successful blocks simultaneous to one another. In these situations, passive nodes add the first block of which they become aware. The chain builds on whichever block the next successful miner chooses to build upon, or whichever block the network as a whole decide to accept. There is no central authority to decide this; individual nodes choose how to build their local blockchain, though it is beneficial to stay consistent with the accepted chain. Due to the competition, some blocks may become discontinued as nodes choose other blocks to mine on and become the consensus chain. These blocks are orphan blocks; Figure 2.4 illustrates this.

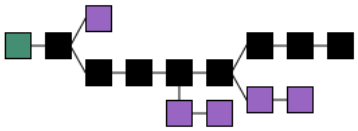


Figure 2.4: The blockchain illustrated: the green block is the genesis block; the black blocks are those on the consensus chain; the purple blocks are the rejected orphan blocks [141].

In some cases, two separate groups of nodes on the network may choose to continue two chains, disagreeing on a particular consensus chain. This is known as a fork, and there are two variations; a soft fork and a hard fork. A hard fork occurs when two chains become incompatible due to changes in the protocol. Some nodes may continue to use legacy software, while some may employ the new rules; as they are incompatible, two chains will diverge from one another. A soft fork is entirely backwards compatible with the existing chain.

PoW is a high-energy consuming task, as all miners use computational power to

[2]The hash rate of the network is the number of hashes calculated per second in order to mine a block.

find a valid block. According to Digiconimist [49] and Casino et al. [25], the Bitcoin platform uses more energy than "159 countries of the world".

Proof-of-Stake PoS selects a random active node to mine the next block based upon their stake in the blockchain [103]. If one particular node has a large amount of the associated currency, selection for this node is more likely. This mechanism reduces energy expenditure as consenters only use computational power upon selection. Furthermore, the network is more robust against a 51% attack, as it is infeasible for a single party to hold 51% of the currency.

Proof-of-Elapsed-Time PoET is a lottery-style consensus mechanism, whereby each node generates a random scalar value to determine the length of time required to wait until permission to generate a new block [30]. This type of consensus mechanism holds advantages over PoW and PoS because it is efficient and fair, as it spreads the chances of winning equally between all users. Nonetheless, it is only suitable for private and permissioned blockchain platforms as it relies upon nodes that are verified before mining, discussed further in Section 2.2.2.

Interactivity

As previously mentioned, the Interactivity layer joins the Compute Interface and Governance layers of the Casino et al. [25] model. The Compute Interface layer sits above the Base Structures layer. It provides tools to access and read the complex information stored in the raw blockchain [25]. For example, it allows users to identify their current balance with ease, as a result of identifying all historical transactions in which they have partaken. Furthermore, it allows users to initiate transactions and store information about users on the blockchain platform. This ensures that users without computer literacy may partake in the network without the need to run the command-line software. The Governance layer abstracts the human administration of the blockchain network.

This refers to the high-level decisions made regarding designing, building, and maintaining the software of a blockchain network. In many situations, it is a community of users and developers who share these decisions, though this is not a set rule; individuals and organisations may also make such decisions. The Governance layer represents each community, from open platforms through to formal committees [25].

2.2.2 State-of-the-Art

Since Nakamoto [90] introduced the concept of a blockchain with the Bitcoin platform, many avenues of research and enterprise have introduced new purposes and implementations of blockchain technology. This section discusses blockchain classification, use cases and applications, and open issues with the technology. There are five avenues of application this review discusses: *business and industry*; *data security and management*; *governance*; *integrity verification*; and *Internet of Things (IoT)*.

As blockchain ecosystems expand into new domains, public interest in the technology has also increased. From December 2016 to December 2017, searching for the word "blockchain" on Google increased by 670%[3]. This is partly due to the interest of investors intending to make a considerable gain of equity in a short period. Nonetheless, the focus is shifting to how blockchain technology can support enterprise solutions in a variety of scenarios across a plethora of industries.

Blockchain Classification

The literature suggests that there are three distinct categories of blockchain platform; public, private, and federated [23, 78, 32, 155, 25]. Casino et al. [25] provide a "fine-grained" classification of these categories that - in addition to the features of ownership and management identified by the state-of-the-art at the time of writing - characterises the platforms by transaction approval time, consensus mechanisms, and the underlying security. Table 2.8 provides a copy of the original classification table. This fails to address

[3]Value calculated from data provided by Google Trends.

Table 2.8: Classification of public, private, and federated blockchain platforms [25].

	Platform Type		
Property	Public	Private	Federated
Consensus Mechanism	Costly PoW Miners	Light PoW Centralised	Light PoW Leader Node Set
Identity Anonymity	Pseudonymous Malicious	Identified Users Trusted	Identified Users Trusted
Protocol Efficiency	Low	High	High
Energy Consumption	High	Low	Low
Immutability	Almost Impossible	Collusion Attacks	Collusion Attacks
Ownership	Public	Centralised	Semi-Centralised
Management	Permissionless	Permissioned Whitelist	Permissioned Nodes
Transaction Approval	Order of Minutes	Order of Milliseconds	Order of Milliseconds

that federated platforms are merely hybrids of public and private platforms. In this sense, federated platforms are flexible; some may be closer to public platforms, whereas some might be more similar to private platforms. Hence, the research conducted in this thesis (especially that of Chapter 4) drops the federated label and describes platforms based upon whether they are private, public, and when hybrid, to which classification they are closer.

Suitability Models and Frameworks

A decision framework is a tool for a user to discover whether their use case is suited towards using blockchain technology. Umeh [144] notes that although there is increasing interest in blockchain integration across several sectors, many do not understand the rudiments of the technology, and often mistakenly deploy it in situations where other technologies might be more appropriate. This statement is supported by Greenspan [61], who state that regular databases provide a more suitable choice in many situations due to the performance benefits. Nonetheless, this literature does not provide a supporting

framework. Moreover, a key consideration of the deployment of blockchain technology is not the mechanics, but the fact that those deciding whether to utilise the technology may not be familiar with the concepts; indeed, some may not have any technical background. Gourisetti et al. [60] notes that any decision framework needs to "assist the organizational owners, managers, engineers, and users in deciding whether blockchain is the right solution to their use case", while Maull et al. [86] highlight that "the business model thinking of managers, entrepreneurs, and developers strongly influences the technology choices they make". With these statements in mind, it is paramount that these decision frameworks condense the technical details into digestible intuitive chunks, appropriate for users ranging from those who already have expertise in the technology, to those who are beginning their ventures into blockchain.

Lo et al. [84], Wust and Gervais [150], and Maull et al. [86] present three possible frameworks to support the decision-making process of whether to use blockchain technology in a particular scenario. Lo et al. [84] use an evaluation framework with seven decision nodes based upon seven key properties: *multi-party*; *trusted authority*; *centralised operation*; *data transparency*; *data integrity*; *data immutability*; and *high performance*. Figure 2.5 illustrates this model. This results in a binary *yes* or *no* answer. This framework is tested against four use cases: supply chains; electronic health records; identity management; and the stock market. The results suggest blockchain suitability for each use case.

Wust and Gervais [150] propose a similar model that uses six decision nodes based upon six properties: *public verifiability*; *transparency*; *privacy*; *integrity*; *redundancy*; and *trust*. Figure 2.6 illustrates this model. This framework advances beyond the work of Lo et al. [84], as the result provides one of four responses; either a recommendation of which class of blockchain platform according to Table 2.8, or whether to avoid using blockchain at all. The framework is tested against nine different use cases: supply chain; bank payments; smart contracts; Decentralized Autonomous Organizations (DAOs); intellectual property; e-voting; IoT; land titles; and trading. As with Lo et al. [84], the

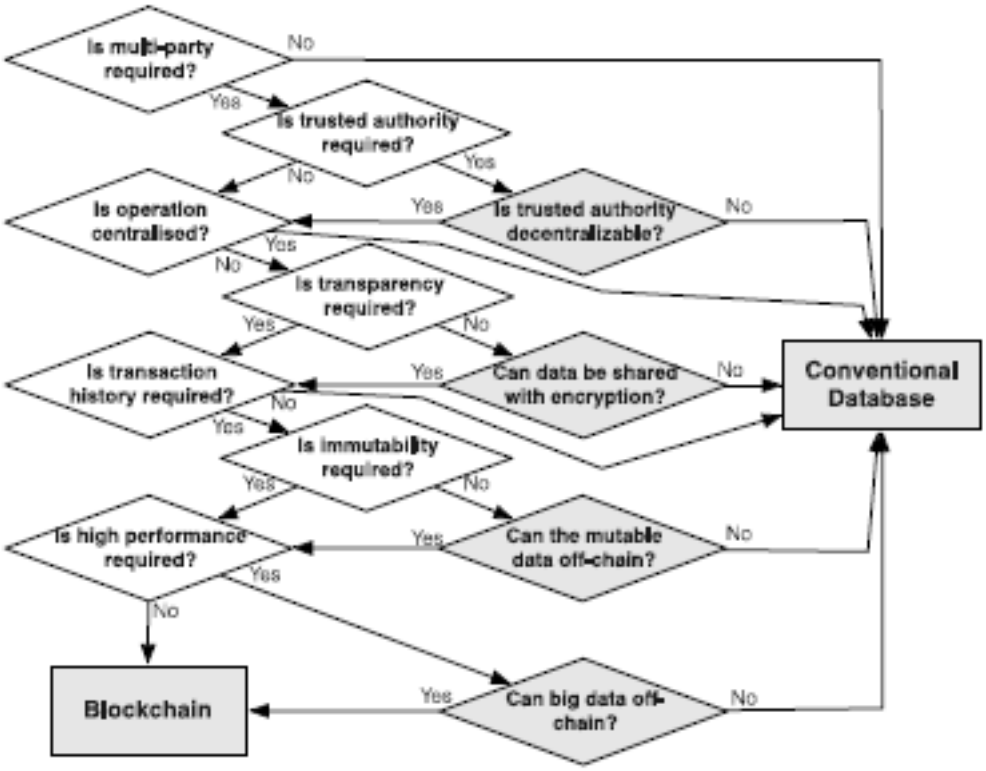


Figure 2.5: The suitability model of Lo et al. [84].

model suggests that each use case is suitable with blockchain technology.

Maull et al. [86] also offer a model in the form of a flow chart, presenting nine decision nodes. This model is similar to the model of Wust and Gervais [150] in that it offers four distinctive outputs: to avoid blockchains[4]; use a permissionless public blockchain; use a permissioned public blockchain; or use a permissioned private blockchain. Figure 2.7 illustrates this model. While Wust and Gervais [150] offer an improvement to the specificity of the result from prior models, Lo et al. [84] offer a more refined level of clarity and detail. Maull et al. [86] offer the model of most detail as of present; having developed it in cooperation with Bitcoin and Ethereum developers. However, not one framework offers a decision with a recommendation for a specific platform, and there is no literature which has addressed this issue directly.

This literature review indicates that there is a limited amount of research into frameworks to aid the decision-making process when integrating blockchain technology. One

[4]According to the original paper, the wider field of Distributed Ledger Technologies (DLT) are considered in place of blockchains.

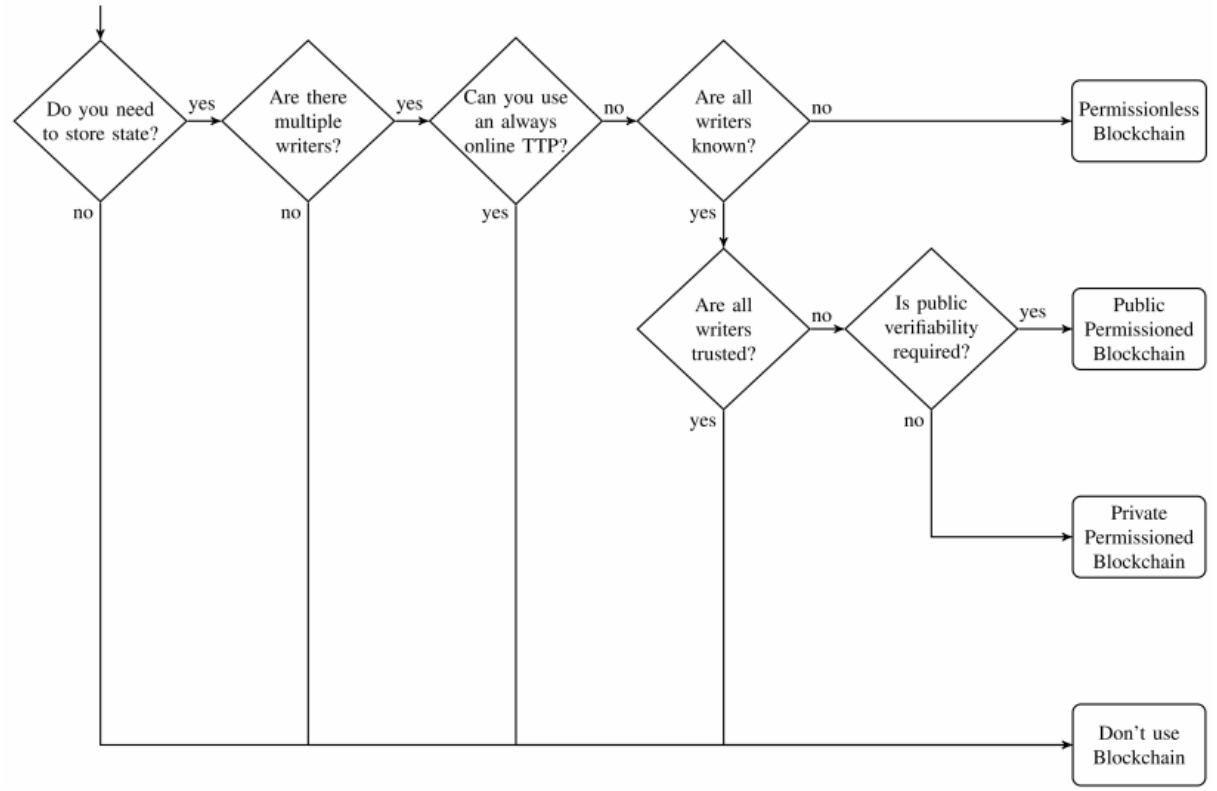


Figure 2.6: The suitability model of Wust and Gervais [150].

express flaw of the existing frameworks is that they adopt the flow chart approach. Flow charts do have noticeable advantages; one significant factor is that flow charts are intuitive to follow. This is crucial to support the notion that these frameworks must be accessible for non-technical users, as it presents the questions in a widely understood format. Moreover, the feedback is strictly quantitative, provide clear binary answers for each category. Binary information underpins the most basic of decisions, and provides intuitively presentable results for non-technical users.

However, the flowchart approach is rigid for three reasons. Firstly, the whole flowchart is visible to the users (unless presented as a number of steps via software or an interview), potentially leading to effects of implicit bias towards what the user wishes the outcome to be[5].

Secondly, the outcomes are compartmentalised. When the user arrives at an answer, it may be taken by them as the final answer to their solutions; in practice, technological

[5] As has been mentioned, many companies and individuals wish the use the technology for the sake of using the latest buzzword; not a healthy approach!



Figure 2.7: The suitability model of Maull et al. [86].

solutions are more nuanced and may require specialised hybrid approaches. This is the downside to quantitative binary feedback; though it is easy to interpret, it may not paint the entire picture. Moreover, the answers of previous models provide no additional information towards *which* platforms to use. While it is the responsibility of qualified software engineers to make final decisions upon technical decisions, managers and additional high-level decisions may be made on the principles of a philosophy, and it would be of great help to explore a qualitative[6] list of potential options.

Thirdly, users provide answers on a binary 'yes or no' basis. As soon as the user

[6]Qualitative is indicative of non-scorable feedback i.e. a list of existing platforms and projects.

answers a question that indicates a blockchain is unsuitable, the model immediately proceeds to declare a blockchain is entirely unsuitable. However, the user may have only answered a fraction of the available questions. For various use cases, differing questions may take priority. Thus the strict ordering of a flowchart may lead to poor suggestions, as the unsuitability of one property does not indicate that another property will also be unsuitable. Furthermore, the answers provided by the user may be more nuanced than just yes or no. For example, take the first question of the model from Lo et al. [84]: "Is multi-party required?" Here, multi-user fails to encapsulate whether the users are all known, trusted, or both. The models of Wust and Gervais [150] and Maull et al. [86] address this; however, they fail to ask the user of in-depth performance requirements. There is potential to exploit the nuances of a use case by designing a framework that is uses multiple-choice questions, with secret weights underlying each answer that acts towards establish new metrics to improve the feedback even further.

With these observations in mind, Chapter 4 investigates brand new avenues of aiding decisions as to whether those interested in blockchain technology should use one, and if so, which particular classification. The new model aims to utilise the features of previous models that have worked (notably the intuitiveness of the questions), whilst introducing new approaches to improve the decision making process, such as building the missing qualitative feedback and shifting from the binary quantitative feedback to a more open quantitative feedback, to provide additional metrics that capture the nuances of real use cases.

Use Cases

Though the first blockchain platforms supported the mechanisms of digital currency, many alternative platforms are emerging that focus on new applications. As of December 2019, there were over 1000 platforms listed on CoinMarketCap, a website that tracks the capitalisation values of numerous cryptocurrencies. Casino et al. [25], Narayanan et al. [92], Crosby et al. [38], Swan [132] and Zhao et al. [153] provide overviews of the

taxonomy and use cases of these emerging platforms.

Business and Industry There are a variety of business and industry use cases, spanning applications of finance, energy and sustainability, supply chains, transportation, healthcare, and education. Within each, several projects are spearheading the movement. Blockchain has disrupted the finance industry in several ways; currencies, automation of financial settlements, and prediction marketplaces [64]. Indeed, the first blockchain platform - Bitcoin - existed to remove intermediaries in financial transactions. Many platforms exist to operate as a currency, providing enhanced performance and design over Bitcoin. *Monero* is one such platform, aiming to achieve true anonymity instead of pseudonymity [101]. Another cryptocurrency is *Ripple*, an example of a cooperative approach between banks Armknecht et al. [4]. An additional area of interest within finance are prediction marketplaces; systems to aid the prediction of market movements. *Viacoin* [140] and *BitShares* [137] are two public platforms, whereas *Rizzo* is a private platform for companies trading on the National Association of Securities Dealers Automated Quotations (NASDAQ) [120].

A further area of potential use for blockchain is supply chains. Currently, supply chains rely on centralised intermediaries to share the journey of a component through the chain. The mechanics of blockchain technology suit the operations of supply chains, where a digital component mimics the journey of the physical component via transactions on the blockchain. *VeChain* is a platform that exists to enhance the supply chain management process between stakeholders [63]. The platform records every aspect of movement within the supply chain onto the distributed ledger in order to provide transparency between all parties involved. The means of asset digitisation achieves this, whereby products have unique identifiers on the platform. A technology known as VeChain Identity (VID) associates with a particular product, using Secure Hash Algorithm (SHA) to generate such a VID. The VID is accessible as one of the following forms: a Near Field Communication (NFC) tag; a Quick Response (QR) code; or a Radio Frequency Identification (RFID) tag. VeChain is currently built on the

Ethereum platform as an ERC20 standard token. However, the VeChain development team have laid out a plan to migrate the existing blockchain on the Ethereum network to its platform. *Waltonchain* and *Ambrosus* are similar platforms.

In an energy context, *Power Ledger*, *Grid+*, and *Ondiflo* work towards facilitating electricity trading, allowing the decentralised selling and buying of renewable energy traced via a blockchain [81, 28]. In the healthcare industry, blockchain is used to share electronic health records of patients. Because the records are stored in a distributed manner, various stakeholders from different organisations can share the same underlying data without going through a centralised intermediary. Platforms attempting to achieve this include *MedicalChain*, *MedRec*, *Patientory*, and *Solve.Care*. The education industry is making use of blockchain technology to share and verify academic credentials, by replicating physical certificates with digital counterparts. Because the data is distributed and tamper-proof, users can check academic credentials from around the world without having to source original transcripts or proof from the issuing institution.

The transport industry is beginning to explore how it might harness blockchain technology. Fetch.AI is seeking to enhance mobility networks by integrating transport, logistics, and supply chains via agent-based solutions to achieve coordination between stakeholders. The Blockchain in Transport Alliance (BiTA) was established to direct the discussion and set standards for integration of blockchain technology in transport. Deutsche Bahn seeks to utilise blockchain to act as an entirely digital counterpart to the railway [104].

Data Security and Management Section 2.2.1 defines a blockchain as a data structure. It is trivial to see that the transactions that form the blocks may contain additional information and data. However, issues with scaling prevent large quantities of data storage within the transactions. Christidis and Devetsikiotis [32] make note that users seeking to host large quantities of data should refer to specific systems such as *TeleHash*, *Whisper*, or the *InterPlanetary File System (IPFS)* [128, 42, 8].

Within the context of security, the threat of post-quantum adversaries is addressed

by Casino et al. [25]. They make a note of the fact that although the hash functions used within blockchain platforms are quantum-resistant, the asymmetric cryptography used is not. Platforms using Elliptic Curve Cryptography (ECC) and cryptosystems that rely on the difficulty of solving the discrete logarithm problem are prone Shor's algorithm [121]. Casino et al. [25] suggest lattice-based and code-based post-quantum cryptosystems, though provide no detail on implementation within the blockchain domain. Currently, most attempts at providing a post-quantum blockchain use hash-based signatures. Chalkias et al. [27] provide a hash-based scheme named Blockchained Post-Quantum Signatures (BPQS), based on the eXtended Merkle Signature Scheme (XMSS) proposed by Buchmann et al. [21]. This scheme provides many-time signatures with faster signing and verification than other approaches to a post-quantum blockchain, such as XMSS [111]. Similarly, van der Linde et al. [145] make use of XMSS and Naor-Yung chaining to provide more practical signatures than conventional hash-based DSSs such as SPHINCS [12]. Chapter 5 of this thesis investigates an appropriate post-quantum cryptosystem (Supersingular Isogeny Cryptography (SIC)) to work in place of ECC.

Governance Systems The combination of distributed consensus and executable smart contracts results in the possibility of creating entirely self-governing entities within a blockchain platform. One such attempt at this was performed on the Ethereum platform. The DAO was a smart contract that entitled Ethereum account holders to buy into a digital venture capital fund, and vote for where the contract would place investments. This took out the intermediaries, allowing stakeholders to make direct decisions about their money. Another avenue of research into governance is eVoting [17]. By and large, blockchain innovation "offers an open-source, shared, decentralised, and autonomous system" to "ensure the confidence required by voters and election coordinators" [33, 79, 68].

Integrity Verification Projects are using blockchain technology to digitally replicate intangible services such as insurance, ownership, and intellectual property. Blockchain

technology is disrupting several facets of the insurance industry, including sales, underwriting, claims processing, payments, asset transfers, and reinsurance Gatteschi et al. [56], Casino et al. [25]. Furthermore, many projects exist to digitally map and monetise intellectual property, such as *Ascribe*, *Mediachain*, and *Monegraph* [71].

Internet of Things According to IBM in 2017, around 90% of the data in the world was created in the preceding two years. Casino et al. [25] acknowledges the pace will lead to the adoption of the IoT in several sectors, with blockchain playing a crucial role in distributing the data between devices and users. Devices will connect to the blockchain network and send data as transactions for verification, providing secure and auditable data. Casino et al. [25] identifies that the low computational power of such devices might hamper the ability to connect with blockchain platforms. Buccafurri et al. [20] present an architecture to overcome such issues.

Generic Blockchain Platforms

There are a variety of existing blockchain platforms that fall under the classifications as described in Table 2.8. These platforms exist with flexibility in mind. Ethereum is a distributed computing platform that utilises smart contracts [22]. *Solidity* is a Turing-complete scripting language used to write the smart contracts. The native currency of the platform is Ether, which is the underlying asset transferred between the Ethereum wallets. Ethereum uses a PoW consensus mechanism, minting new Ether upon the mining of new blocks. There are plans to move to a PoS mechanism to reduce environmental impact and to increase the robustness of the network [124].

To execute a smart contract, a user sends Ether to the wallet address of the contract. The contract requires a minimum amount of Ether to execute, specified by the contract itself. This minimum amount is the amount of *Gas*. Gas determines how much processing power the contract will receive on the Ethereum Virtual Machine (EVM). This mitigates spam on the network, as contracts are costly to run and will time out if the

Gas runs out. The block time of Ethereum is approximately 15 seconds, and the network has a practical and theoretical transaction speed of 15 tps and 30 tps, respectively.

Ethereum allows for a variety of Decentralised Applications (DApps) to execute upon the Ethereum platform. Hosting a DApp on such a platform means the smart contracts can run "without any possibility of downtime, censorship, fraud or third-party interference", as the computation distributes across the nodes running the EVM on the Ethereum network. Other tokens can exist on top of the Ethereum network to transact alongside Ether. These tokens must meet Ethereum Request for Comments Proposal 20 (ERC20) protocol standards in order to work with the network. ERC20 is a set of rules that a token must follow in order to provide consistency between all other tokens operating on the Ethereum platform.

Ethereum has already had one major fork after an adversary stole a large quantity of Ether exploited by a vulnerability in a smart contract provided by The DAO. While some users decided to continue mining blocks on top of these fraudulent transactions, others decided to refund the victims with minted Ether and proceed mining from before the fraudulent transactions. A fork occurred, and split into two blockchains; Ethereum Classic and Ethereum respectively.

NEO has a similar philosophy to Ethereum. (That is to say, NEO is a platform for building DApps.) However, there are fundamental differences between the platforms, both in the protocols and the objectives. NEO will focus on becoming the preferred platform for a smart economy, meaning access to and ownership of tokens on a distributed and decentralised ledger which is still under the regulation of government. This is important for the trust and safety of law-backed assets while maintaining the distributed nature of the blockchain to avoid a centralised monopoly [67].

In terms of technical specifications, NEO uses an advanced PoS consensus mechanism known as Delegated Byzantine Fault Tolerance (dBFT). NEO token holders vote when to mine a block. Should less than 66% of the voters agree that the block is valid, the network will ignore this block and proposes a new block. The nodes proposing these

blocks are known as bookkeeper nodes, and they maintain the network. This allows NEO to scale with a higher number of transaction requests, as only the bookkeeper nodes add the blocks to the chain. This also means that no forking occurs on the NEO blockchain.

One further aspect of NEO is the decoupling of the NEO and GAS platforms. As explained above, Ether is the native currency of Ethereum, and the Gas used for computation is a small amount of Ether. NEO is different in that NEO and GAS are separate blockchains. NEO tokens are a share in the platform, in which holders can vote for bookkeepers and receive GAS dividends depending on their NEO stake. All operations on the NEO platform use GAS tokens, e.g. to pay for computation for smart contracts. There are two ways the platforms mints new GAS; Mining NEO blocks, or as dividends to existing NEO token holders.

Hyperledger Fabric is an open-source distributed ledger framework designed for enterprise solutions [2]. As a permissioned blockchain, it bypasses the global behaviours of public blockchain platforms (such as Bitcoin and Ethereum), while maintaining the core mechanics and securities that blockchain technologies provide. The framework uses a Membership Service Provider (MSP) in the form of X.509 certificates to verify the user permissions on the network. Organisations, who regulate the network policies via hosting peer nodes, use the concept of memberships to determine the rights of individuals within the network and peer nodes alike. Channels regulate who can access various ledgers hosted on a network. Hyperledger Fabric uses a modified transaction flow to that of a conventional blockchain. The *execute-order-validate-update* flow allows transactions to execute before appearing on the ledger, thereby improving throughput. As such, the throughput of Hyperledger Fabric is approximately 3,500 transactions per seconds [2], though Gorenflo et al. [59] propose an architectural modification to achieve up to 20,000 transactions per second.

2.2.3 Blockchain and Web 3.0

Web 3.0 is a term that encapsulates the evolution from a centralised World Wide Web (WWW) to a fully distributed, semantically backed network. This is in order to overcome the issues of centralisation and legacy technologies [1]. The term does come with an air of ambiguity; there is no formal definition adopted in the literature. Nonetheless, many agree that blockchain technology, with its distributed nature, will help the development and adoption of the next generation of the WWW. This section discusses approaches to the semantic web and distributed storage for Web 3.0, and the relationship of each with blockchain technology.

Ontologies: Providing Data With Semantics

Classically, ontology is a branch of metaphysics that concerns entities[7] and the relationships that may exist between them and other entities. Though a highly theoretical concept in philosophy, the principles inspired the field of ontology within the subject of information science. In this sense, an ontology is a formal model of nomenclature and definition of types, properties, and relationships of entities within a domain. Ontologies are described as "the specification of conceptualisations, used to help programs and humans share knowledge" [62].

Taxonomy Ontologies are intrinsically associated with the Semantic Web for several reasons: there is a necessity to realise that different identifiers from various data sources apply to identical concepts [9]; they provide a high level of modelling formality for data ubiquity; and the frameworks to construct an ontology are already provided by the World Wide Web Consortium (W3C), who are responsible for the standards provided for the Semantic Web.

[7] According to the Oxford dictionary, an entity is "a thing with distinct and independent existence". That is to say, an entity may take the form of a subject or object, either physically or non-physically, and either abstract or concrete.

The idea of *jelly* provides an example of contrasting identifiers relating to identical concepts [142, 143]. In the United Kingdom, *jelly* is defined as "a fruit-flavoured dessert made by warming and then cooling a liquid containing gelatin or a similar setting agent in a mould or dish so that it sets into a semi-solid, somewhat elastic mass". Meanwhile, the United States, *jelly* is defined as a "sweet spread or conserve made from fruit and sugar boiled to a thick consistency". Despite the word being identical in both branches of the English language, they are semantically independent of one another. Any program that wishes to use these two entities must know that they are semantically different. The solution would be to establish an ontology with appropriate taxonomy and inference rules in order for any system that wishes to utilise such data to understand the meaning behind the data and not just the value.

The taxonomy of an ontology is very powerful and goes beyond the ability to differentiate and identify the semantics of the underlying data as described above. By using the existing W3C standards of Resource Description Framework (RDF) and Web Ontology Language (OWL) [149], one can construct a model of a domain where the relationships between entities can be described. OWL provides the ability to define new relationship properties, and to define under which conditions such relationships can be defined. Furthermore, there is the ability to define entities as sub-classes of other entities, in which the sub-class takes the properties of the super-class, as is common in Object-Oriented Programming (OOP). With the standards provided by W3C, the ontologist can construct a concrete model of the domain they are modelling, with the relationships between entities declared, and links to external sources identified appropriately according to the domain entities in order for the model to understand the semantics behind the data.

Reasoning One of the significant properties of an ontology is its ability to infer non-explicit information from the explicitly provided information. To demonstrate, assume the relatively simple class hierarchy based upon train classification on British railways, as illustrated in Figure 2.8. There are seven classes displayed. At the top of the hierarchy

is `Train`, which has two sub-classes: `Locomotive`, which refers "a powered railway vehicle used for pulling trains"; and `MultipleUnit`, which is "a diesel or electric passenger train of two or more carriages powered by integral motors which drive several axles throughout the train". As they are sub-classes of `Train`, any instance of either class is also an instance of `Train`. The same principle applies to the sub-classes of `Locomotive` (`DieselLoco` and `ACElectricLoco`) and `MultipleUnit` (`DMU` and `EMU`). Further to these seven classes, there are two instances: `43_041` and `142_051`. The British Railways train classifications name these instances; the first two/three digits represent the class of the train, and the last three digits indicate a unique number for this class. Hence, `43_041` is train 041 belonging to Class 43, and `142_051` is train 51 belonging to Class 142. Figure 2.8 shows these instances related to their respective classes. (For clarity, the diagram shows the class numbers of trains associated with the respective classes in italics.)

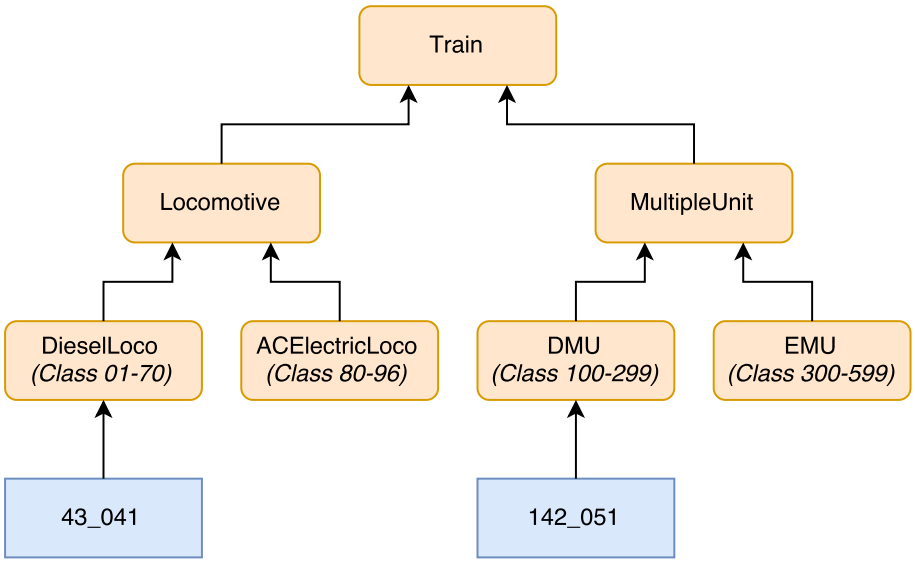


Figure 2.8: Asserted information in a simple ontology.

Figure 2.8 illustrates asserted triples (ABox)[8], because the ontologist explicitly states them. Once all the axioms of the ontology are in place, a reasoning engine classifies the ontology from the existing information to infer new information and relationships - the data store stores these as inferred triples (TBox). Figure 2.9 demonstrates how a

[8] A triple is the format describing the information for RDF. Triples consist of three components: a subject, a predicate, and an object.

reasoner would infer facts over a simple model. The existing ABox from Figure 2.8 remains, as presented by the solid arrows. However, new inferred information (TBox) is now present, as shown by the dashed arrows. It is inferred that instance `142_051` is a `Train` and a `MultipleUnit` on top of the assertion that it is a `DMU`. It is the assertion which leads the reasoning engine to make these assertions, as `DMU` is the subclass of `MultipleUnit`, which is the subclass of `Train`.

The example above demonstrates the power of reasoning. By the fact that the eight asserted triples become eight asserted triples with an additional eight inferred triples; the ontologist need not specify every bit of information about a model, as the reasoning engine can determine the extra facts. One can imagine that with a complex ontology with many relationships and rules that reasoning is a beneficial feature of determining useful information.

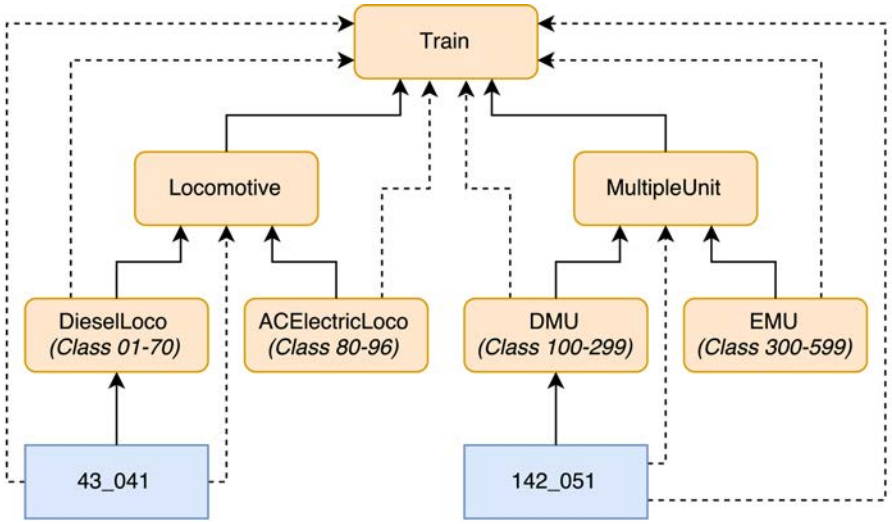


Figure 2.9: Asserted and inferred information in a simple ontology.

The approach of reasoning determines the computational intensity of an ontological reasoner. There are two major approaches, which both concern when the time of invocation of a reasoning engine: a top-down reasoner, where the reasoning engine invokes upon request; and a bottom-up reasoner, where the reasoning engine invokes upon the assertion of information into the ontology. Using a formal and declarative rule language such as Semantic Web Rule Language (SWRL) reduces systematic side-effects

that are problematic in production rule systems [85], and that reasoners built around this concept - such as the reasoner integrated into the Ontology-Driven Architecture and Service Engineering (ODASE) platform - are superior in terms of performance. A benchmark test, in which six queries are performed over demo ontologies using two reasoners: the top-down ODASE reasoner; and the bottom-up Pellet reasoner. The results are indicative of the contrasting design philosophies of the two reasoners; whereas Pellet gears towards static data, ODASE gears towards dynamic data.

Ontologies and Blockchains Many researchers utilise ontologies to provide classification and identification to components. De Kruijff and Weigand [44] provide some of the first formulations of the terminology associated with blockchain technologies, by wrapping it in an ontological model. This is because "academic literature so far is predominantly written from either a purely technical or an economic application perspective", potentially preventing access of understanding to those without a technical background. Tasca et al. [139] introduce a similar ontological model, with a particular focus on the hierarchical structures of blockchain technology. They propose an ontological matrix that "disentangle[s] [the blockchain] into building blocks". Kim and Laskowski [77] introduce a refined ontological model with a particular focus on supply-chain provenance. They propose a tractability ontology, providing the taxonomy to real blockchain platforms via smart contracts to support the transactions with semantics.

Distributed Storage

The distributed web is a term associated with transforming the WWW from a centralised model to a distributed model. Instead of accessing information stored at centralised servers, the information is distributed efficiently at every node on the network. Any node accessing information can then check locally or access remotely by locating a node that has the information. Because the network distributes the information, there is no point-of-failure, and information can be accessed even if the information's source node

is offline.

Many existing platforms allow P2P sharing. One such is BitTorrent [34], which "coordinates networks of untrusting peers" and makes up 40% of all internet traffic [31]. However, Hypertext Transport Protocol (HTTP) is the dominant file-sharing protocol and defines the majority of internet interactions this present day. Nonetheless, Benet [8] declares that HTTP falls short in several regards, stating it does not take advantage of modern distributed techniques. As such, Benet [8] introduces the IPFS. The IPFS is an open-source, peer-to-peer distributed file system, where each node on the network stores content it holds an interest in, along with indexing information to make tracking the location of other files easier. The IPFS identifies each node by a hash of a public key, where the key is generated by S/Kademlia static cryptographic puzzle and then stored in the distributed hash table. The network can make use of existing transport protocols such as HTTP, Web Real-Time-Communication (WebRTC), and Micro Transport Protocol (μTP). It introduces a new routing system that combines the technologies of Coral and Kadmelia, to maintain information to help locate information on the network, both locally and remote. Furthermore, the IPFS introduces a new block exchange protocol to govern efficient block distribution across the network. The file storage on the IPFS is analogous to Git. An object on the IPFS can store up to 256kb of data, and links to other objects. For files larger than 256kb, the file is broken down into these 256kb objects and linked via a Merkle Directed Acyclic Graph (MDAG). Though files are immutable and permanent, the IPFS supports version control and has a mutable naming system.

There are advantages that the distributed web (and implementations such as the IPFS) provide. One significant advantage is the reduction of bandwidth. Assume a video streaming from YouTube. A user performs this approach by requesting the video from the YouTube server. The same data is transferred between the server and the client every time the video is accessed. With the distributed web approach, the video is stored locally after accessing it once, removing the need to use bandwidth to stream the video further. Benet [8] estimates that the distributed approach could reduce bandwidth costs

by up to 60%.

Several authors propose protocols to use the IPFS with blockchain technology. Chen et al. [31] and Zheng et al. [154] identify the restrictive storage capacity of blockchains, and propose conceptually similar models that promote off-chain data storage on the IPFS. However, neither of these papers address the pressing issue of secure off-chain storage, where users must transfer secure data to one another. Chapter 5 investigates using distributed web technologies as a basis for storing data alongside a blockchain.

2.3 Cryptographic Prerequisites

Cryptography enables blockchains to provide 'baked-in' security. Narayanan et al. [93] identify two primitive facets of cryptography that are crucial to blockchain technology; hash functions and asymmetric cryptography. Only by understanding these primitives is one able to comprehend the mechanisms and protocols of a fully-fledged blockchain platform. This section is a technical review of the underlying cryptography fundamentals. Section 2.3.1 discusses hash functions, and various implementations thereof. Section 2.3.2 discusses symmetric cryptography. Section 2.3.3 discusses asymmetric cryptography, with particular insight into Key Exchange Schemes (KESs) and Digital Signature Schemes (DSSs), and cryptosystems of Elliptic Curve Cryptography (ECC) and Supersingular Isogeny Cryptography (SIC).

2.3.1 Hash Functions

A hash function H (illustrated in Figure 2.10) is a mathematical function which maps an input x of any binary representation length l to a an output y (known as a *hash digest*) of fixed-length n [110]. The input x can be data of any sort: a short message consisting of a small number of words; an entire document; or the resultant hash digest of another input.

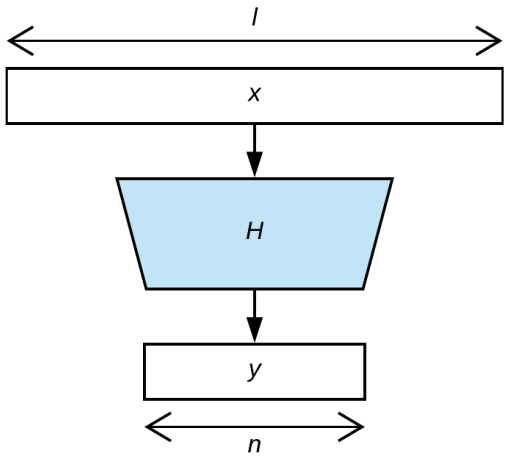


Figure 2.10: Abstraction of a hash function. An input x of arbitrary length l passes through a hash function H to form a hash digest y of a predetermined length n.

Properties of Hash Functions

Hash functions must withstand adversarial attacks. Three significant properties define the security level of a hash function; *first preimage resistance*, *second preimage resistance*, and *collision resistance*. Furthermore, hash functions suitable for use in certain blockchain platforms should have *hiding* and *puzzle friendliness* properties [93].

Preimage Resistance There are two types of preimage resistance; *first preimage resistance* and *second preimage resistance* [93, 110]. First preimage resistance states that given a hash digest y, it is infeasible to find an input x such that $H(x) = y$, where H is a hash function. Second preimage resistance states that given an input x, it is infeasible to find a second input $x' \neq x$ such that $H(x) = H(x')$. Here, "infeasible" relates to the theory of polynomial time under computational complexity theory. Simply put, should an adversary be unable to find an attack against H with lower time complexity, H is both first and second preimage resistant [122].

Collision Resistance A collision occurs when a hash function returns an identical output value y for two unique input values $x \neq x'$, such that $H(x) = H(x')|x \neq x'$. In theory, it is impossible to avoid collisions, even for the most robust hash functions. The domain of inputs is non-finite (as one can provide an input of any length), while the range of outputs has a limit, as the output length is of fixed length. For a 256-bit output, the range of distinct outputs is $2^{256} - 1$; we see that the domain of outputs is smaller than the domain of inputs. Nonetheless, H is collision-resistant if it is infeasible to find x, x' such that $H(x) = H(x')$ [93]. This means it is unlikely for a collision to occur despite being possible, and it would take an unreasonable amount of time and computing power to discover such a collision.

Hiding It must be infeasible to discover the original input value x to a hash function H with knowledge of the output value y, thereby keeping the original information passed to H secure. However, in certain scenarios, it is easier to infer the value of x. For example, in the scenario of hashing the result 'heads' or 'tails' from a coin toss, there will only be two possible values of y. The input x is deducible by running H for both possible values and then comparing any results to these computed values. To avoid this technique of working back to discover x, a random Initialisation Vector (IV) r (chosen from a high minimum entropy probability distribution) concatenates with x to 'hide' the input. Mathematically, H is hiding when it is infeasible to derive x given $H(r\|x)$, even if $x \in \mathbb{Z}$ is known and r is unknown [93].

Puzzle Friendliness This property is not essential to every hash function but plays a crucial role in blockchain platforms with Proof-of-Work (PoW) consensus mechanisms. The hash digest y must fulfil a criterion regarding the value of the output of the data. The set of acceptable values for y is Y; if y is not in Y, the hash digest y is invalid. Assuming $H(x)$ doesn't meet this criterion, one modifies x to return a different y by concatenating various values of a nonce[9] r (from a high minimum entropy probability

[9] A nonce is a number used only once.

distribution) with x until y is in Y. It is necessary that $H(r\|x) \in Y$. The hash function H is puzzle friendly if for every possible n-bit output value y then it is infeasible to find x (such that $H(i\|x) = y$) in a time less than $O(2^n)$. This means that it must take a long period to find a suitable nonce.

Quantum Resistance With Chapter 5 intending on investigating post-quantum solutions to blockchains, it is crucial to consider the security of hash functions against post-quantum adversaries. The literature suggests that hash functions are only mildly susceptible to the threat of quantum computing [29]. There are two main reasons for this statement. Firstly, generic quantum attacks achieve at most a square-root speed up compared to classical counterparts [40] and can be proven asymptotically optimal [18, 152, 69]. Secondly, no dedicated quantum attacks exist against specific hash functions; instead, only the best malicious attempts rely on generic quantum attacks [40].

Implementations

There are a variety of available implementations of hash functions. Though a thorough discussion is beyond the scope of this thesis, this section summarises two leading families of algorithms; the Message-Digest (MD) family and the Secure Hash Algorithm (SHA) family.

Message-Digest Family The MD family of hash functions is a set of four unique variants designed by Ronald Rivest by the names of MD2, MD4, MD5, and MD6. MD2 is a software-oriented algorithm that produces a 128-bit digest [75]. Developed in 1990, Rogier and Chauvaud [123] discovered collisions for a variant of this algorithm. MD4 is a related hash function, also producing a 128-bit digest [117]. It was the first algorithm to make optimal use of logical operations and integer arithmetic on particular 32-bit processors, thus differentiating it from the software-based approach of MD2. Nonetheless, the original algorithm experienced several attacks; Dobbertin [50, 51] uses genetic algorithms to produce collisions for MD4, and proves MD4 is not preimage

resistant. After these attacks, along with early attacks by Den Boer and Bosselaers [46], Rivest [118] to developed MD5 in 2006, and MD6 in 2009 as a proposal for the next generation of the SHA. Howeverm, he stated that it did not "meet our own standards for what we believe should be required of a [SHA] candidate" [119].

Secure Hash Algorithm Family The SHA family of standards is a group of cryptographic hash algorithms based upon the MD family. Industry-leading blockchain platforms such as Bitcoin and Ethereum make use of SHA algorithms [90, 22]. The National Institute of Standards and Technology (NIST) host regular open competitions to select the next generation of hash algorithm [65]. There are currently three series of algorithms; SHA-1, SHA-2, and SHA-3 [99, 100]. An initial implementation, now known as SHA-0, had an initial flaw whereby adversaries could easily discover collisions; Chabaud and Joux [26] confirm this. SHA-1 replaced SHA-0 before release, and produces a 160-bit hash digest [99]. However, SHA-1 is obsolete, with chosen-prefix attacks against it now practical [82]. The successor to SHA-1 is SHA-2, which has a set of six various hash functions that apply the Merkle-Damgård structure to produce hash digests of varying lengths [100]. The best attacks on SHA-2 work on reduced versions of the hash functions, meaning it is still safe to use. As such, many technologies (including Bitcoin) still apply functions from the SHA-2 subfamily. The successor to SHA-2 is SHA-3 (a subclass of Keccak) which uses a novel approach known as sponge construction [13]. The SHA family of algorithms remains most commonplace due to the maintenance and recommendation from NIST [65].

2.3.2 Symmetric Cryptography

Symmetric encryption is a type of encryption whereby a single key k encrypts and decrypts plaintext data $\mathcal{P}$. An encryption function E calculates a ciphertext $\mathcal{C}$ where

$$\mathcal{C} = E(k, \mathcal{P}), \tag{2.3}$$

and a decryption function D retrieves $\mathcal{P}$ where

$$\mathcal{P} = D(k, \mathcal{C}), \tag{2.4}$$

such that

$$D(k, E(k, \mathcal{P})) \equiv \mathcal{P}. \tag{2.5}$$

Symmetric cryptography is suitable as a single party who stores the key on a local system (a system only accessible by them), or by two parties who trust one another with the same key [127, 76]. Figure 2.11 illustrates this process.

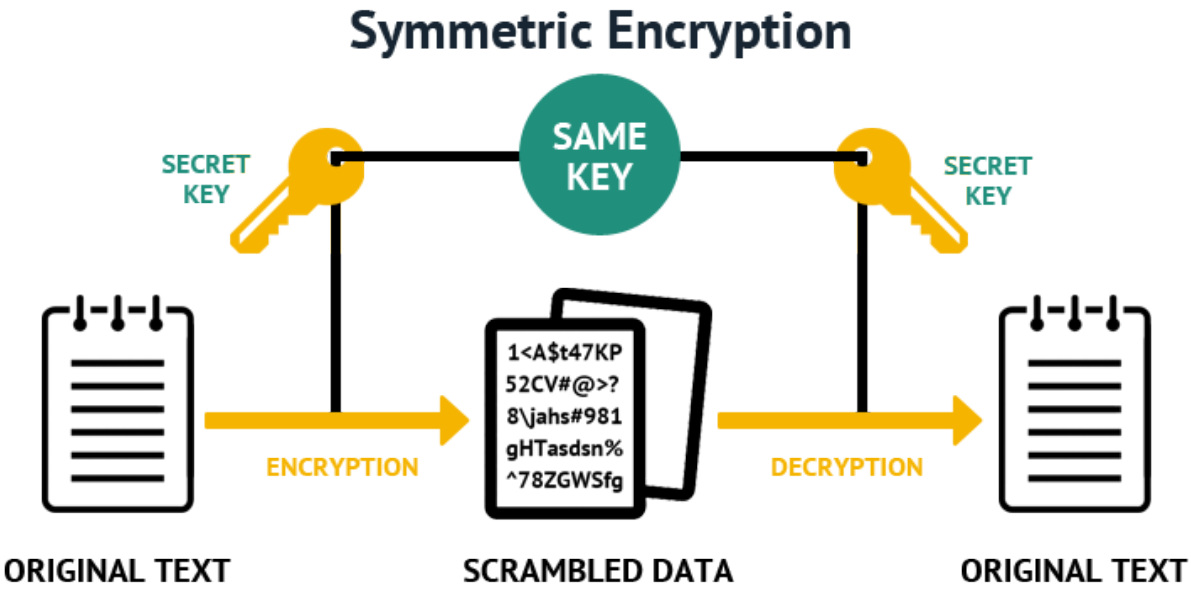


Figure 2.11: Illustration of symmetric key cryptography [130].

There are two distinct approaches to implementing symmetric cryptosystems: using stream ciphers, which encrypts each byte one-by-one; or block ciphers, which encrypts a block of bytes at the same time. There are a large number of implementations of each type. This thesis makes use of the Galois/Counter Mode (GCM) of the Advanced Encryption Standard (AES). (A detailed description of the AES is included as Appendix A, for interested readers.) As of the writing of this thesis, the literature considers most symmetric cryptography algorithms secure against post-quantum attacks [11, 10], and despite Grover's algorithm posing a threat by speeding up the attacks, increasing the key size may counteract this [10].

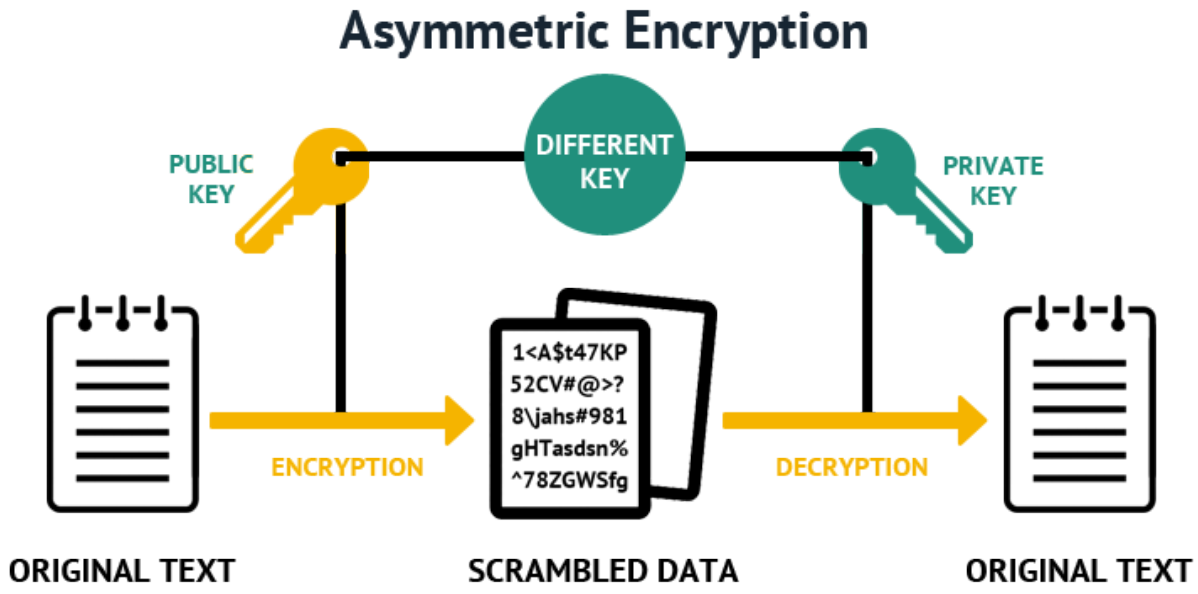


Figure 2.12: Illustration of asymmetric key cryptography [130].

2.3.3 Asymmetric Cryptography

Asymmetric cryptography - also known as public-key cryptography - is a branch of cryptography whereby two keys (a private key k and a public key K) encrypt and decrypt the plaintext $\mathcal{P}$ and ciphertext $\mathcal{C}$ respectively, and it is safe to make one of the keys known in the public domain. An encryption function E calculates a $\mathcal{C}$ where

$$\mathcal{C} = E(K, \mathcal{P}), \tag{2.6}$$

and a decryption function D retrieves $\mathcal{P}$ where

$$\mathcal{P} = D(k, \mathcal{C}), \tag{2.7}$$

such that

$$D(k, E(K, \mathcal{P})) \equiv \mathcal{P}. \tag{2.8}$$

This contrasts symmetric cryptography, whereby a single key is used in both encryption and decryption. Figure 2.12 illustrates asymmetric cryptography.

There are a large number of cryptosystems that come under the umbrella of asymmetric cryptography. A significant proportion of blockchain platforms make use

of cryptosystems based on elliptic curves. Other standard cryptosystems, such as Rivest–Shamir–Adleman (RSA) and ElGamal over prime fields, are seldom used in blockchains due to the size of the public and private keys. Hence, this section this looks into cryptosystems that use elliptic curves; namely, the classical system of ECC, and the post-quantum system of SIC. Within these cryptosystems, we address two particular schemes related to blockchain technology; KESs, and DSSs.

Mathematical Preliminaries

The following provides an introduction to the mathematical prerequisites of ECC and SIC. Elliptic curves are used in cryptography due to the efficiency of computation, and the resultant size of the keys in comparison to cryptosystems based upon other avenues of mathematics.

Elliptic Curves The Weierstrass normal form of elliptic curves defines an elliptic curve E as

$$y^2 = x^3 + ax + b | 4a^3 + 27b^2 \neq 0. \tag{2.9}$$

The restriction on the values of a and b according to Equation 2.9 exists to exclude singular curves i.e. curves with cusps and self-intersections. We define a group over E, whereby:

- the elements of the group are the points on E;
- the identity element is the point at infinity $\mathcal{O}$;
- of a selected point P, its inverse $-P$ is the point symmetric to P about the x-axis;
- provided three aligned non-zero points P, Q, R, their sum is $P + Q + R = 0$

The group is an abelian group, implying that point addition is commutative. By rearranging, we define the point addition of E such that

$$P + Q = -R. \tag{2.10}$$

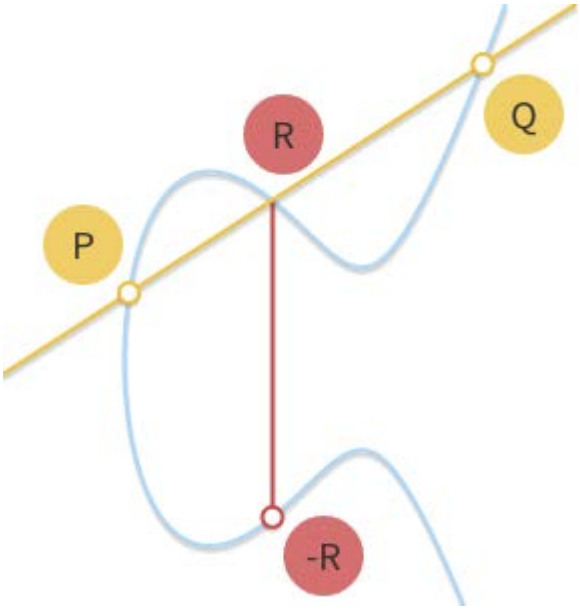


Figure 2.13: The point addition of $P + Q = -R$ over an arbitrary elliptic curve [35].

Figure 2.13 illustrates this point addition for an arbitrary elliptic curve E. It shows that $-R$ is the point symmetric to R about the x-axis, and R is the intersection of the line passing through P and Q on the curve. It is possible to add a point with itself, such that

$$P + P = -R. \tag{2.11}$$

Scalar multiplication is defined as the repetition of point addition n times on the same point P to calculate a new point Q, such that

$$Q = nP = \underbrace{P + P + ... + P}_{n}, n \in \mathbb{N}. \tag{2.12}$$

There exists an algorithm to perform these operations in $O(\log n)$, known as the double-and-add algorithm, reducing the need to perform all n additions. Equation 2.12 indicates that $Q = nP$; and that it is possible to calculate Q in $O(\log n)$. However, this is not the case should we attempt to retrieve the value of n from P and Q. This simple fact is the basis of using elliptic curves within cryptography; that computing Q is an easy problem, whereas computing the logarithm to find n is a hard problem.

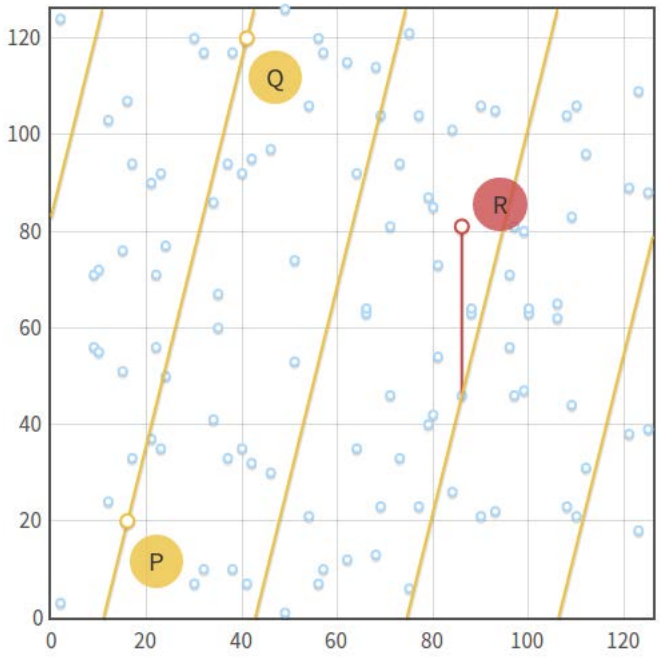


Figure 2.14: The point addition of $P + Q = -R$ over an arbitrary elliptic curve in a finite field $\mathbb{F}_p$ [35].

Elliptic Curve Cryptography We define a curve E over a finite prime field $\mathbb{F}_p$ as

$$\{(x, y) \in (\mathbb{F}_p)^2 \mid y^2 \equiv x^3 + ax + b \pmod{p}, 4a^3 + 27b^2 \not\equiv 0 \pmod{p}\} \cup \{\mathcal{O}\}, \tag{2.13}$$

where a and b are integers in $\mathbb{F}_p$, and $\mathcal{O}$ is the point at infinity. ECC uses elliptic curves over finite fields to provide a restricted domain of possible points. Point addition holds for points in the finite field; Figure 2.14 illustrates this. Scalar addition also holds under a finite field, forming cyclic subgroups. In a cyclic subgroup, the original point P is referred to as the *base point*. The order of the the base point is the smallest positive integer n such that

$$nP = 0. \tag{2.14}$$

Furthermore, Lagrange's theorem states that the order of a subgroup n is a divisor of the order of the parent group N, and the cofactor is given as

$$h = \frac{N}{n}. \tag{2.15}$$

ECC curves use a number of public parameters: the order p of the field $\mathbb{F}_p$; the values of a and b as denoted in Equation 2.9; a point G as a generator of the cyclic subgroup on the curve E; the order n of this subgroup; and a cofactor h of this subgroup. For ease, these domain parameters D are henceforth referred to as

$$D = (p, a, b, G, n, h). \tag{2.16}$$

A secret key d and a a public key Q form an elliptic curve keypair (d, Q), such that

$$d \in \mathbb{R} \mid 1 < d < n - 1, \tag{2.17}$$

and

$$Q = dG. \tag{2.18}$$

It is feasible to calculate Q from knowing d and G, but hard to calculate d from knowing Q and G due to the discrete logarithm problem [126].

Supersingular Isogeny Cryptography Shor's algorithm is a hypothetical quantum algorithm that computes discrete logarithms to break the underlying ECC, with an estimate of 2330 qubits and 126 Toffoli gates required to break a 128-bit level of security [121]. As such, several alternative post-quantum cryptosystems exist to replace classical cryptosystems if necessary. One such cryptosystem places itself as the natural successor to ECC; SIC. Jao and De Feo [72] use supersingular isogenies of elliptic curves, advancing beyond the mathematics of ECC.

SIC defines the algorithms and curves for protocols using supersingular isogenies. Assume two elliptic curves E_1 and E_2 over a quadratically extended finite field $\mathbb{F}_{p^2}$. An isogeny between these two curves $\phi : E_1 \rightarrow E_2$ is a surjective and homomorphic rational map between domain curve E_1 to the co-domain curve E_2. The group law of elliptic curves preserves the point at infinity $\mathcal{O}$. The kernel of an isogeny κ is the degree of its rational map. It is necessary that the degree of E_1 and E_2 over $\mathbb{F}_p$ is identical. A common method of calculating isogenies is using Velu's formula [146].

An isogeny $\phi : E_1 \to E_2$ with degree $|\kappa|$ has a unique dual isogeny $\hat{\phi} : E_2 \to E_1$ of identical degree, whereby the composition $\hat{\phi} \circ \phi : E_1 \to E_2$ is the multiplication map from a point P, such that $P \to [|\kappa|]P$. This is the endomorphism ring of the isogeny. An elliptic curve E is supersingular if its endomorphism ring is isomorphic to an order in a quaternion algebra. A supersingular curve E over $\mathbb{F}_p$ is isomorphic over $\mathbb{F}_{p^2}$.

The j-invariant of an elliptic curve E is a descriptor of a the curve calculated as

$$j(E) = 1728 \left(\frac{4a^3}{4a^3 + 27b^2} \right). \tag{2.19}$$

It is important to note that this is the form used for elliptic curves provided in the Weierstrass form, where $y^2 = x^3 + ax + b$. Elliptic curves that are isomorphic to one another share an identical j-invariant [87, 126, 112].

SIC defines public parameters: a prime p such that

$$p = 2^{e_A} 3^{e_B} - 1, 2^{e_A} \approx 3^{e_B}; \tag{2.20}$$

a supersingular elliptic curve E over $\mathbb{F}_{p^2}$, and four fixed points P_A, Q_A, P_B, Q_B on E. During a SIC interaction between two users Alice and Bob, Alice computes a secret point S_A such that

$$S_A = P_A + [k_A]Q_A, k_A \in [0, 2^{e_A}), \tag{2.21}$$

and Bob computes a secret point S_B such that

$$S_B = P_B + [k_B]Q_B, k_B \in [0, 3^{e_B}). \tag{2.22}$$

Key Exchange Schemes

Asymmetric cryptosystems derive a shared secret key between two parties for use in symmetric cryptography operations. Figure 2.15 presents an abstracted form of key exchange using the theory of colour in place of mathematics, the objective of which is to share a common key (or colour) via negotiation. Alice and Bob begin by choosing a secret key, analogous to a secret colour. The pair wish to produce a common colour,

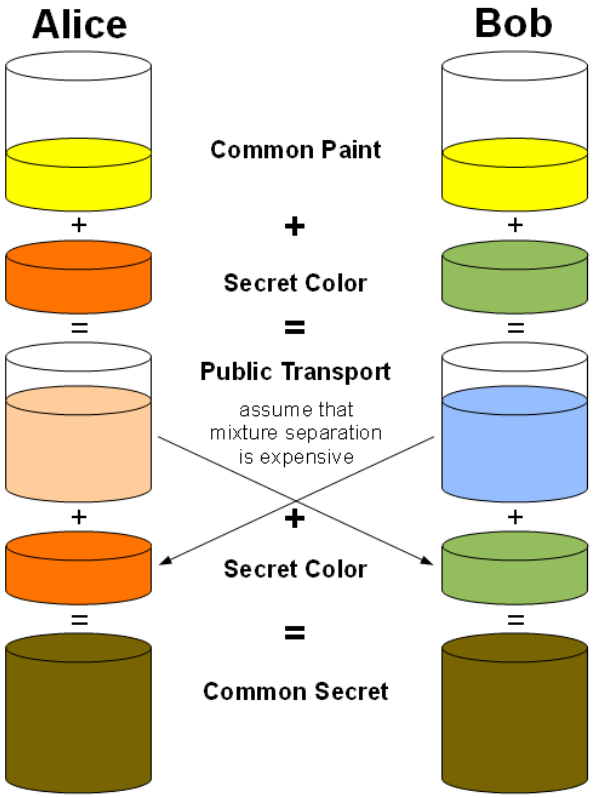


Figure 2.15: Abstraction of a key exchange protocol. The paint mixture represents mathematical operations [147].

analogous to a shared key. To achieve this, they agree on a common paint colour (the public key) and mix their secret colours with this common colour to derive a unique colour. They exchange these new colours between one another and then mix in their secret colours. By doing this, they both end up with the common secret, without having to disclose their secret colours. Because paint separation is an expensive process, the secret colours are irretrievable. Eve - an adversary watching the process - only sees the exchanged colours of the common paint and mixtures, meaning she cannot establish the common secret or either of the secret colours.

Diffie-Hellman Key Exchange The earliest method of key exchange is the Diffie-Hellman Key Exchange (DHKE), proposed in 1976 by Diffie and Hellman [48]. Each party establishes the secret independent of one another, removing the need to send sensitive data (i.e. the secret key) across an unprotected network.

Elliptic Curve Diffie Hellman Key Exchange Elliptic Curve Diffie Hellman Key Exchange (ECDH) is a modification of the original DHKE which makes use of ECC. Assume Alice and Bob have elliptical curve key-pairs (Q_A, d_A) and (Q_B, d_B) respectively, where d is a secret key and Q is a public key. Furthemore, assume a generator point G. The process of ECDH is as follows:

1. (a) i. Alice generates a random scalar d_A
   ii. Alice computes a point $Q_A = d_A G$
   iii. Alice sends Q_A to Bob
   (b) i. Bob generates a random scalar d_B
   ii. Bob computes a point $Q_B = d_B G$
   iii. Bob sends Q_B to Alice
2. (a) i. Alice computes the point $R = d_A Q_B$
   ii. Alice uses the x-coordinate of R as the shared key
   (b) i. Bob computes the point $R = d_B Q_A$
   ii. Bob uses the x-coordinate of R as the shared key

Because $d_A Q_B = d_A d_B G = R$ and $d_B Q_A = d_B d_A G = R$, we see that $d_A Q_B = d_B Q_A = R$, and that the shared secret R is equal for both parties. The Elliptic Curve Integrated Encryption Scheme (ECIES) is an integrated encryption scheme that uses ECDH in order to establish a symmetric key; Figure 2.16 illustrates this at a high-level.

Supersingular Isogeny Diffie Hellman Key Exchange A post-quantum solution to DHKE is Supersingular Isogeny Diffie Hellman Key Exchange (SIDH). As with the ECDH, certain parameters are public prior to use: a prime p; a supersingular elliptical curve E; fixed elliptical points (P_A, Q_A) and (P_B, Q_B); and the orders of the points 2^{e_A} and 3^{e_B}.

The SIC combines the mathematics of supersingular isogenies with the protocols of DHKE to form the SIDH. Assume Alice and Bob wish to derive a shared key k to

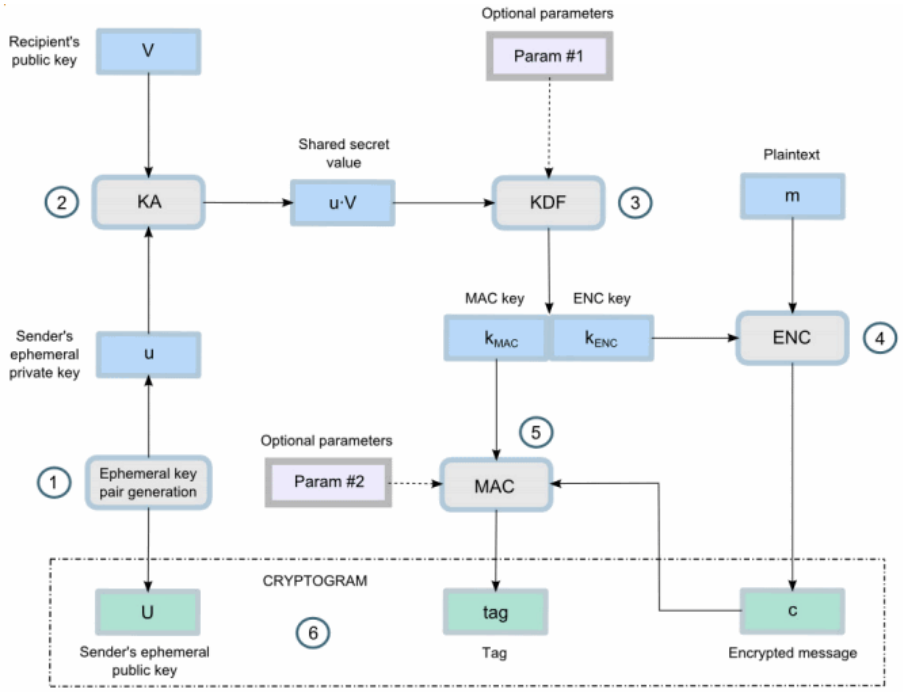


Figure 2.16: High-level overview of the ECIES [91].

encrypt data. The protocol runs the following process:

1. (a) i. Alice generates a random scalar $k_A \in [0, 2^{e_A})$
   ii. Alice computes a point $S_A = P_A + [k_A]Q_A$
   iii. Alice computes the isogeny $\phi_A : E \to E/\langle S_A \rangle$
   iv. Alice computes the points $P'_B = \phi_A(P_B)$ and $Q'_B = \phi_A(Q_B)$
   v. Alice sends $[E/\langle S_A \rangle, P'_B, Q'_B]$ to Bob

   (b) i. Bob generates a random scalar $k_B \in [0, 3^{e_B})$
   ii. Bob computes a point $S_B = P_B + [k_B]Q_B$
   iii. Bob computes the isogeny $\phi_B : E \to E/\langle S_B \rangle$
   iv. Bob computes the points $P'_A = \phi_B(P_A)$ and $Q'_A = \phi_B(Q_A)$
   v. Bob sends $[E/\langle S_B \rangle, P'_A, Q'_A]$ to Alice

2. (a) i. Alice computes $S_{AB} = P'_A + [k_A]Q'_A$
   ii. Alice computes the isogeny $\phi_{AB} : E/\langle S_A \rangle \to E/\langle S_A, S_{AB} \rangle$
   iii. Alice computes the j-invariant $j_{AB} = j(E/\langle S_A, S_{AB} \rangle)$

iv. Alice now has the shared key $K_A = j_{AB}$

(b) i. Bob computes $S_{BA} = P'_B + [k_B]Q'_B$

ii. Bob computes the isogeny $\phi_{BA} : E/\langle S_B \rangle \to E/\langle S_B, S_{BA} \rangle$

iii. Bob computes the j-invariant $j_{BA} = j(E/\langle S_B, S_{BA} \rangle)$

iv. Bob now has the shared key $K_B = j_{BA}$

Because the curves $E/\langle S_A, S_{AB} \rangle$ and $E/\langle S_B, S_{BA} \rangle$ are isogenous to one another, the values of the j-invariant are equal. Thus Alice and Bob arrive at the shared key. Biasse and Jao [14] Galbraith et al. [54] confirm that SIDH has a classical security of $O(p^{1/4})$ and Jao and De Feo [72] hypothesises that the quantum security is $O(p^{1/6})$. This suggests that the security level is 128-bit if the p is a 768-bit prime number. Supersingular Isogeny Key Exchange (SIKE) is an implementation of SIDH that specifies safe parameter values to use during the protocol, and encapsulates the keys to protect against man-in-the-middle attacks [74]. Galbraith et al. [54] identify that if Alice or Bob use a secret key for many instances, then an attacker can actively learn the secret key by performing n interactions, where n is the bitlength of the key. To overcome this, Bob uses a secret key s_B that depends on Alice's public key K_A such that

$$s_B = H(K_A, m), \tag{2.23}$$

where H is a hash function and m is a random scalar value. For Alice to derive s_B, Bob sends K_B and encapsulates the hash of j_{BA} (from the SIDH protocol) with m using XOR. That is to say, Bob sends

$$(K_B, H(j_{BA}) \oplus m). \tag{2.24}$$

Alice retrieves the value of m by calculating $H(j_{AB}) = H(j_{BA})$ and computing the value. Thus, Alice confirms that s_B is correct.

Digital Signature Schemes

A digital signature is analogous to a physical signature; the signature is unique to the signer and verifiable by anybody. A keypair helps perform this: a secret key k, which only the signer knows and uses to sign the message m; and a public key K, which anybody may know and validates any signatures made. A signature algorithm S generates the signature σ such that $S(k, m) \rightarrow \sigma$. The signature σ is then verified using the verification algorithm V such that $V(K, m, \sigma) \rightarrow \{\top, \bot\}$ i.e. the verification algorithm confirms signature ownership as true $\top$ or false $\bot$.

Elliptic Curve Digital Signature Algorithm The Elliptic Curve Digital Signature Algorithm (ECDSA) works by using the three algorithms; *key generation*, *sign*, and *verify*. Peggy (the prover) wishes to prove that a message m was signed by her. Firstly, she generates an ECC keypair via `KeyGeneration`, making use of the public parameters described in Equation 2.16; Algorithm 2.1 demonstrates this process. Once Peggy has obtained her keypair d, Q, she uses the `Sign` algorithm (Algorithm 2.2) to sign m.The algorithm returns a signature consisting of two values (r, s). Vic (the verifier) wishes to verify that Peggy is the owner of m, so uses the `Verify` algorithm in Algorithm 2.3 along with (r, s) to authenticate Peggy.

Algorithm 2.1: The `KeyGeneration` algorithm for the ECDSA.

func `KeyGeneration(`D`)`**:**
  $d \xleftarrow{R} \{1, ..., n-1\}$
  $Q = dG$
  return d, Q

Supersingular Isogeny Digital Signature Algorithm Jao and De Feo [72] lay the groundwork for the SIC, and there have since been many developments for various DSSs built using the SIC: Sun et al. [131] present research addressing strong designated verifier signatures; Jao and Soukharev [73] propose isogeny-based undeniable signatures; and Srinath and Chandrasekaran [129] present an undeniable blind signature

Algorithm 2.2: The `Sign` algorithm for the ECDSA.

func `Sign`(d, m)**:**
- $s \xleftarrow{R} \{0, ..., n-1\}$
- $R = sP$
- $r = R_x \mod n$, where R_x is x-coordinate of R
- $e = H(m)$, where H is a hash function
- $s = k^{-1}(e + dr)$
- **return** (r, s)

Algorithm 2.3: The `Verify` algorithm for the ECDSA.

func `Verify`(r, s, m)**:**
- **if** $\neg(1 < r < n-1) \vee \neg(1 < s < n-1)$ **then**
  - **return** $\perp$
- $e = H(m)$
- $\omega = s^{-1} \mod n$
- $u_1 = e\omega \mod n$
- $u_2 = r\omega \mod n$
- $X = u_1 G + u_2 Q$
- $v = X_x \mod n$
- **if** $v = r$ **then**
  - **return** $\top$
- **else**
  - **return** $\perp$

scheme. There have been two independent proposals of a DSS that allows for general-authentication; Yoo et al. [151] base the DSS on a Non Interactive Zero Knowledge Proof (NIZKP) proposed by Iovino and Visconti [70], whereas Galbraith et al. [54] base their DSSs on endomorphism rings and the NIZKP. The latter DSS of Galbraith et al. [54] uses an identical concept to the DSS proposed by Yoo et al. [151]; however, the security levels provided are only classical, and there are no benchmarks. This work makes use of Unruh's construction [70], whereby the authors modify an interactive Zero Knowledge Proof (ZKP) detailed by Jao and Soukharev [73] into the NIZKP.

Assume Peggy wishes to prove her knowledge of a generator S for a kernel $\langle S \rangle$ to a verifier by the name of Vic. The protocol runs the following process:

1. (a) Peggy chooses a random scalar $k_A \in [0, 2^{e_A})$ and computes a point $S = P_A + [k_A]Q_A$

(b) Peggy computes her public key as the isogeny $\phi : E \to E/\langle S \rangle$

2. (a) Peggy chooses a random scalar $k_B \in [0, 3^{e_B})$ and computes a point $R = P_B + [k_B]Q_B$

   (b) Peggy computes the isogeny $\psi : E \to E/\langle R \rangle$ with kernel $\langle R \rangle$

   (c) Peggy computes the isogeny $\phi' : E/\langle R \rangle \to E/\langle R, S \rangle$ with kernel $\langle \psi(S) \rangle$

   (d) Peggy computes the isogeny $\psi' : E/\langle S \rangle \to E/\langle R, S \rangle$ with kernel $\langle \phi(R) \rangle$

3. Peggy sends a commitment $C = [E/\langle R \rangle, E/\langle R, S \rangle]$ to Vic

4. Vic chooses a challenge bit $b \in \{0, 1\}$ and sends it to Peggy

5. (a) If $b = 0$, Peggy sends a response $\rho = [R, \phi(R)]$

   (b) If $b = 1$, Peggy sends a response $\rho = \psi(s)$

6. (a) If $b = 0$, Vic verifies that R and $\phi(R)$ are of the order of the order 3^{e_B} and generate the kernels of the isogenies ψ and ψ' respectively

   (b) If $b = 1$, Vic verifies that $\psi(S)$ is of the order of 2^{e_A} and generates the kernel of the isogeny ϕ'

By using classical methods [53, 58], Jao and Soukharev [73] proves that this ZKP satisfies completeness, soundness, and zero-knowledge. To achieve λ bits of security, the protocol must run 2λ times [151] for reasons indicated by Section 5.1 in Yoo et al. [151].

By utilising the non-interactive ZKP detailed by Iovino and Visconti [70], the Supersingular Isogeny Digital Signature Algorithm (SIDSA) is strongly unforgable under a chosen message attack and that satisfies completeness and special soundness. The SIDSA consists of three algorithms: `KeyGeneration` for generating keys; `Sign` for signing a message m; and `Verify` for verifying that a signature σ signed m. Algorithms 2.4, 2.5, and 2.6 illustrate these respectively.

The SIDSA proposed by Yoo et al. [151] offers an average signature size of signature size of $122,880$ bits. The following components for each round of the ZKP protocol

Algorithm 2.4: The SIDSA `KeyGeneration` algorithm [151].

func `KeyGeneration()`**:**
- Generate a random scalar value $k_A \in [0, 2^{e_A})$
- Compute a secret point $S \leftarrow P_A + [k_A]Q_A$
- Assign prv $\leftarrow S$
- Compute the secret isogeny $\phi : E \rightarrow E/\langle S \rangle$
- Assign pub $\leftarrow E/\langle S \rangle$
- **return** prv, pub

compose the signature σ: two compressed curves $E/\langle R \rangle$ and $E/\langle R, S \rangle$, consisting of one field element of 12λ bits each; a challenge bit of 1 bit; and (depending on the challenge bit) either a response of $[R, \phi(R)]$ of 3λ bits, or $\psi(S)$ of 12λ bits. The algorithm compresses the response $\psi(S)$ to 3λ bits instead. On average, the required number of bits for a compressed signature is

$$2\lambda \left(12\lambda + 12\lambda + 1 + 3\lambda + \frac{3\lambda + 3\lambda}{2} \right) = \lambda \left(60\lambda + 2 \right), \tag{2.25}$$

The approach by Galbraith et al. [55] is similar in conceptual approach. However, the j-invariant replaces the field element. Thus the required number of bits is

$$2\lambda \left(3\lambda + 3\lambda + 1 + 3\lambda + \frac{3\lambda + 3\lambda}{2} \right) = \lambda \left(24\lambda + 2 \right). \tag{2.26}$$

There is the potential to compress this value further by taking the hash value of the j-invariant. By doing so, the space required is reduced, and security is not affected as the j-invariant is in the public domain. Chapter 5 investigates this.

2.4 Data and the Rail Industry

This section discusses the state-of-the-art of data solutions in the rail industry. Section 2.4.1 explains the various attempts at achieveing ubiquitous data in the railway industry. Section 2.4.2 introduces the Rail Settlement Plan (RSP), and the data architectures used for developing statistics for the rail network in Great Britain (GB). Finally, Section 2.4.3 analyses the approach to ticketing in GB, from both a technical and

Algorithm 2.5: The SIDSA `Sign` algorithm [151].

func `Sign(prv, message)`**:**
  for $i \leftarrow 0$ **to** $2\lambda - 1$ **do**
    Generate a random scalar value $k_B \in [0, 3^{e_B})$
    Compute the point $R \leftarrow P_B + [k_B]Q_B$
    Compute the isogeny $\psi : E \rightarrow E/\langle R \rangle$
    Compute the isogeny $\phi' : E/\langle R \rangle \rightarrow E/\langle R, S \rangle$
    Compute the isogeny $\psi' : E/\langle S \rangle \rightarrow E/\langle R, S \rangle$
    Assign commitment$[i][0] \leftarrow E/\langle R \rangle$
    Assign commitment$[i][1] \leftarrow E/\langle R, S \rangle$
    Generate a bit $b \xleftarrow{\text{rand}} \{0, 1\}$
    Assign challenge$[i] \leftarrow b$
    if challenge$[i] = 0$ **then**
      response$[i][0] \leftarrow (R, \phi(R))$
      response$[i][1] \leftarrow \psi(S)$
    else
      response$[i][0] \leftarrow \psi(S)$
      response$[i][1] \leftarrow (R, \phi(R))$
    hash$[i][0] \leftarrow G($response$[i][0])$
    hash$[i][1] \leftarrow G($response$[i][1])$
  end
  Generate a stream of bits bits $\leftarrow H($pub, message, commitment, challenge, hash$)$
  for $i \leftarrow 0$ **to** $2\lambda - 1$ **do**
    signature$[i] \leftarrow$ commitment$[i]$, challenge$[i]$, hash$[i]$, response$[i][$bits$[i]]$
  end
  return signature

passenger-oriented standpoint.

2.4.1 Ubiquitous Data

One issue at the forefront of research into Web 3.0 technologies is the topic of data interoperability. Data interoperability is the ease at which data from several different sources and formats can interact with one another. Due to the vast number of Information Technology (IT) systems within the railway industry, the level of interoperability is low; systems are designed independently of one another, leading to varying formats for the same concepts. There have been several separate projects that have attempted to address the issue of data unification.

Algorithm 2.6: The SIDSA `Verify` algorithm [151].

```
func Verify(pub, message, signature):
    bits ← H(pub, message, commitment, challenge, hash)
    for i ← 0 to 2λ − 1 do
        if hash[i][0] = H(response[i][0])  &  hash[i][1] = H(response[i][1]) then
            if challenge[i][0] = 0 then
                Check R has an order of 3^{e_B}
                Compute the isogeny ψ : E → E/⟨R⟩
                Check commitment[i][0] = E/⟨R⟩
                Compute the isogeny ψ' : E/⟨S⟩ → E/⟨R, S⟩
                Check commitment[i][1] = E/⟨R, S⟩
            else
                Check ψ(S) has the order 2^{e_A}
                Compute the isogeny φ : E → E/⟨ψ(S)⟩
                Check commitment[i][1] = E/⟨R, S⟩
        else
            return false
    end
    return true
```

The *Intelligent Integration of Railway Systems (InteGRail)* was one of the first projects to tackle the problem of data integration within the railway industry, to "enable information to be shared within the Railway World to enable optimisation of decision making for improved performance" [80]. The InteGRail achieves this by identifying the information that needs sharing, ensuring correct information is obtainable, and by identifying methods of combining and using information for management requirements. Furthermore, it ensures that information can be transmitted effectively to decision-makers. The InteGRail intended on demonstrating significant performance improvement by creating and sharing appropriate information, and never to replace systems; rather, to be used in conjunction with them [80]. A visualisation of this methodology is illustrated in Figure 2.17, where a cyclic model infers the idea of continuously improving the model. Unfortunately, InteGRail seemed to be ahead of its time; while theoretically grounded, it would have been difficult to implement with the technologies of the era and thus no implementations exist at a practical level.

Railway Markup Language (RailML) is an eXtensible Markup Language (XML) based data exchange format. Its purpose is to provide a universal format for railway applica-

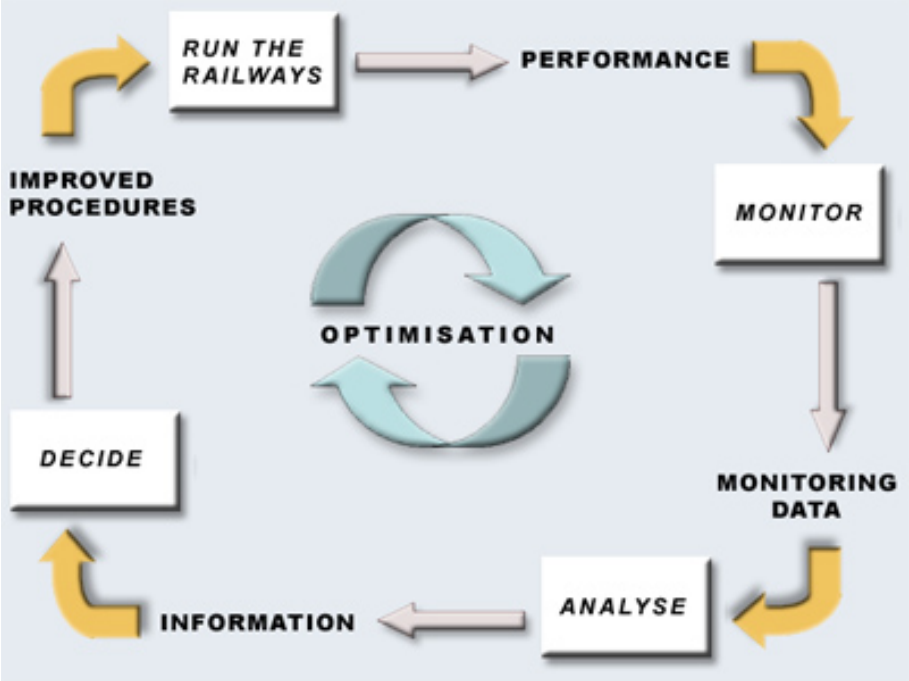


Figure 2.17: The InteGRail methodology [80].

tions to ensure interoperability between them. Hansen [66] describes how if there are n bespoke applications that require exchanging data with one another, then

$$N = \frac{n^2 - n}{2}, \tag{2.27}$$

where N is the total number of interfaces between applications required. This quadratic relationship results in a vast number of interfaces required for domains such as the railway industry, where 100 unique systems would result in 4950 interfaces. To implement this number of systems would be a costly, time-consuming, and convoluted approach to interoperability. Instead, organisations should use the enterprise application integration model, in which a universal exchange format is supported by all systems, reducing N to be n. In the railway industry, RailML is this enterprise application. Figure 2.18 demonstrates the benefits of a universal exchange format.

RailML has schemes for timetabling, infrastructure, and rolling stock. Within its lifetime, it has mainly been implemented in computer programmes, including: *OpenTrack* (railway simulator); *Viriato* (scheduling system); *OpenTimeTable* (real-time analysis of network operation data); and *railVIVID* (validates RailML files). It is currently on

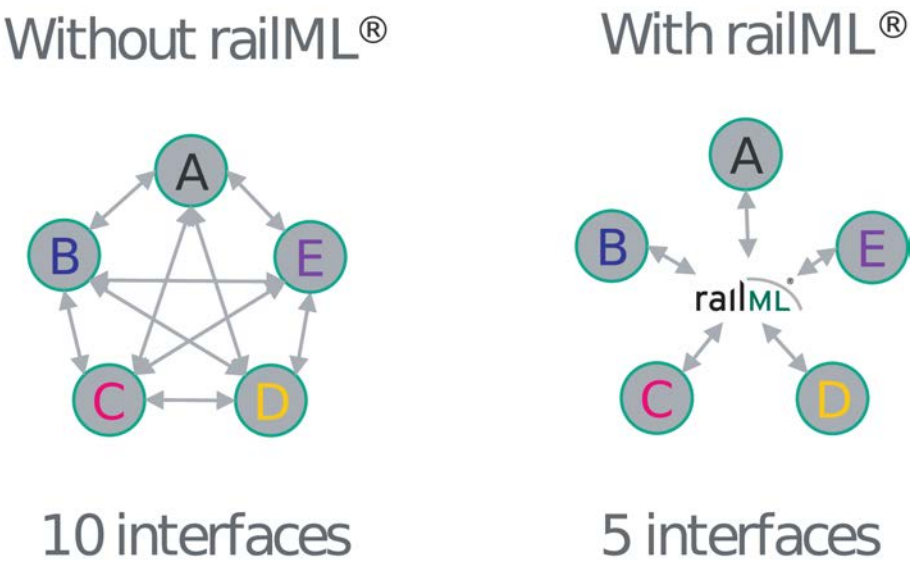


Figure 2.18: The benefits of utilising a universal exchange format [113].

Version 3 [113] and is continually expanding, especially within the European railway community.

RailTopoModel is a model for topology-based railway infrastructure that exists alongside RailML. Their website describes it as follows [115]:

> "The RailTopoModel is a logical object model to standardise the representation of railway infrastructure-related data. Together with railML, which defines the schema for the exchange of data, it will revolutionise the sharing of information in the railway industry".

This quote indicates that RailTopoModel models the railways, whereas RailML is the exchange format. RailTopoModel is developed per the International Union of Railways (UIC), and exists as International Railway Standard (IRS) 30100. The IRS demonstrates the intention to become a standard format across railway applications.

Alongside these attempts at data unification is research that focuses on using an ontology as a mechanism to relate data from various sources through their underlying semantic meaning as opposed to their raw format. Inspired by semantic data models in other enterprise environments, the *Railway Core Ontology (RaCoOn)* is an ontology tailored towards the railway industry, allowing the integration of information from

diverse and heterogeneous sources [142]. The RaCoOn comprises of four tiers: *upper* (cross-domain terms from existing ontologies); *core* (core railway industry terminology); *subdomain* (timetable, rolling stock, infrastructure relationships); and *application specific* (applications that reference RaCoOn for their own use cases). This is illustrated in Figure 2.19.

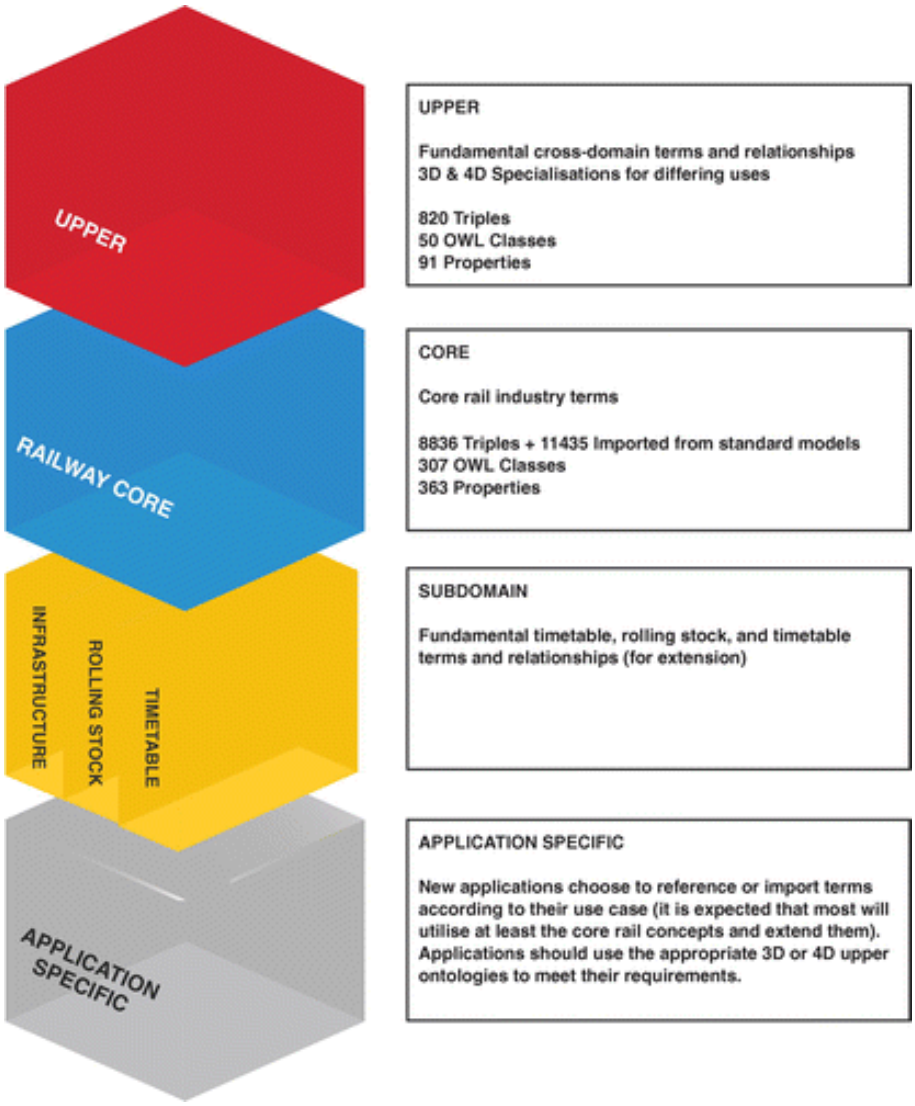


Figure 2.19: The RaCoOn architecture [143].

Tutcher et al. [142] present two use cases of RaCoOn, demonstrating the abilities of ontologies within the railway industry. One discusses legacy system functionality, and the other discusses degraded system functionality.

For the first use case, assume a railway has legacy train positioning systems, such as

track circuits or axle counters. The data produced by these systems supply passenger information systems. However, if the track migrates to European Railway Traffic Management System (ERTMS) technology, this will be unable to supply the same format of information to the legacy system. Furthermore, it would be a costly process to update all of the systems reliant on the legacy format. By utilising an appropriate ontology, the model knows what data is available, and can infer legacy formats from new data sources through the use of rules written into the ontology. The ontology allows new and old systems can coexist whilst the prioritised updates roll out across the network.

For the second use case, assume a railway line upgrades to ERTMS technologies and provides location data for each train, rather than only track circuit data. Developers develop new applications using new data formats. However, specific systems must continue functioning in times of degraded operations, e.g. if ERTMS systems are unavailable and the line reverts to fixed block operation. The standard approach would be to application logic decide specified at system design time as to which data source to select. However, the use of an ontology would improve upon this by providing flexibility for current and future systems.

Theoretical approaches to ontologies are well established within the academic domain. With the notions of their capabilities now commonplace, an emerging area of interest is the integration of an ontological model within a physical system architecture. Furthermore, the discussion of whether alternative models may suit varying domains is a topic open for discussion. Figure 2.20 visualises how data requests and feeds are passed between physical systems for a proposed architecture to achieve data ubiquity in the railway domain. At the core of the model is the ontology, which describes the domain. The RaCoOn is used in this case [142]. The ontology - with its asserted triples (ABox) and inferred triples (TBox) facts - is inserted into a physical graph-based triple store. In this case, the program *StarDog* provides the platform for this. Once this has happened, facts from various data sources are parsed to triples and added to the

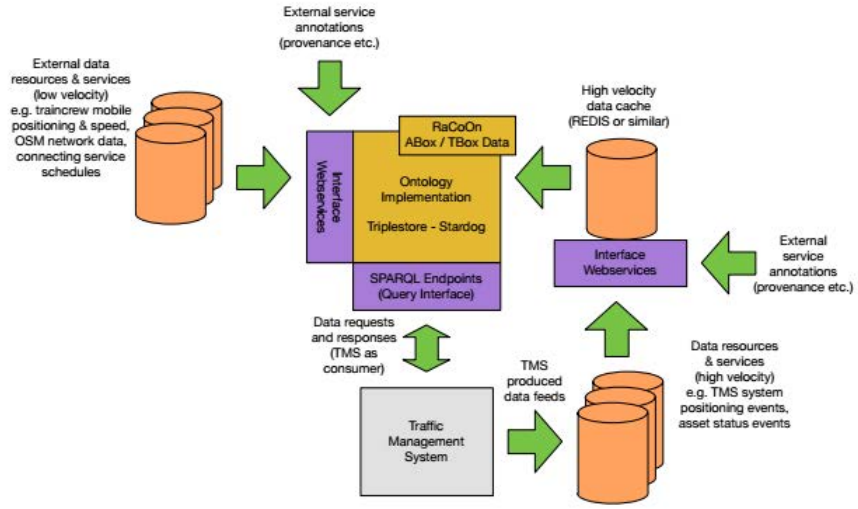


Figure 2.20: Candidate architecture for ontology deployment within railway operations [142].

triple store. StarDog is a bottom-up reasoner and therefore invokes upon asserting each fact into the triple store. As the ontology adds more of these facts, the reasoning engine applies the ontological axioms to the available data, updating it with inferred facts accordingly. The reasoning is particularly useful for a dynamic environment such as the railway domain, as the triple store will hold asserted and inferred facts of a recent view of the domain, allowing for an accurate response to a query at a given time.

The model utilises the concept of *linked data* architecture, enabling the integration of up-to-date external data sources as opposed to static local copies. Furthermore, it reduces the computational intensity of the system. Much of the data provided to the model will be *high-velocity*, meaning the data stream has a high volume of entries. It is best to avoid placing these instances straight into the triple store, as this would require the ontology to reason every time it adds something new. Given the rate of data transfer, this would not be viable. Instead, this data goes into a key-value database which acts as a *cache*. The ontology will only reason upon a specific query and can request this high-velocity data from the database when required.

2.4.2 Data Platforms

Despite the range of models that exist to advance data interoperability in the rail industry, barriers have prevented these models from deployment into operation. One factor is that these models do not address the fundamental problem of data sharing. The rail sector has continued improve in regards to the sharing of data. In 2012 Network Rail released open data including access to a number of operational data feeds, including train positioning and train scheduling data across the network. In addition to this, the Rail Delivery Group (RDG) launched an open data portal in 2015. Both released have led to greater levels of information, and has supported vast improvements in the consistency of information between channels, through the development and maintenance of a National Rail Information database (Darwin).

In 2018, the Department for Transport (DfT) released a document laying out plans for the joint rail data action plan [47]. This plan "[addressed] barriers to make better use of rail data", and focused on five themes: *data transparency*, to clearly and openly categorising rail datasets and defining data sets that are commercially sensitive; *data use and access*, to build knowledge and understanding of the rail data landscape and provide more clarity on access, use and ownership to different datasets; *data standards and quality*, to improve the quality of published datasets and provide indicators of their accuracy, and standardise publishing formats; *data value and principles*, to understand the value and potential of open data and champion the open data agenda and its benefits across all rail organisations; and *rail culture and information*, to encourage more collaboration and partnering with outside businesses and innovators as well as promoting data and information sharing within the industry to improve business efficiencies, performance monitoring and customer experience, and addressing the data and information skills gap through training and apprenticeships.

Since this release, various parties have introduced potential candidates that fit within these themes. Porterbrook, partnered with Elastacloud, offer a platform to "unlock these

silos and enable our customers to drive practical value from data" [105]. The University of Birmingham is developing a centralised data platform with the company Zipabout in an attempt to ease storage and analysis of data from a variety of industry stakeholders. However, this platform currently has no concrete implementation. Siemens offer Railigent, which makes "makes intelligent use of rail data to create added value" [125]. The issue with many of these platforms is that they are competing to solve the same issue, thereby exacerbating the issue of mistrust and lack of sharing. A distributed consensus is required to truly advance the problem, and Chapter 5 intends to investigate how.

2.4.3 Ticketing and the Rail Settlement Plan

The RSP is responsible for the digital services of the rail industry in GB, serving Train Operating Companies (TOCs) and third-party providers with a combination of information and retail services. The company established upon the privatisation of British Rail in the early 1990s, primarily to distribute the ticket fare revenue between copmpeting TOCs. The revenue splits between the vendor and the TOCs that operates along the route. According to anecdotal sources [89], the RSP currently:

- Collects retail sales data from all ticket issuing systems;
- Carries out the correct allocation of ticket revenue to train operators;
- Settles that revenue to the operators;
- Sets standards and accredits all industry ticket issuing systems;
- Maintains the central industry fares database and provides tools for train operators to set fares;
- Distributes fares, timetable, station and other industry data to ticket issuing and information systems;
- Provides the National Reservations Service enabling retailers to book reservations on all trains with reservable seats;

- Provides the capability to pick up pre-ordered tickets at station ticket machines (ticket-on-departure);
- Provides the industry-standard ticket stock.

The privatised nature of the railway industry in GB requires the ticket vendors to record their ticket transactions, and share them on an extensive database known as Latest Earnings Network Nationally Overnight (LENNON), operated by the RSP. LENNON replaced the legacy British Rail system, *Capri*, in April 2002. Operated by the company *Worldline*, LENNON collects and processes all ticket fare data and sends it to the centralised industry *Data Warehouse*, applying allocation factors from the legacy Operational Research Computerised Allocation of Tickets to Services (ORCATS) model and agreements between the TOCs. It controls reservation and revenue sharing on tickets between TOCs. The system divides ticket revenue when a ticket or journey involves trains operated by multiple TOCs.

LENNON calculates rail industry commission to ticket vendors, which varies between 2% - 9% of the sale price. LENNON is a centralised database that relies on a cloud solution provided by Google. Data queries and the addition of data must go through the Application Programming Interface (API) provided by the RDG. This API is static and is subject to modification when the database requires upgrading.

The Office for National Statistics (ONS) provides a series of images to illustrate the process flow between LENNON and other entities; Figure 2.21 illustrates the process map of passenger rail administrative data. Figures A.6 through A.11 in Appendix A display the remaining diagrams.

Passengers using the railways in GB may purchase tickets via two distinct methods: at any station on the network using cash or bank cards, henceforth known as *station purchase*; or via the World Wide Web (WWW), henceforth known as *online purchase*. Station purchase allows passengers to buy from a machine or at the ticket office from a member of staff (subject to whether a station has a ticket office). In contrast, online purchase bypasses the need to use station facilities and allows the passenger to purchase

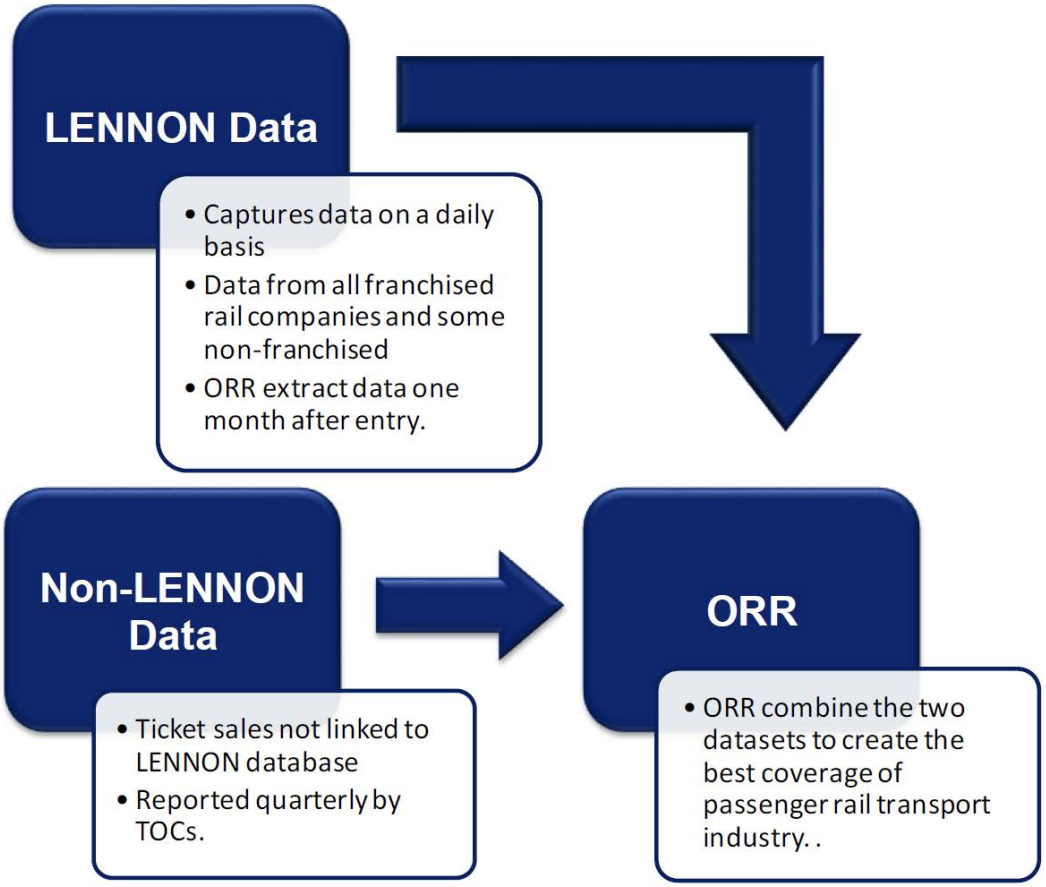


Figure 2.21: Process map showing the collection of passenger rail transport administrative data [114].

a ticket via a browser or a phone application.

The passenger obtains the tickets depending on their method of purchase. Should a passenger opt for station purchase, the ticket machine (or member of staff at the ticket office) prints the ticket at the point-of-sale in the form of a physical paper ticket, or uploads the ticket to a compatible Integrated Transport Smartcard Organisation (ITSO) standard smartcard. Online purchase allows passengers to print tickets at home using a personal printer, via station ticket machine or staff member, or downloaded to a mobile application as an *m-Ticket*.

Passengers validate tickets when entering or leaving stations through electronic gates, or via a conductor while onboard a train. With the former method, it is more common for the barriers to support the paper tickets with magnetic strips, where criteria

of the instance (i.e. the correct date and route) test the validity of the ticket. Some gates offer scanners that allow Quick Responses (QRs) from m-Tickets and home-printed tickets, and also Near Field Communication (NFC) for smartcards. If the passenger has a digital ticket of a kind that the barriers do not accept, interaction with a staff member at the barriers is necessary to gain access.

Though m-tickets are a step in the right direction, this approach to digital ticketing is not without limitations. In order for a passenger to use an m-tickets, it is a prerequisite that a passenger must own a mobile phone, have access to the WWW, and be willing to pay with a digital method of payment. A report conducted by Blank et al. [15] identified that approximately 20% of residents in the United Kingdom (UK) do not have access to the WWW. Therefore, this approach to digital excludes a large group of passengers who may not own mobile phones, have no access to the WWW, or those who seek to pay with cash.

Moreover, passengers who opt to use m-Tickets are subject to specific issues. Should the mobile phone hosting the application become inoperable (due to a flat battery or a hardware issue), the passenger has no way of proving their ticket's validity and will be subject to a fine. Furthermore, the organisations selling tickets restrict digital tickets to their applications. If this application suffers from any downtime, the ticket is no longer accessible, and the passenger is subject to a fine.

Smartcards offer a viable alternative to those excluded from using m-tickets. ITSO standards address how smartcards store and validate tickets, ensuring interoperability across the network. However, this has several restrictions. Each smartcard is only capable of storing five tickets at one time, and a ticket may take up to two hours to deliver to the card [97]. Furthermore, some ITSO smartcards are incapable of storing certain tickets on them, rendering them near useless if a passenger requires a journey between areas using two or more different smartcards. As such, the current approach to digital ticketing is not yet a suitable replacement for legacy ticketing. Because of this, Chapter 6 investigates new solutions to these issues.

2.5 Conclusion

This chapter has provided a literature review and technical background on several topics about the research conducted in this thesis, and discussed the findings. Section 2.2 discussed the mechanics of blockchain technology and the subsequent use-cases and state-of-the-art of existing platforms. From the literature, it is apparent that there is a lack of research into blockchain platforms specific to the railway industry. Moreover, there are several technical limitations of blockchains concerning the storage of data. The fact that blockchains are unsuitable for storing large quantities of data leads to more in-depth investigations into the Web 3.0. The review discusses ontologies as data models to provide data with semantics, and various approaches to distributed data storage using blockchains and other technologies. The majority of literature does not consider security and data transfer as part of their protocols; something that is crucial to Hypertext Transport Protocol (HTTP), and will prove to be a barrier of adoption of Web 3.0. Moreover, there are various flaws to the current decision frameworks that attempt to overcome the barriers of adoption of blockchain technology.

Section 2.3 presented the prerequisite cryptography for blockchain technology and data security, including hash functions, symmetric cryptography, and asymmetric cryptography. This review found that state-of-the-art cryptosystems for both hash functions (the Message-Digest (MD) and Secure Hash Algorithm (SHA) families) and symmetric cryptography (the Advanced Encryption Standard (AES) and Data Encryption Standard (DES)) hold up against proposed attacks offered in the literature, both in the classical and quantum domains. However, this is not the case with asymmetric cryptography, where hypothetical quantum algorithms poise to disrupt the current standards. (This is, of course, dependant on the advancement of quantum computing hardware, which is beyond the scope of this thesis.) This review focused on the classical cryptosystem of Elliptic Curve Cryptography (ECC) and subsequent Key Exchange Scheme (KES) and Digital Signature Scheme (DSS) of Elliptic Curve Diffie Hellman

Key Exchange (ECDH) and the Elliptic Curve Digital Signature Algorithm (ECDSA), respectively. The review also focused on the post-quantum alternative of Supersingular Isogeny Cryptography (SIC), and the implementations of a KES and DSS of Supersingular Isogeny Diffie Hellman Key Exchange (SIDH) and the Supersingular Isogeny Digital Signature Algorithm (SIDSA), respectively. The decision to look at SIC as opposed to other post-quantum cryptosystems was to retain the core mathematics of ECC that are regularly associated with blockchain technology to retain the small key sizes.

Finally, Section 2.4 summarises the field of data in the railway industry, including a discussion of the approach to rail industry data in Great Britain (GB). The majority of research into data in the rail industry focuses on the semantics of the data. Though necessary, the literature fails to address the underlying issue of sharing this data. Semantics are useless if companies do not have access to it in the first place! Furthermore, the review focuses on the Rail Settlement Plan (RSP) that handles the majority of ticketing data. It also becomes apparent that the majority of this technology relies upon legacy systems, to which blockchain might provide a suitable alternative approach without having to reinvent the wheel.

Chapter 3

Problem Statement

This chapter identifies the issues presented in the literature review in Chapter 2, and presents the research intentions of Part II of the thesis. Section 3.1 discusses the disruptive nature of blockchain technology, its suitability as a solution to the data-related issues, and the barriers to adoption. Section 3.2 analyses the issues of data ubiquity and sharing within the Great Britain (GB) rail industry, and the pitfalls of existing solutions. Section 3.3 presents seven research questions based on these findings, and identifies which chapters within Part II answer each specific question.

3.1 A Disruptive Technology or Marketing Buzzword?

There is little doubt that the mechanics of blockchain technology apply to many specific situations. Nonetheless, this has not prevented the technology straying into areas that stretch to mechanics to their limits, and where other technologies might provide a more suitable fit. Because of this, blockchain is seen by some as a magic wand; point and solve the problem. This is, of course, not the case, and careful consideration is mandatory in deciding whether blockchain technology is the most appropriate for a situation.

The active cryptocurrency communities perpetuate this level of distrust, as well as

the economic behaviour of public blockchain platforms. Cryptocurrencies are notorious for their volatile nature, leaving those who have dabbled in the digital market either celebrating newfound riches or commiserating in their rags. The mainstream media jump upon these extreme highs and lows and paint a dangerous and unforgiving picture of blockchain technology.

There are decision models to distinguish between the variety of approaches possible to integrate blockchain technology. However, certain flaws in these models will slow the adoption of the technology and will lead many to regard blockchain technology as unfit for purpose when in fact, it might be well suited. Thus, it is paramount to work towards an environment that aids sensible decisions before committing to large projects with the incorrect toolkit. This thesis will serve towards this issue, by establishing a new decision framework for users to decide which blockchain technology might suit them best.

3.2 The Relationship Between Data and the Rail Industry

In this thesis, we will define the Digital Rail Framework (DRF) of Great Britain (GB) as the cluster of legacy and modern systems discussed in 2.4. These systems are often forced to cooperate with one another, leading to many issues. One significant issue that many new technologies fall by the wayside, as it is a difficult task to integrate them with the legacy systems. As such, there is a pressing need to update these legacy systems to modern state-of-the-art systems. However, one aspect of the railways in GB prevents a coherent process towards developing new systems: privatisation.

This shift to a privatised industry has resulted in data siloation and a lack of trust between stakeholders, to the point that the sharing of crucial data is a painstaking process. The many attempts at solving this problem offered by The University of Birmingham, Porterbrook, Siemens, Network Rail, the Rail Delivery Group (RDG),

and the Department for Transport (DfT) (discussed in Section 2.4.2) fail to address the fundamental issue of sharing the data and form a paradox; if every company offers their own solution to data sharing to combat siloation, then solution of these new platforms occur, and the industry falls back to the original problem!

The challenge facing the industry is finding a suitable platform that allows for easy storage and smooth transfer of data while securing the data using appropriate cryptographic protocols. There are two elements to consider; the enterprise, and the technological. On the enterprise side, there are stakeholder relationships and the business logic to consider, not to mention financial implications. On the technical side, is how to weave these complex abstractions into a concrete system that works reasonably for each participant. Fundamentally, any shift to new technologies should ascend beyond politics and focus purely on technological advantages. This thesis will take the data issues of the rail industry (namely data sharing for industrial data and ticketing) and investigate the potential for blockchain technology to provide a brand new approach to these old problems.

3.3 Research Questions

Drawing from the literature review and the discussions of Sections 3.1 and 3.2, this section proposes five technical research questions as follows:

1. Provided that a significant barrier to adoption of blockchain technology is the misinterpretation of when to appropriately deploy it, and that the current models in the literature have certain flaws, how can a new model become more robust and provide the users with more in-depth feedback?

2. Given that data siloation and centralisation is a common occurrence in the rail industry, can blockchain technology overcome the underlying issues of data centralisation and the lack of trust between stakeholders?

3. Provided that blockchains themselves are not suitable for native storage of large quantities of data, and that current literature suggests secure centralised storage or insecure distributed storage as a solution, how can the auditing and distributed properties of blockchain technology integrate with a secure distributed storage solution?

4. If such a platform exists as an answer to Question 2, which post-quantum cryptosystem is a suitable replacement for the classical cryptosystems commonly used in blockchain technology?

5. Provided that the current data-driven approach to ticketing is fragmented and has numerous flaws, can the distributed mechanics of blockchain technology provide a solution?

6. How can such a system help in overhauling the Rail Settlement Plan (RSP)?

7. Assuming answers for all previous questions, will blockchain technology disrupt the rail industry as it has with other industries?

Table 3.1 illustrates the chapters where this thesis investigates and answers these questions.

Table 3.1: The locations of where this thesis answers the research questions.

Question	Investigated In
1	Chapter 4
	Section 5.2.1
	Section 6.2.1
2	Chapter 5
3	Chapter 5
4	Chapter 5
5	Chapter 6
6	Chapter 6
7	Chapter 4
	Chapter 5
	Chapter 6

Part II

Novel Research

Chapter 4

An Improved Model for Identifying Blockchain Suitability

This chapter introduces and explains the Model of Blockchain Suitability (MoBS). The MoBS is a new model designed to aid decisions as to whether users require blockchain technology for their use cases. The model prompts the user with a property, who then selects a relevant statement from a pre-determined list of statements for said property. Once the user has answered to each property, the MoBS provides them with quantitative, human-readable feedback (a percentage of the suitability of various technology suggestions). It then suggests specific blockchain platforms according to this feedback. Section 4.1 discusses previous models of the literature; both the advantages and disadvantages and the subsequent need to extend and adapt them. Section 4.2 explains the methodology of the development and the mechanics of the MoBS. Section 4.3 validates the MoBS by providing results about the performance, the scope, and the outcomes for specific use cases for the MoBS, and cross-checks these results against the previous models. Section 4.4 discusses the advantages and disadvantages of the MoBS, and considers the impact of the research within the field and industry. Section 4.5 concludes the findings of the research.

4.1 Introduction

As evaluated in Section 2.2.2, many industries are beginning to adopt blockchain technology. Though there exist plenty of use cases to which blockchains are highly suited, there is anecdotal and referential evidence that suggests blockchain technology is merely something users adopt for the sake of embracing new technology [83]. This inappropriate adoption is likely due to misunderstanding of what blockchain technology truly is, and its suitability (and unsuitability) to various situations.

To avoid these situations of misunderstanding and haste in adopting the technology into inappropriate scenarios, and to bridge the gap between understanding the technology and adoption, various models exist to aid the decision process. Section 2.2.2 describes three existing models to aid the decision process of choosing whether to implement blockchain technology. The first framework to appear in the literature, by Lo et al. [84], offers the user a flow chart with seven decision nodes. By traversing the flow chart, the user arrives at a simple binary conclusion of whether they should or should not use a blockchain. Wust and Gervais [150] stay with the flow chart model and develop upon this idea by expanding the number of final suggestions, introducing a distinction between private and public blockchain platforms. Both Lo et al. [84] and Wust and Gervais [150] self-validate their models according to known use cases and the knowledge of the authors. The model of Maull et al. [86] uses a similar flow chart based approach. With nine decision nodes, a range of suggestions that align best with the classifications discussed in Section 2.2.2, and development with real blockchain developers, this currently model stands as the most in-depth method of reaching a decision.

As indicated in Section 2.2.2, flow chart models offer a rigid approach. Because of this (potentially to the disadvantage of users), the Model of Blockchain Suitability (MoBS) aims to depart from the binary answer convention and develop a new framework that delivers a percentage score metric for each possible category (public blockchain, private

blockchain etc.), and allows the user to answer all questions before the model reaches a conclusion. These additions will overcome the flaws of flow chart approaches. We summarise the aims of the MoBS by the following five goals:

1. Ask users questions covering all aspects of a platform, including the nature of the users and the data, and the desired performance.

2. Retain the intuitive approaches from previous models, with a straightforward process with simple questions and answers.

3. Provide users with new quantitative metrics to direct them to the most suitable technology, with easily presentable feedback.

4. Provide users with qualitative feedback to suggest real platforms suitable for them within the suggested category.

5. Ensure the model is flexible in developing and growing along with the field of blockchain technology.

This work is presented in order to answer Question 1 of the research questions proposed in Section 3.3: "Is it possible to expand upon previous models of identifying blockchain suitability for a use case in order to overcome the flaws with flow chart approaches?"

4.2 Methodology

The Model of Blockchain Suitability (MoBS) divides into two main components: the *Core*, responsible for storing the fundamental parameters and functionality of the model; and the *Interactive Process*, the question-and-answer process the user undergoes to obtain feedback. The design means that users never see the Core, and can focus on responding during the Interactive Process. The MoBS requires a software implementation to enable the relationship between the Core and the Interactive Process; Figure 4.1 illustrates this relationship. The Core is further subdivided into three layers: the parameters (*Layer 1*);

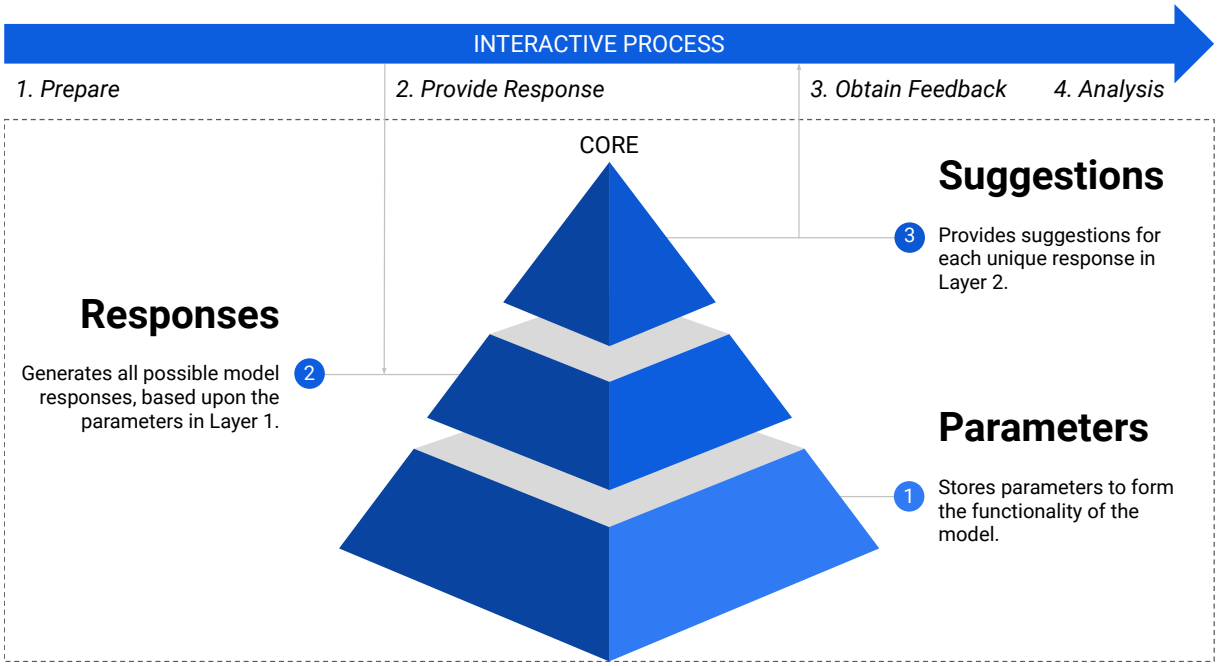


Figure 4.1: The hierarchical layers of the Core of the MoBS and the Interactive Process. Note that higher layers are dependent on lower layers.

the responses (*Layer 2*); and the suggestions (*Layer 3*). These layers form a hierarchy of dependency, whereby each successive layer depends on the state of its predecessor. The Interactive Process subdivides into four distinct phases: *Prepare*; where the user prepares the MoBS software; *Provide Response*, where the user selects the most apt statements to provide Layer 2 of the Core with required data; *Obtain Feedback*, where the user obtains the feedback calculated by Layer 3 of the Core; and *Analysis*, where the user analyses the feedback provided.

The following sections provide an in-depth description of the three layers of the Core in full, along with their relationship to the Interactive Process. Section 4.2.1 describes Layer 1, which stores the framework parameters, and its relationship with the Prepare phase. Section 4.2.2 describes Layer 2, which derives two quantitative score arrays (an Individual Score Array (ISA) and Domain Score Array (DSA), the purpose of which is explained in due course) from results from the Provide Responses phase. Section 4.2.3 describes Layer 3, which determines the outcome of suggestions based upon the responses of Layer 2, and supplies these back to the user via the Obtain

Feedback phase. The construction of the model (in comparison to the user experience) is complex; Section 4.3.2 will demonstrate an example runthrough once the construction is understood.

4.2.1 The Prepare Phase and Layer 1

Layer 1 stores the internal parameters that determine the functionality of the MoBS Core. Layer 1 is responsible for storing the statements that users will select during the Provide Responses phase of the Interactive Process. During the Prepare phase of the Interactive Process, Layer 1 provides these statements to the user.

The underlying structure of Layer 1 is mathematically represented to enable ease of development of the software; Figure 4.2 supplements these representations as an abstracted illustration. Layer 1 L divides into a number $0 < j < i$ of distinct properties

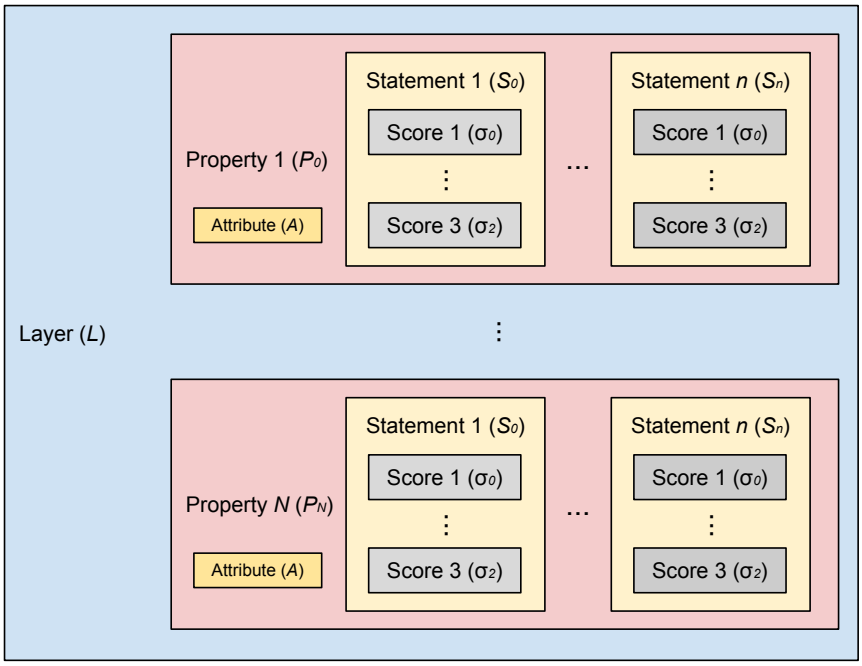


Figure 4.2: Abstracted illustration of Layer 1 of the Core. There are four major concepts: the layer (blue); properties (red); statements (yellow); and scores (grey).

P_j, whereby

$$L = [P_0, \ldots, P_i]. \tag{4.1}$$

Each property has an associated attribute tag, and a number j of associated statements S such that

$$P_j = [S_0^j, \ldots, S_{k_j}^j], \tag{4.2}$$

where K_j is the last element. Each statement has three associated category scores σ, such that

$$S_k^j = [\sigma_0^{j,k}, \sigma_1^{j,k}, \sigma_2^{j,k}], \tag{4.3}$$

where σ_0 is the score of suitability for a non-blockchain technologies, σ_1 is the score of suitability for a private blockchain, and σ_2 is the score of suitability for a public blockchain. The score σ may take three possible values such that

$$\sigma \in \{-1, 0, 1\}. \tag{4.4}$$

Table 4.1 explains the meaning of the scores presented in Equation 4.4, and demonstrates the logic behind the selection of just three values of $\{-1, 0, 0\}$. A wider range of values was considered but decided against, as Table 4.1 demonstrates the adequacy of having three.

Table 4.1: A summary of the scores each category may take. Henceforth, the positive score is highlighted in blue, and the negative score is highlighted in red.

Score	Meaning
1	It will work, and will work well.
0	It will work, but will not be beneficial.
-1	It will not work.

The mathematical definition of Layer 1 provides the flexibility to modify the MoBS to various numbers of properties and statements. This is to allow easy updates to the model as the field of blockchain technology expands. Nonetheless, the MoBS is redundant without a specific *implementation*. The implementation of the MoBS Layer 1

used in this thesis has three attribute tags, and ten properties. Table 4.2 introduces the attribute and property nomenclature, along with a simple identity naming convention.

Table 4.2: The attributes and properties of the MoBS Layer 1 implementation used in this thesis. The ordering within the table is retained, thus the identity convention is straightforward; for example, *A2P3* means the third property (P3) of the second attribute (A2).

Attribute		Property		
ID	Name	ID	Name	Total Statements
A1	Platform	A1P1	Participants	4
		A1P2	Writers	3
		A1P3	Trust	4
		A1P4	Consensus	4
A2	Data	A2P1	State	2
		A2P2	Transparency	3
		A2P3	Integrity	2
A3	Performance	A3P1	Latency	3
		A3P2	Scale	3
		A3P3	Redundancy	2

A1: Platform

The *Platform* attribute tag is for properties with statements about the network architecture and participant responsibilities. There are four properties: *Participants*, to determine the interaction between the platform and the participants; *Writers*, to determine the write permissions of the platform; *Trust*, to determine governance responsibilities; and *Consensus*, to determine the consensus of writing to the platform.

A1P1: Participants The Participants property determines whether the system has either a single participant or multiple participants. Participation refers to interacting with the system in any way; whether by reading or writing to the system state. Table 4.3 provides the four statements and the associated suitability scores for this property.

These statements cover all angles, providing a variety of options based around the read and write privileges. (The MoBS infers that all writers have read privileges.)

Table 4.3: The statements for the Participants property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A1P1S1	The system will have multiple participants. All participants should have read and write privileges.	0	0	1
A1P1S2	The system will have multiple participants. All participants should have read privileges, but only some participants will have write privileges.	0	1	-1
A1P1S3	The system will have multiple participants. All participants should have read privileges, but only one participant will have write privileges.	1	0	-1
A1P1S4	The system will have one single participant.	1	0	-1

Statement S1[1] provides an option for systems that will have multiple participants and that are seeking to have read and write privileges for every user on the platform. A public blockchain is most suited in this case, as the technology has the permissions built-in; nonetheless, this is still achievable via a private blockchain or non-blockchain technologies but requires extra overhead to regulate the restrictions. Statements S2 and S3 restrict the write privileges. In these situations, public blockchains are redundant, as it is impossible to impose such restrictions. Statement S4 provides an option for systems that will have a solitary participant; in this case, although a private blockchain could work, non-blockchain technologies (such as databases) are the better option as they require less overhead regarding set-up and maintenance.

A1P2: Writers The Writers property determines which participants with write access to the system (i.e. the writers) are acknowledged and known by the trust anchor before writing, and trusted by the trust anchor[2] to aid administrative processes. Table 4.4 provides the statements and the suitability scores relevant to this property.

[1]For ease of reference, statements are henceforth referred to by a truncated identity, as the tables infer the property and attribute.

[2]The statements refer to a "trust anchor" of governance. A trust anchor is the source of authority concerning the running and modification of the platform and its rules.

Table 4.4: The statements for the Writers property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A1P2S1	Not all writers are known prior to writing.	-1	-1	1
A1P2S2	All writers are known. However, only some writers are trusted.	-1	1	0
A1P2S3	All writers are known. Furthermore, all writers are trusted prior to writing.	1	0	0

Statements S1, S2, and S3 represent the three possibilities that a platform has with regards to the participants who are writers. Either the system knows and trusts all writers before writing (S3), or the system knows all writers but does not trust all writers before writing (S2), or the system does not know all writers before writing (S1). The difference between "known" and "trusted" is subtle. "Known" refers to writers who are explicitly described by their platform credentials and are neither pseudonymous nor anonymous. "Trusted" refers to the level of trust between participants, and whether each participant trusts the others to behave in a non-adversarial and non-tampering manner.

A1P3: Trust The Trust property determines with whom the authority of governance and policy lies. Governance and policy refer to administrative rights over maintaining the software, and deciding upon which participants have which rights. Table 4.5 provides the statements and the suitability scores relevant to this property.

Statement S1 offers the option of a distributed trust anchor, such that every participant has a right to influence the governance. Statements S2 and S3 place this trust in the hands of a specific group of participants or a third party, respectively. Statement S4 covers the scenario that an individual participant maintains governance.

Table 4.5: The statements for the Trust property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A1P3S1	There should be no trust anchor. Governance should be the responsibility of all participants.	-1	-1	1
A1P3S2	There should be a restrictive trust anchor. Only only certain participants entitled to modify governance.	-1	1	-1
A1P3S3	There should be a third party-trust anchor to modify governance.	1	-1	-1
A1P3S4	There should be a single-participant trust anchor to modify governance.	1	-1	-1

A1P4: Consensus The Consensus property determines the method of consensus for transactions on the system. Table 4.6 provides the statements and the suitability scores relevant to this property.

Table 4.6: The statements for the Consensus property, and the relevant suitability scores for non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A1P4S1	Consensus should be provided by any system participant.	0	0	1
A1P4S2	Consensus should be restricted to known participants.	0	1	-1
A1P4S3	Consensus should be provided by a trusted third party.	1	-1	-1
A1P4S4	Consensus is not required.	1	-1	-1

Statements S1, S2, and S3 provide various options for platforms that require consensus for adding data in a similar format to that of the Trust property. Statement S1 allows consensus by any participant, Statement S2 allows consensus by a pre-selected group of participants, and Statement S3 hands the consensus to a trusted third party. Statement S4 covers the scenario where the consensus of the change of the platform state is not required; for this approach, only non-blockchain technologies are suitable.

A2: Data

The *Data* attribute tags the properties that pertain to the fundamental data within the platform. There are three properties: *State*, to determine the necessity of real-time distribution and updating of the platform state; *Transparency*, to determine whether the platform data must be confidential in any way; and *Integrity*, to determine whether a change in the state requires a ledger to trace the transactions of data.

A2P1: State The State property determines whether the state of the system needs to be consistent amongst all participants. Table 4.7 provides the statements and the suitability scores relevant to this property.

Table 4.7: The statements for the State property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A2P1S1	The state of the system requires real-time consistency amongst all participants.	-1	1	1
A2P1S2	The state of the system does not require real-time consistency amongst all participants.	1	-1	-1

There is a binary choice of statement for the State property. Statement S1 determines that the platform must enforce real-time consistency of the state of the data, whereby Statement S2 determines that the distribution of state is not required. Blockchains suit the former, while non-blockchain technologies suit the latter.

A2P2: Transparency The Transparency property determines which system data requires transparency, and which data requires confidentiality. Transparency means that each participant with read access to the system (i.e. the readers) may read the data. Confidentiality restricts read access to a certain audience via cryptographic protocols. Table 4.8 provides the statements and the suitability scores relevant to this property.

Table 4.8: The statements for the Transparency property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A2P2S1	All data on the system must remain transparent.	1	1	1
A2P2S2	System data requires a mixture of transparency and confidentiality.	1	1	-1
A2P2S3	All data on the system must remain confidential.	1	-1	-1

The statements for the Transparency property offer the options of whether: all data requires confidentiality; all data requires transparency; or a hybrid of confidentiality and transparency. Statements S3, S1, and S2 offer these, respectively. Public blockchains suit transparent data, as this is part of the design philosophy. However, they become redundant when data required confidentiality, as it is impossible to impose this without restrictions; a private blockchain may overcome these. Non-blockchain technologies will work in all scenarios.

A2P3: Integrity The Integrity property determines whether the provenance of data is necessary within the system. In this situation, provenance refers to the full history of data creation and transfer, supported by transaction history. Table 4.9 provides the statements and the suitability scores relevant to this property.

Table 4.9: The statements for the Integrity property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A2P3S1	Transaction history is required to create provenance of data ownership.	-1	1	1
A2P3S2	Transaction history is not necessary.	1	-1	-1

There is a binary choice of statement for the Integrity property, between Statement 1

that determines the system requires transaction history and Statement 2 that determines the system does not require transaction history. Blockchains suit the former, while non-blockchain technologies suit the latter.

A3: Performance

The *Performance* attribute tags the properties that pertain to the desired performance of the platform. There are three properties: *Latency*, to determine the throughput of transactions required; *Scale*, to determine the quantity of data the platform must handle; and *Redundancy*, to determine whether the platform needs to avoid duplication of data.

A3P1: Latency The Latency property determines the required number of transactions per second. Here, the model defines a transaction as any movement of data on the system. Table 4.10 provides the statements and the suitability scores relevant to this property.

Table 4.10: The statements for the Latency property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A3P1S1	The system will perform with a low low latency of under 100tps.	1	1	1
A3P1S2	The system will perform with medium latency between 1000 tps and 20000 tps.	1	1	-1
A3P1S3	The system requires a high latency of over 20000 tps.	1	-1	-1

The Latency statements categorise the desired number of transactions per second (tps). Statement 1 determines this number is below 100 tps, Statement 2 determines this number is between 100 tps and 20,000 tps, and Statement 3 determines this number is above 20,000 tps. The property derives these numbers from commonly associated latencies in the literature; O'Neal [102] provides a list of common public platforms

that have tps rates below an average number of 100 tps, whereas Gorenflo et al. [59] has proved that Hyperledger Fabric, a private blockchain, is scalable to 20,000 tps. Non-blockchain technologies such as databases are naturally suited to high latencies.

A3P2: Scale The Scale property determines the amount of data the system must handle. Handling data refers to the hosting and retrieval of data. Table 4.11 provides the statements and the suitability scores relevant to this property. The Scale statements

Table 4.11: The statements for the Scale property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A3P2S1	The system will only need to handle a small amount of data.	1	1	1
A3P2S2	The system will need to handle a medium amount of data.	1	1	0
A3P2S3	The system will need to handle a large amount of data.	1	0	-1

are similar to those of the Latency property, whereby the statements pertain to definitive quantities; in this case, the amount of data the platform must handle. We derive these values from common block sizes of popular platforms.

A3P3: Redundancy The Redundancy property determines whether the system needs to maintain the data and prevent duplication. Table 4.12 provides the statements and the suitability scores relevant to this property.

Table 4.12: The statements for the Redundancy property, and the relevant suitability scores for a non-blockchain technologies σ_0, a private blockchain σ_1, and a public blockchain σ_2.

Statement		Scores		
ID	Description	σ_0	σ_1	σ_2
A3P3S1	The system must avoid data redundancy.	-1	1	1
A3P3S2	The system has no need to avoid data redundancy.	1	-1	-1

There is a binary choice of statement for the Redundancy property, between Statement 1 that determines the platform must avoid duplication of data, and Statement 2 that determines duplication is not an issue. Blockchains suit the former, while non-blockchain technologies suit the latter.

4.2.2 The Provide Response Phase and Layer 2

Layer 2 utilises the parameters from Layer 1 and the user response from the Provide Response phase to calculate two score arrays; an ISA ι, and a DSA δ. Each score array consists of three elements; a normalised value indicating the suitability for each of a non-blockchain technology, a private blockchain, and a public blockchain. Henceforth, these are referred to as *categories*. Each element has a value between 0 and 1, where 0 means most unsuitable, and a value of 1 means most suitable. (These values convert to percentages by multiplying them by 100.)

In order to generate these score arrays for the use case, a user responds to Layer 1 L via the Provide Response phase of the Interactive Process by carrying out the following instructions:

- The user traverses each property P in L;
- The user selects one statement S from each P, such that the S is the most apt for the use case.

Given i properties, the complete set of i selected statements forms a model response r, such that[3]

$$r = [S_0, \ldots, S_i], S_n \in P_n. \tag{4.5}$$

Given r, the layer calculates a total score array T for the response as

$$T = \left[\sum_{S \in r} S_{[0]}, \sum_{S \in r} S_{[1]}, \sum_{S \in r} S_{[2]} \right]. \tag{4.6}$$

[3]For a clarification of the notation used in this section, please refer to Appendix B.1.

Individual Score Array

Once Layer 2 has determined T, it calculates the ISA ι. The ISA exists to demonstrate to the user which of the three categories (public blockchain, private blockchain, or non-blockchain) is best suited to this response, and this response alone. This score array requires an adjustment to T to replace any negative numbers with zero. The adjusted score array A is

$$A = [T^{+}_{[0]}, T^{+}_{[1]}, T^{+}_{[2]}], \tag{4.7}$$

whereby the layer replaces any negative values with zero, and then calculates an absolute total a as

$$a = A_{[0]} + A_{[1]} + A_{[2]} \tag{4.8}$$

which aids the normalisation of A to provide ι as an array of percentages, given as

$$\iota = \left[\frac{A_{[0]}}{a}, \frac{A_{[1]}}{a}, \frac{A_{[2]}}{a} \right]. \tag{4.9}$$

A verbose description of the above calculation is that it ensures that Layer 2 converts any negative category scores to zero, and normalises the scores. It converts negative values, as any negative value indicates the category is unsuitable, and the MoBS is attempting to grade suitability alone.

Domain Score Array

The DSA is a score for each category relative to all possible scores for this category in the MoBS. The DSA exists to compare the suitability of the three categories against other responses, to provide the 'whole picture'. To achieve this, Layer 2 determines the scores for each possible response to the MoBS, such that

$$R = [r_0, \ldots, r_k], \tag{4.10}$$

where k is the total number of possible responses. A total t for all adjusted score array values A for each response r is calculated such that

$$t = \left[\sum_{r \in R} A_{[0]}, \sum_{r \in R} A_{[1]}, \sum_{r \in R} A_{[2]} \right]. \tag{4.11}$$

These scores are then normalised against each other category score possible, to provide an array of scores δ for each category relative to the entire domain, such that

$$\delta = \left[\frac{A_{[0]}}{t_{[0]}}, \frac{A_{[1]}}{t_{[1]}}, \frac{A_{[2]}}{t_{[2]}} \right]. \tag{4.12}$$

4.2.3 The Obtain Feedback Phase and Layer 3

Once Layer 2 calculates the ISA and DSA, it provides these values back to the user via the Obtain Feedback phase of the Interactive Process. Each score array provides a normalised value indicating the suitability for each category; non-blockchain, private blockchain, and public blockchain. Multiplying each element by 100 achieves a percentage, which is a more human-readable value. Layer 2 normalises these values so that the total of all three elements equates to 100%. Layer 3 then infers the final suggestion by utilising the principle of majority, such that any category with a score x above 66.6% is more suitable than the other two categories combined. Table 4.13 provides a detailed description of the possible levels x might take, alongside a description of what this score indicates.

Platform Suggestions

The user provides their field of interest for their response. The MoBS correlates this with the platform category to determine whether any real platform suggestions match. The following section presents these platforms under the five use cases identified in Section 2.2.2.

Table 4.13: The suitability level values and their respective indications for a score x.

Level	Score	Indication
Low	$0\% < x \leq 33.3\%$	Low level of suitability.
Medium	$33.3\% < x \leq 66.6\%$	Medium level of suitability. Consider using with other medium categories, or forming hybrid approach with additional technologies.
High	$66.6\% < x \leq 100\%$	High level of suitability. Strongly consider using over the other categories.

Business and Industry Table 4.14 presents the potential platforms for business and industry use cases, with a particular insight into energy and sustainability, education, healthcare, supply chains, and transportation.

Data Security and Management Table 4.15 presents the potential platforms for data security and management use cases, with a particular insight into data distribution, and secure storage.

Governance Table 4.16 presents the potential platforms for governance use cases, with a particular insight into eVoting, and Decentralized Autonomous Organizations (DAOs).

Integrity Verification Table 4.17 presents the potential platforms for integrity verification use cases, with a particular insight into intellectual property, and insurance.

Internet of Things Table 4.18 presents the potential platforms for Internet of Things (IoT) use cases, with a particular insight into device management, and data capture and distribution.

Table 4.14: The potential platforms for business and industry use cases, with a particular insight into energy and sustainability, supply chains, and transportation.

	Platform	
Application	Private Blockchain	Public Blockchain
Currency	Hyperledger	Bitcoin Monero Ripple
Education	Hyperledger	Blockcerts APPII Gilgamesh
Energy and Sustainability	Hyperledger	Power Ledger Grid+ Ondiflo Transactive Grid
Healthcare	Hyperledger	MedicalChain MedRec Patientory Solve.Care
Prediction Marketplace	Hyperledger	Augur Bitshares Viacoin
Supply Chains	Hyperledger	VeChain Ambrosus Waltonchain
Transportation	Hyperledger	BiTA Standards

4.3 Results

This section provides results about various aspects of the Model of Blockchain Suitability (MoBS) implemented in this thesis. Section 4.3.1 looks at the scope of the model, i.e. the number of possible unique responses to the model and the distribution of all possible scores based upon Table 4.13. Section 4.3.2 selects some particular use cases and tests them against the MoBS, with a particular insight into the process and the results. Finally, Section 4.3.3 looks at the performance of the model as an application.

Table 4.15: The potential platforms for data security and management use cases, with a particular insight into data distribution, and secure storage.

	Category	
Application	Private Blockchain	Public Blockchain
Data Distribution	Hyperledger	Ethereum Filecoin
Secure Storage	Hyperledger Off-Chain Storage	Filecoin Off-Chain Storage

Table 4.16: The potential platforms for governance use cases, with a particular insight into eVoting, and DAOs.

	Category	
Application	Private Blockchain	Public Blockchain
eVoting	Hyperledger	Ethereum BEMPAS
DAOs	Hyperledger	Ethereum

4.3.1 Model Scope

The total number of unique responses N for the MoBS is

$$N = \prod_{P \in L} j, \tag{4.13}$$

where $P \in L$ is each property P of Layer 1 L, and j is the number of statements for the property. For the implementation of the MoBS in this thesis and with reference to Table 4.2, this results in $41,472$ unique responses. Tables 4.19 and 4.20 provide the rate of appearance of suggestion level (from the suggestion levels in Table 4.13) for each category and all $41,472$ possible Individual Score Arrays (ISAs) and Domain Score Arrays (DSAs).

By analysing the suitability levels of Table 4.13, we conclude that there are ten possible outcomes per response. Table 4.21 provides these outcomes and the frequency of these outcomes as a percentage across all unique responses. Three of these outcomes are *majority* outcomes, whereby one of the categories has a score greater than 66.6% and

Table 4.17: The potential platforms for integrity verification use cases, with a particular insight into intellectual property, and insurance.

	Category	
Application	Private Blockchain	Public Blockchain
Intellectual Property	Hyperledger	APPII Ethereum DEIP IP Ledger
Insurance	Hyperledger	Black B3i ChainThat Immediate Lemonade RiskBazaar Teambrella Tieron

Table 4.18: The potential platforms for IoT use cases, with a particular insight into device management, and data capture and distribution.

	Category	
Application	Private Blockchain	Public Blockchain
Device Management	Hyperledger	FetchAI Power Ledger Waltonchain IOTA
Data Capture	Hyperledger	FetchAI

receives a high level of suitability. Another six of these outcomes are *minority* outcomes, whereby one or two of the categories have values higher than 33.3% and receive a medium level of suitability. There are rare cases where each of the category scores is 33.3%, thus each receiving low suitability scores, and neither a majority nor a minority.

4.3.2 Use Case Evaluation and Validation

As an example, assume a hypothetical user by the name of Ernest, who wishes to identify whether blockchain technology is appropriate for his use case. Before Ernest's

Table 4.19: The distribution of scores for every possible ISA in the MoBS.

Platform	Suggestion	Score (%)
Non-Blockchain Technologies	Low	18.6
	Medium	26.3
	High	55.1
Private Blockchain	Low	61.5
	Medium	29.3
	High	9.2
Public Blockchain	Low	89.8
	Medium	9.3
	High	0.9

Table 4.20: The distribution of scores for every possible DSA in the MoBS.

Platform	Suggestion	Score (%)
Non-Blockchain Technologies	Low	47.9
	Medium	40.6
	High	11.5
Private Blockchain	Low	83.7
	Medium	14.7
	High	1.6
Public Blockchain	Low	96.8
	Medium	3.0
	High	0.2

interaction, the Core of the model predetermines every possible response to the model. This is so that once Ernest provides his response, the model can efficiently retrieve this response without performing any new calculations.

Ernest proceeds via the Interactive Process by selecting the aptest statement from each property. Once completed, Ernest has a set of ten statements; in this example, each statement is the first statement from each property. Table 4.22 visualises the process within the model to achieve the ISA and the DSA for the response that matches Ernest's. Once Ernest provides his response, the model retrieves the predetermined response and associated ISA and DSA.

In order to check the validity of this implementation, the MoBS tests against three real use cases; currency, supply chains, and file systems. Assume two users of the MoBS

Table 4.21: An alternative distribution of the results of each ISA in the MoBS, with a focus on the suggested platforms.

Variety	Suggested Platforms	Score (%)
Majority	Non-Blockchain Technologies	55.1
	Private Blockchain	9.2
	Public Blockchain	0.9
		65.2
Minority	Non-Blockchain Technologies	6.5
	Private Blockchain	6.1
	Public Blockchain	1.9
	Non-Blockchain Technologies and Private Blockchain	14.5
	Non-Blockchain Technologies and Public Blockchain	1.6
	Private Blockchain and Public Blockchain	3.6
		34.2
None	No Platform Suggested	0.6

for each use case. For ease, Frank is the first user, and Susan is the second user. Each user takes a slightly different approach to their use case, in order to demonstrate the MoBS is not a one-size-fits-all model. The tables provided henceforth are a shortened version of that of the example in Table 4.22; Appendix B.2 provides supplementary extended tables. Each result is cross-checked with the previous models of the literature to ensure that the feedback is closely aligned to the results. In each case, the results matched with the outputs of said models, demonstrating the validity of the MoBS.

Currency

Frank has the intention of creating a fully decentralised, high-latency currency platform. Table 4.23 indicates Frank's responses to the MoBS, which strongly suggests that Frank uses a preexisting or bespoke public blockchain, as both the ISA and DSA values for the public blockchain are above the value of 66.6%.

Meanwhile, Susan seeks to create a government-backed digital currency. Table 4.24 indicates Susan's responses, which strongly suggest she uses a private blockchain, as both the ISA and DSA values for the private blockchain are above the value of 66.6%.

Table 4.22: An example response to MoBS. The simplified string identity is "1111111111".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S1	0	0	0	0	1	1
A1P2S1	-1	-1	-1	-1	1	2
A1P3S1	-1	-2	-1	-2	1	3
A1P4S1	0	-2	0	-2	1	4
A2P1S1	-1	-3	1	-1	1	5
A2P2S1	-1	-4	1	0	1	6
A2P3S1	-1	-5	1	1	1	7
A3P1S1	1	-4	1	2	1	8
A3P2S1	1	-3	1	3	1	9
A3P3S1	-1	-4	1	4	1	10
Total Score	-4		4		10	
Adjusted Score	0		4		10	
ISA (%)	0		29		71	
DSA (%)	0		40		100	

Table 4.23: Frank's response to the MoBS for his approach to currency. The simplified string identity is "1111111321".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	-2	2	7
Adjusted Score	0	2	7
ISA (%)	0	22	78
DSA (%)	0	20	70

Supply Chains

Regarding supply chains, Frank opts for a system where the stakeholders of the supply chain provide consensus and trust. Table 4.25 indicates the ISA and DSA for Frank's approach. The results strongly recommend the use of a private blockchain.

Susan takes a similar approach to the platform but moves the consensus and trust to a trusted third party, in-line with traditional approaches to supply chains. Table 4.26 indicates the ISA and DSA for Susan's approach. Despite a slight increase to the ISA and DSA scores for the non-blockchain technologies, the MoBS still strongly recommends a

Table 4.24: Susan's response to the MoBS for her approach to currency. The simplified string identity is "2322121321".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	0	7	-2
Adjusted Score	0	7	0
ISA (%)	0	100	0
DSA (%)	0	70	0

Table 4.25: Frank's response to the MoBS for his approach to supply chains. The simplified string identity is "2222121221".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	-2	10	-2
Adjusted Score	0	10	0
ISA (%)	0	100	0
DSA (%)	0	100	0

private blockchain.

Table 4.26: Susan's response to the MoBS for her approach to supply chains. The simplified string identity is "2233121221".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	1	6	-2
Adjusted Score	1	6	0
ISA (%)	14	86	0
DSA (%)	10	60	0

File System

Frank wishes to host a local file system for his company in order to store and share data. Table 4.27 indicates the ISA and DSA for this approach. These results indicate that non-blockchain technologies (and most likely a database) is the only suitable approach to Frank's use case. On the other hand, Susan requires a file system to handle to storage and manipulation of big data. Table 4.28 indicates the ISA and DSA for this approach. These results indicate the same suggestion of non-blockchain technologies, though with a lower DSA, suggesting that non-blockchain technologies might require new

Table 4.27: Frank's response to the MoBS for his approach to data storage. The simplified string identity is "4344222212".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	10	-2	-7
Adjusted Score	10	0	0
ISA (%)	100	0	0
DSA (%)	100	0	0

Table 4.28: Susan's response to the MoBS for her approach to data storage. The simplified string identity is "2322222332".

Statement ID	Non-Blockchain	Private Blockchain	Public Blockchain
Total Score	6	0	-9
Adjusted Score	6	0	0
ISA (%)	100	0	0
DSA (%)	60	0	0

technologies to bolster Susan's use case.

4.3.3 Software Performance

The MoBS software imports the parameters into the Core via a Comma-Separated Values (CSV) file. Once imported, the Core predetermines every possible response. This is in order to determine the DSAs. However, it is also advantageous in regards to the retrieval time during the Interactive Process, as the calculation of the ISA and DSA need not recompute every time. Table 4.29 provides the performance of the importing of the parameters, and the generation of the responses; an Intel Core i7 6700k processor with a clock speed of 4 GHz runs these tests.

Table 4.29: Performance results for the MoBS.

Part	Elapsed Time (ms)	Clock Cycles
File Import	16.60	66,400,000
Response Generation	438.16	1,752,640,000
Total	454.77	1,819,080,000

The model stores the responses in a hash map, where the key is the simplified

response string, and the value is the full response object. This results in an efficient placement and retrieval time complexity of $O(1)$. The average retrieval time of a response from the model, given the simplified response string, is 0.27 ms.

4.4 Discussion

This section discusses the methodology and the results of the Model of Blockchain Suitability (MoBS) in greater detail. Section 4.4.1 discusses the advantages of the MoBS, while Section 4.4.2 discusses the disadvantages. Finally, Section 4.4.3 discusses the impact of the research.

4.4.1 Advantages

As described in Sections 2.2.2 and 4.1, the models of Lo et al. [84], Wust and Gervais [150], and Maull et al. [86] use flow charts to arrive at restrictive conclusions. The MoBS overcomes these restrictions by diverting from the rigid flowchart approach. Users of the MoBS respond to all properties, regardless of any early answers. Answering all properties allows the framework to "catch-up"; that is, to overcome early negative results, should later results prove positive towards a particular category. This removes the necessity for the strict ordering of the properties as with a flow chart, and allows the MoBS to randomise order during the Interactive Process to remove the unconscious bias of users attempting to force a result.

Furthermore, MoBS presents comprehensible properties and statements, retaining the ease of use of the previous models. On the user's end, they merely need to select the ten aptest statements from the pre-determined lists; a similar logic to the previous models in the literature. The MoBS computes two score arrays based upon the response; the Individual Score Array (ISA) and the Domain Score Array (DSA). The previous models present fixed results towards one category, despite cases when more than one category may prove suitable. The ISA and the DSA compensate for this problem, by

offering human-readable quantitative results on the suitability of each category. The ISA suggests the best category for the user's particular response, whereas the DSA suggests the best categories relative to all the other possible suggestions. This ensures that the MoBS does not divert users from categories that may still provide benefit, and explains when there is potential to use a mixture of technologies.

The mathematical representation of the MoBS indicates the flexibility of the model. Blockchain technology is an ever-changing field of research, and the framework reflects this. It is easy to add or modify attributes, properties, statements, and suggestions to reflect the state-of-the-art of the field. The software implementation of the MoBS in this thesis subsists as a lightweight Java programme. Section 4.3.3 highlights the efficiency of this application, with the potential to expand the number of properties and statements without a negative impact on the performance.

4.4.2 Disadvantages

One disadvantage of the model is the existence of *infeasible responses*. An infeasible response is where a response reflects an "infeasible desire" of the user; something that is difficult to produce with any technology, let alone those categorised in the MoBS. For examples, a user might respond with a series of conflicting statements such as "the system has only one participant" and "not all writers are known before writing". This is a situation that should not arise. (It is not an impossible situation, but one that should not arise.) The design of the properties and statements of the MoBS ensures they do not *contradict* one another; nonetheless, this does not imply that all statements are *compatible* with one another. From one perspective, this is not a significant issue; it is unlikely that an honest user of the model will respond to the MoBS in such a manner in the first place. Furthermore, DSAs catch such responses, whereby the inevitably low DSAs suggest low suitability against other, better-suited responses. These low DSAs scores also help reinforce the higher scores, as each possible response affects the DSAs.

The current implementation only uses the attribute tags to separate properties for

readability. There is the potential to use these tags to break down the results into groups. This will indicate suitability at a more refined level of granularity, and increase the understanding of the feedback for the user. Moreover, the MoBS does not yet support suggestions for particular implementations of non-blockchain technologies. Table 4.21 is indicative of the fact that non-blockchain technologies are the most likely suggestion across every possible response. Thus, it makes sense to develop the model to support the cases where non-blockchain technologies are the suggested outcome. Nonetheless, the investigation into particular non-blockchain technologies is beyond the scope of this thesis.

4.4.3 Impact

We have acknowledged the issues of previous decision frameworks in the literature and sought to overhaul such models with a quantitative feedback approach. While this establishes a more flexible underlying model, there is no paradigm shift in user experience; users merely have to answer to a series of prompted properties with pre-selected statements. Nonetheless, the feedback provided is far richer than the feedback provided in previous models, describing which category is aptest as a percentage, and offering suggestions on which existing platforms to use. As such, the MoBS has the potential to support individuals and organisations from various industries and domains in the process of deciding whether blockchain technology is suitable for their needs, which type of blockchain is most appropriate, and which existing platforms might provide the most significant benefit.

4.5 Conclusion

This chapter has described the Model of Blockchain Suitability (MoBS), with insights into the construction and performance of the model. The Core of the MoBS is defined mathematically to indicate flexibility and provides instruction on how to expand and

modify the current implementation of the model. The current implementation performs efficiently, executing both the loading and retrieval stages in 454.77 ms. The model is most likely to suggest non-blockchain technologies, at a rate of 55.1% for high suitability and 6.1% for medium suitability, reflecting the performance advantages of non-blockchain technologies over blockchain technology. Nonetheless, private and public blockchains appear as suggestions for several responses to the model, indicating their suitability for particular use cases. This chapter proves this by testing the model against several use cases.

The development of the MoBS is important for two reasons. Firstly, it is a key component within the scope of this thesis. We use the MoBS to validate the use cases in Chapters 5 and 6. Simultaneously, we further validate and tweak the MoBS from these two use cases, forming a cyclical process of validation entirely within the context of the thesis. Secondly, and arguably more significant, this research paves the way towards the next stage of blockchain decision frameworks for real usage within academia and industry. Despite the benefits that existing frameworks provide, this chapter and Section 2.2.2 acknowledged the issues of previous decision frameworks in the literature, and sought to overhaul such models with a novel quantitative approach. While this establishes a more rigorous model, there is no paradigm shift in user experience; users merely have to answer a series of questions with pre-selected statements. This is important, as it does not over-complicate processes for those using the model; the complex underlying mechanics of the MoBS are shielded by an intuitive layer, meaning those using the model who might not have a technical background may still use the model and interpret the results. We discuss future research concerning the MoBS in Section 7.3.

Chapter 5

Towards a Distributed Data Platform for Rail Industrial Data

This chapter introduces and describes the Rail Industry Data Environment (RIDE)[1]. The RIDE is a platform designed to securely store and transfer data between industrial stakeholders using a distributed web architecture as a replacement for commonplace centralised architectures that make use of traditional databases. This approach allows the underlying Peer-to-Peer (P2P) network to distribute the storage of all the data between stakeholders, negating any limitations of network downtime and centralisation. A blockchain platform, operating in synchronous with the distributed web, controls access and ownership of data. The blockchain provides the necessary cryptographic toolkit to enforce proof-of-ownership of the data and enables the possibility of marketisation the data. The blockchain component utilises Elliptic Curve Cryptography (ECC), and the RIDE reuses the ECC keypairs to enable data security on the distributed web. It does this by adding protocols to provide secure data transfer and storage using the principles of Elliptic Curve Diffie Hellman Key Exchange (ECDH) (to enable users to share keys) and the Advanced Encryption Standard (AES) (to encrypt the data). (Furthermore, this chapter specifies a possible post-quantum version of the RIDE, that uses

[1]This work (and therefore this chapter) is strongly related to that presented in one of the author's publications [108].

Supersingular Isogeny Cryptography (SIC) in place of classical ECC.) Thus, it makes redundant the traditional method of bespoke Application Programming Interfaces (APIs) and restrictive web portals to provide access to siloed databases to obtain data.

Section 5.1 establishes the aims of the RIDE. Furthermore, it explains the necessity to investigate both classical and post-quantum cryptographic approaches to the mechanisms of the RIDE. Section 5.2 describes the methodology of the research undertaken. First, we discuss the decisions taken to arrive at a final proof-of-concept for the RIDE. After this, we describe the development of a suitable post-quantum blockchain based on SIC, with focus on the implementation of a Supersingular Isogeny Digital Signature Algorithm (SIDSA) by Yoo et al. [151] and Galbraith et al. [55]. We also identify a subtle improvement to the SIDSA and present this here. Section 5.3 analyses the performance of the new SIDSA. It then presents a walk-through of the RIDE (both the classical and post-quantum versions) to illustrate the process and confirm the mechanics perform correctly. Section 5.4 provides a security proof for the new SIDSA, and discusses the impact the RIDE will have for the future of data sharing and marketisation in the industry. Section 5.5 concludes the findings of this chapter by summarising the results and discussion.

5.1 Introduction

Since the advent of the World Wide Web (WWW) in the early 1990s, internet users have adhered to transport protocols supported by the World Wide Web Consortium (W3C). While these protocols perform to a standard that most users regard as adequate, the outdated network architectures that the WWW relies upon are in desperate need of modernisation to reflect modern practices. Instead of the reliance upon centralised servers, the Internet should distribute information across the nodes of the network to make the accessibility of data both easier and faster. For example, using the distributed approach to streaming a video can save up to 60% in bandwidth costs over traditional

architecture [98]. Moreover, as identified in Section 2.4.1, data sharing in a rail industry context is a fundamental problem, as centralised architectures struggle to integrate. Moving to a distributed solution is a potential solution to the restrictive nature of centralised networks. However, this distributed approach comes the problem of sharing sensitive data. The nature of a distributed web means that data on the network is visible to anybody who wishes to access it. Thus, encryption is paramount before uploading data to ensure the secrecy of any sensitive data.

Local system[2] self-encryption methods are straightforward; a user (Alice) encrypts a file with a symmetric key (from a relevant symmetric cryptosystem such as the Advanced Encryption Standard (AES) and a suitable mode), stores the encrypted data, and then uses the symmetric key to decrypt the file when required. The symmetric key is never available in any public domain. The issue arises when Alice needs to share the file with another user; let us call him Bob. Transporting the encrypted data is easy, but securely transferring the symmetric key required to decrypt this file is difficult without exposing it to adversaries. As such, Diffie and Hellman [48] developed the principle of Diffie-Hellman Key Exchange (DHKE), to enable remote shared key derivation between two users. This technology (and the many variations developed) can work symbiotically with symmetric cryptography and other technologies to provide secure key transit. For example, Transport Layer Security (TLS) defines both critical exchange and cipher suites for key negotiation that operate safely on traditional network architectures [116].

Nevertheless, there is nothing that strictly defines these principles of security concerning distributed web architectures. The Rail Industry Data Environment (RIDE) utilises the definitions of TLS and adapts them to operate on a distributed network. Beyond this, the RIDE encompasses the proof-of-ownership and access permissions by utilising a blockchain platform to work in synchronous with the distributed web. Using a blockchain platform alongside the distributed web provides an array of benefits: it records transaction history; it enables users to prove their digital identities; it creates a

[2]We define a local system as a system that only one party can access.

robust platform of provenance and shared responsibility; and it opens the possibility of marketising data by using digital tokens.

In particular, the intention of the RIDE is to allow rail industry stakeholders to distribute encrypted data amongst one another via a distributed Peer-to-Peer (P2P) network, such that there are no data silos and no centralised points of storage, thus overcoming the existing barriers of current solutions (discussed in Section 2.4.1). However, the RIDE only entitles these stakeholders to see certain pieces of data, depending on the permissions of the original data owner. As such, it is not merely a platform to dump data but to control the access and sell it to interested parties. With this in mind, the platform offers the potential to monetise the transactions, creating an instant-access marketplace for rail data. The aims of the RIDE are:

1. Allow stakeholders to upload data securely to a distributed storage platform.

2. Allow stakeholders to share (and monetise) this data with other users on the platform.

3. Use blockchain technology to trace and audit ownership and permissions of the data.

The mechanics of the RIDE suit both classical (pre-quantum) and post-quantum domains of cryptography. With regards to the quantum approach to the system, this chapter presents a new post-quantum based blockchain platform. As Section 2.2.2 suggests, researchers are looking at methods of developing post-quantum blockchains; however, the RIDE requires cryptosystem keypairs compatible with both a Key Exchange Scheme (KES) and a Digital Signature Scheme (DSS). Current approaches to post-quantum blockchains, such as hash-based schemes, do not yet permit this, as there are no compatible KESs. Thus, one suitable potential candidate is the Supersingular Isogeny Cryptography (SIC), based on the mathematics of supersingular isogenies of elliptic curves.

We are aware of the development of the Ocean Protocol platform alongside this

research. Despite similarities in goals and aims, this chapter provides details into the mechanics of the platform, and explores both classical and post-quantum options. We also investigate fully distributed data using distribited web technologies.

5.2 Methodology

This section discusses the development of the Rail Industry Data Environment (RIDE). Section 5.2.1 analyses the suitability of the use case that the RIDE aims to satisfy (as per the aims in Section 5.1) through use of the Model of Blockchain Suitability (MoBS) developed in Chapter 4. Section 5.2.2 describes the various approaches to data storage and collaboration protocols. We first approach this in a conceptual manner to provide readers with an overview of the RIDE. After this, we investigate the technical implementations. Section 5.2.3 introduces the Supersingular Isogeny Digital Signature Algorithm (SIDSA) required to operate a post-quantum version of Ethereum used in the post-quantum implementations of the RIDE.

5.2.1 Suitability of Blockchain Technology

Prior to developing a platform upon the hypothesis presented in Section 5.1, we sought to verify the suitability of the application of blockchain technology for this use case. To achieve this, we utilised the MoBS developed in Chapter 4, which was beneficial to this chapter and overall thesis for two reasons. The first was to use the MoBS for its main purpose; to obtain detailed results to provide an informed decision as to which technology to approach this use case with. The second was to provide additional clarification and development of the MoBS as part of this thesis, refining and verifying the work in Chapter 4. These two reasons formed cyclical benefits, whereby we iteratively improved the MoBS whilst still providing concrete justifications.

In terms of the results, Table 5.1 illustrates the response of this use case to the MoBS, taking the aims provided in Section 5.1 into account. The resultant ISA indicates

Table 5.1: The Individual Score Array (ISA) and Domain Score Array (DSA) of the response to the MoBS for the RIDE use case.

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S1	0	0	1	1	-1	-1
A1P2S1	-1	-1	1	2	0	-1
A1P3S1	-1	-2	1	3	-1	-2
A1P4S1	0	-2	1	4	-1	-3
A2P1S1	-1	-3	1	5	1	-2
A2P2S1	1	-2	-1	4	-1	-3
A2P3S1	-1	-3	1	5	1	-2
A3P1S3	1	-2	1	6	-1	-3
A3P2S2	1	-1	0	6	-1	-4
A3P3S1	1	0	-1	5	-1	-5
Total Score	0		5		-5	
Adjusted Score	0		5		0	
ISA (%)	0		100		0	
DSA (%)	0		50		0	

a 100% suitability for private blockchain to the RIDE, as opposed to non-blockchain technologies and a public blockchain. Moreover, the DSA provides scores of 0% for non-blockchain technologies, 50% for private blockchain technologies, and 0% for public blockchain technologies. This is indicative of a medium level of suitability (when compared against every use case) for private blockchain, and a low level of suitability for non-blockchain and public blockchain.

By analysing the results, we confirm that the MoBS clearly endorses private blockchain technology over non-blockchain and public blockchain. The significant factor to note is the medium level of suitability provided by the DSA. As per Table 4.13, a medium level of suitability suggests to "consider using with other medium categories, or forming hybrid approach with additional technologies." In this case, the other categories both offer a low suitability categories, rendering their usage questionable. However, the MoBS provides the option to form hybrid approaches with additional technologies. By investigating the recommended platforms, the MoBS provides greater insight as to

why the DSA scores are as they are. The MoBS suggests utilising off-chain storage to supplement the private blockchain as the primary mechanism. As such, the evolution of the RIDE proceeds with the symbiotic development of distributed web and blockchain technologies, in line with the suggestions of Table 4.13.

With this analysis in mind, this chapter forms a proof-of-concept that uses two technologies. The first is the private blockchain technology; in this case, a local implementation of the Ethereum platform. Though the main Ethereum network is deployed as a public blockchain, it is a useful tool to test proofs-of-concept for private blockchains by initiating a private network. Results may then translate to more sophisticated platforms, such as those under the Hyperledger umbrella. The second technology is the distributed web platform; in this case, we use the InterPlanetary File System (IPFS). Once more, this is a suitable tool for testing and lightweight proofs-of-concept which may evolve to other similar technologies.

5.2.2 Data Collaboration Protocols

The RIDE operates over a series of steps. Assume users of the RIDE own a blockchain keypair to interact with the system. There are at least two agents at play in the framework: Alice, who wishes to commit and send data; and Bob, who wishes to retrieve data from Alice. Eve, who wishes to actively intercept and read the data without prior permission, is also considered.

Conceptual Approaches

There are two conceptual approaches to the RIDE. (For the concepts, we do not specify the cryptographic protocols in use; the technical approaches define these later in this section.) The RIDE One-Key Approach (RIDE-1) encrypts the plaintext data with a symmetric key derived from a shared key between two users; there is only *one* symmetric key between interacting users, hence RIDE-1. Alternatively, the RIDE Two-

Key Approach (RIDE-2) encrypts the plaintext with a randomly generated symmetric key and then proceeds to encrypt this symmetric key with the shared symmetric key between two users (analagous to a Key Encryption Key (KEK) system); hence RIDE-2. A whitelist, stored as part of a smart contract on the blockchain, determines who users will share their data with.

RIDE One-Key Approach Figure 5.1 illustrates the process of the RIDE-1. The disad-

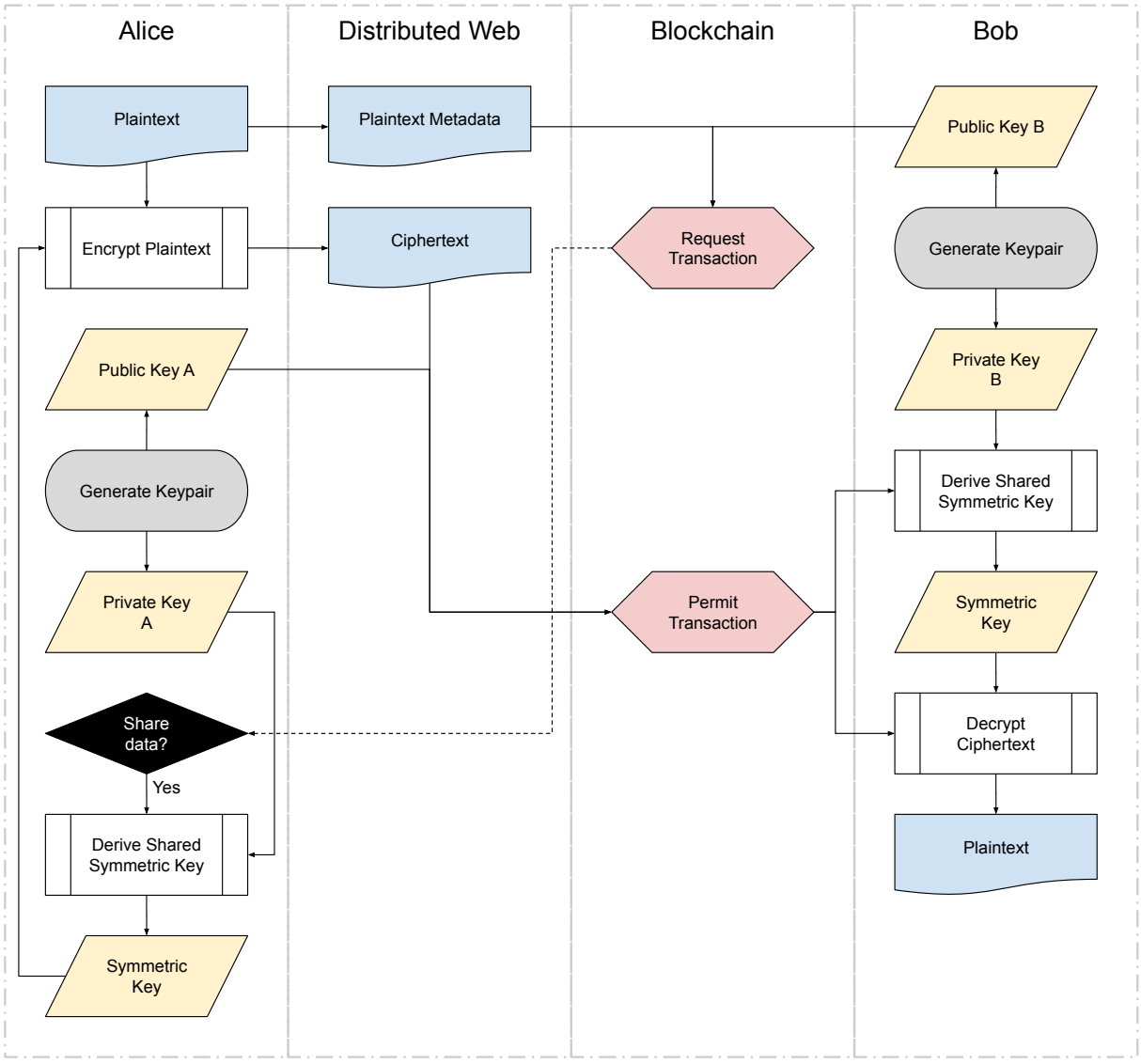


Figure 5.1: The process of the RIDE-1. Dashed lines and solid lines are synonymous, and only different to clarify crossover points.

vantage of the RIDE-1 concerns the fact that for each pair of blockchain accounts, there

is only one shared key. Should Alice wish to share a piece of data with n users, she will have to establish n symmetric keys and encrypt the data n times. Encrypting data n times is a computationally demanding process, and results in an exponential increase in the number of uploads to the IPFS.

RIDE Two-Key Approach To overcome the issue identified with the RIDE-1, we developed the RIDE-2. Figure 5.2 illustrates the process of the RIDE-2. The advantage of the RIDE-2 is that each piece of data uses a unique randomly generated symmetric key which encrypts the data before other users make any requests. If two users agree to share data, they establish their shared symmetric key via their respective keypairs and use this shared key to encrypt the randomly generated key. The RIDE-2 maintains one shared key between the users while protecting the original symmetric key should Alice wish to withhold the data from Bob.

Technical Approaches

The conceptual approaches translate into technical approaches through the application of existing platforms and cryptosystems. The classical approaches (the RIDE-1 Classical (RIDE-1C) and the RIDE-2 Classical (RIDE-2C)) use a standard version of Ethereum with Elliptic Curve Cryptography (ECC) keypairs, compatible with Elliptic Curve Diffie Hellman Key Exchange (ECDH) protocols. The quantum approaches (the RIDE-1 Quantum (RIDE-1Q) and the RIDE-2 Quantum (RIDE-2Q)) aim to use a modified version of Ethereum that uses Supersingular Isogeny Cryptography (SIC) keypairs, so to make it compatible with Supersingular Isogeny Diffie Hellman Key Exchange (SIDH). As such, Section 5.2.3 creates a modified Ethereum platform that utilises the 330-byte public keys used in SIC in order to achieve the same framework mechanics of the previous two implementations. For symmetric encryption, the RIDE uses the Galois/Counter Mode (GCM) mode of the Advanced Encryption Standard (AES), to meet the standards of Transport Layer Security (TLS) 1.3.

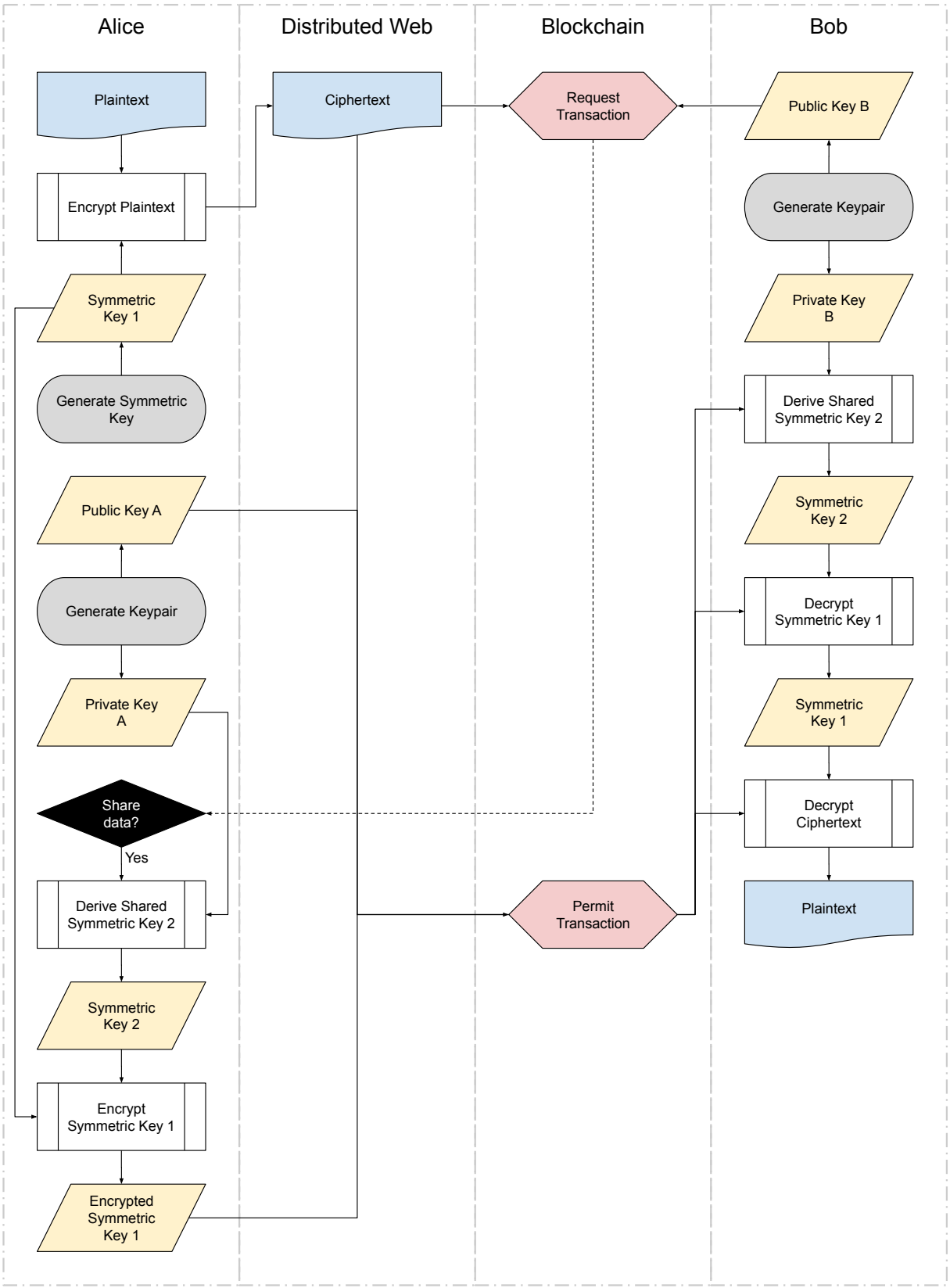


Figure 5.2: The process of the RIDE-2. Dashed lines and solid lines are synonymous, and only different to clarify crossover points.

RIDE-1 Classical The RIDE-1C utilises the ECDH methodology by enabling the two parties to generate a shared secret coordinate on an elliptic curve (x, y) independent of one another. The process derives a robust shared symmetric key k by passing the x-coordinate of the shared secret through the Hash-Based Message Authentication Code Key Derivation Function (HKDF). Alice uses k to encrypt her plaintext $\mathcal{P}$. There are a set of domain parameters D (explained in Section 2.3.3 by Equation 2.16), and Alice and Bob generate keypairs (Q_A, d_A) and (Q_B, d_B) respectively. The ECC keypairs form the basis of the ECDH process and the Elliptic Curve Digital Signature Algorithm (ECDSA). As such, they are used as the blockchain addresses. The list below describes the process of the RIDE-1C, and Figure 5.1 provides a supplementary illustration.

A1 Alice uploads metadata about the plaintext $\mathcal{P}$ to the IPFS.

B1 Bob invokes a request transaction to obtain $\mathcal{P}$ that the metadata describes.

A2 The platforms notifies Alice of the request transaction on the blockchain. If Bob's address is on her whitelist, and Bob has the correct sum to purchase the data, Alice calculates the point (x, y) on the elliptical curve E such that

$$(x, y) = d_A Q_B.$$

If Bob is not on Alice's whitelist, Alice may decide whether to append Bob to the whitelist and continue the process of sharing the data with him, or to halt the process.

A3 Alice passes the shared secret x (the x coordinate of the computed point on E) through the HKDF to derive a robust shared symmetric key k.

A4 Alice encrypts $\mathcal{P}$ with k using the GCM mode of AES to achieve the ciphertext $\mathcal{C}$ and a Message Authentication Code (MAC).

A5 Alice uploads $\mathcal{C}$ and MAC to the IPFS.

A6 Alice invokes a permit transaction to inform Bob that C is on the IPFS, providing the hash location of the data.

B2 Bob computes the point (x, y) on E such that

$$(x, y) = d_B Q_A.$$

B3 Bob passes the shared secret x through the HKDF to derive the shared symmetric key k.

B4 Bob accesses $\mathcal{C}$ from the IPFS and uses k to decrypt it to achieve $\mathcal{P}$.

RIDE-2 Classical As with the RIDE-1C, assume that Alice and Bob have generated keypairs (Q_A, d_A) and (Q_B, d_B) respectively. The full process of the RIDE-2C is described below, and Figure 5.2 illustrates this process. For this process, we utilise the processes of the Elliptic Curve Integrated Encryption Scheme (ECIES).

A1 Alice generates a symmetric key k_1 and encrypts the plaintext $\mathcal{P}$ with k_1 to form the ciphertext $\mathcal{C}$.

A2 Alice uploads $\mathcal{C}$ with a metadata header to the IPFS.

B1 Bob invokes a request transaction to obtain k_1, in order to decrypt $\mathcal{C}$ to obtain $\mathcal{P}$.

A2 The platforms notifies Alice of the request transaction on the blockchain. If Bob's address is on her whitelist, and Bob has the correct sum to purchase the data, Alice calculates the point (x, y) on the elliptical curve E such that

$$(x, y) = d_A Q_B.$$

If Bob is not on Alice's whitelist, Alice may decide whether to append Bob to the whitelist and continue the process of sharing the data with him, or to halt the process.

A4 Alice passes the shared secret x (the x coordinate of the computed elliptical curve point) through the Key Derivation Function (KDF) of the ECIES to derive a shared symmetric key k_2, and k_3 to generate a MAC.

A5 Alice encrypts k_1 with the GCM mode of the AES, using k_2 as the key, to form an encrypted symmetric key $k_{1,\text{enc}}$.

A6 Alice invokes a permit transaction to send Bob $k_{1,\text{enc}}$ and the MAC.

B2 Bob receives the permit transaction and computes the point (x, y) on E such that

$$(x, y) = d_B Q_A.$$

B3 Bob passes the shared secret x through the ECIES KDF to derive the symmetric key k_2.

B4 Bob uses k_2 to decrypt $k_{1,\text{enc}}$ to achieve k_1.

B5 Bob uses k_1 to decrypt $\mathcal{C}$ to achieve $\mathcal{P}$.

RIDE-1 Quantum The RIDE-1Q utilises SIDH; a post-quantum version of Diffie-Hellman Key Exchange (DHKE) that uses supersingular isogenies of elliptic curves. This ensures security against adversaries with quantum computing capabilities. Prior to engagement, Alice and Bob generate the relevant parameters. Alice chooses two public basis points (P_A, Q_A) on E with order of 2^{e_A}. From this, Alice generates a secret point S_A such that

$$S_A = P_A + [k_A]Q_A, k_A \in [0, 2^{e_A}). \tag{5.1}$$

Once Alice has calculated S_A, she computes the secret isogeny ϕ_A such that

$$\phi_A : E \to E/\langle S_A \rangle \tag{5.2}$$

where

$$E/\langle S_A \rangle \equiv E_A \equiv \phi_A(E). \tag{5.3}$$

Bob mirrors the steps Alice takes; the main difference being that Bob chooses base points (P_B, Q_B) with an order of 3^{e_B}. Note that SIDH requires points of different orders between a pair establishing a shared key. This is compensated by each user generating public points on E with both order of 2^{e_A} and 3^{e_B}. Each user holds two accounts on the Ethereum network using these public points; the former for a permitting account (which uploads files and grants permits), and the latter for a requesting account (which requests files). The process below indicates the steps taken between Alice and Bob during the RIDE-1Q. Figure 5.1 illustrates the process.

A1 Alice uploads metadata about the plaintext $\mathcal{P}$ to the IPFS.

B1 Bob invokes a request transaction to obtain $\mathcal{P}$ that the metadata describes.

A2 The platforms notifies Alice of the request transaction on the blockchain. If Bob's address is on her whitelist, and Bob has the correct sum to purchase the data, Alice calculates

$$S_{AB} = \phi_B(P_A) + [k_A]\phi_B(Q_A)$$

on the curve E_B in order to compute the isogeny ϕ_{AB}, such that

$$\phi_{AB} : E_B \to E_B/\langle S_{AB} \rangle,$$

where

$$E_B/\langle S_{AB} \rangle = E_{AB}.$$

If Bob is not on Alice's whitelist, Alice may decide whether to append Bob to the whitelist and continue the process of sharing the data with him, or to halt the process.

A3 Alice passes the shared secret $j_{AB} = j(E_{AB})$ through the HKDF to derive the shared symmetric key K.

A4 Alice encrypts $\mathcal{P}$ with K and the GCM mode of AES to achieve the ciphertext $\mathcal{C}$.

A5 Alice uploads $\mathcal{C}$ to the IPFS.

A6 Alice invokes a permit transaction to inform Bob that $\mathcal{C}$ is on the IPFS.

B2 Bob receives the permit transaction and calculates

$$S_{BA} = \phi_A(P_B) + [k_B]\phi_A(Q_B)$$

in order to compute the isogeny

$$\phi_{BA} : E_A \to E_A/\langle S_{BA}\rangle,$$

where

$$E_A/\langle S_{BA}\rangle = E_{BA}.$$

B3 Bob passes the shared secret $j_{BA} = j(E_{BA})$ through the HKDF to derive the shared symmetric key K.

B4 Bob accesses $\mathcal{C}$ from the IPFS and uses K to decrypt it to achieve $\mathcal{P}$.

RIDE-2 Quantum The RIDE-2Q uses the same keypairs as the RIDE-1Q. The process below indicates the steps taken between Alice and Bob during the RIDE-2Q. Figure 5.2 illustrates the process.

A1 Alice generates a symmetric key k_1 and encrypts the plaintext $\mathcal{P}$ with k_1 to form the ciphertext $\mathcal{C}$.

A2 Alice uploads $\mathcal{C}$ to the IPFS.

B1 Bob invokes a request transaction to obtain $\mathcal{P}$ from $\mathcal{C}$. Within this transaction, Bob provides $E_B = E/\langle S_B\rangle$, $\phi_B(P_A)$, and $\phi_B(Q_A)$.

A3 The platforms notifies Alice of the request transaction on the blockchain. If Bob's address is on her whitelist, and Bob has the correct sum to purchase the data, Alice calculates

$$S_{AB} = \phi_B(P_A) + [k_A]\phi_B(Q_A)$$

on the curve E_B in order to compute the isogeny ϕ_{AB}, such that

$$\phi_{AB} : E_B \to E_B/\langle S_{AB} \rangle,$$

where

$$E_B/\langle S_{AB} \rangle = E_{AB}.$$

If Bob is not on Alice's whitelist, Alice may decide whether to append Bob to the whitelist and continue the process of sharing the data with him, or to halt the process.

A4 Alice passes the shared secret $j_{AB} = j(E_{AB})$ through the HKDF to derive the shared symmetric key k_2.

A5 Alice encrypts k_1 with the GCM mode of AES, using k_2 as the key, to form an encrypted symmetric key κ_1.

A6 Alice invokes a permit transaction to send Bob κ_1, E_A, $\phi_A(P_B)$, and $\phi_A(Q_B)$.

B2 Bob receives the permit transaction and calculates

$$S_{BA} = \phi_A(P_B) + [k_B]\phi_A(Q_B)$$

in order to compute the isogeny

$$\phi_{BA} : E_A \to E_A/\langle S_{BA} \rangle,$$

where

$$E_A/\langle S_{BA} \rangle = E_{BA}.$$

B3 Bob passes the shared secret $j_{BA} = j(E_{BA})$ through the HKDF to derive the shared symmetric key k_2.

B4 Bob uses k_2 to decrypt the κ_1 to achieve k_1.

B5 Bob uses k_1 to decrypt $\mathcal{C}$ to achieve $\mathcal{P}$.

5.2.3 Integration of Quantum-Resistant Cryptography

The fundamental principle of the RIDE is to utilise a Key Exchange Scheme (KES) and a Digital Signature Scheme (DSS) with the same keypair to avoid redundancy and increase efficiency. This principle does not pose an issue with classical ECC, as the same keypair is usable for both ECDH and the ECDSA. The majority of approaches to post-quantum blockchains use cryptosystems that do not have an associated KES. Hence, this section develops a blockchain platform that uses SIC and the relevant schemes of SIDH and the SIDSA.

Integration With Existing Platforms

The RIDE proof-of-concept integrates the SIDSA with the Ethereum platform by overhauling the existing classical ECDSA. Assume that public parameters are an elliptic curve E, two points on the elliptic curve (P_A, Q_A) of the 2-torsion group with order of 2^{e_A}, and two points on the elliptic curve (P_B, Q_B) of the 3-torsion group with order of 3^{e_B}. Because SIC works on the principle of keys in these different torsion groups, and RIDE users will likely seek to share and receive data, users require two sets of key-pairs. (To demonstrate why this is necessary, imagine that Alice and Bob both used the points of the 2-torsion group to generate their secret keys; this would mean that no secure exchange through SIDH could happen.) The private key for sending S_2 is a point on E such that

$$S_2 = P_A + [k]Q_A, k \in [0, 2^{e_A}). \tag{5.4}$$

Note the secret key for sending is in the 2-torsion group, hence S_2. The public key is the elliptic curve $E/\langle S \rangle$, such that

$$\phi : E \to E/\langle S \rangle. \tag{5.5}$$

The private key for receiving S_3 is a point on E such that

$$S_3 = P_B + [k]Q_B, k \in [0, 3^{e_B}). \tag{5.6}$$

Note the secret key for receiving is in the 3-torsion group, hence S_3. The public key is the elliptic curve $E/\langle S \rangle$, such that

$$\phi : E \to E/\langle S \rangle. \tag{5.7}$$

The RIDE utilises both these keypairs in both the key exchange protocol and as the blockchain keypair. This is one of the disadvantages of the SIC approach to the RIDE and post-quantum blockchains in general; two keys makes the benefit of smaller key sizes of SIC redundant.

An Improved Digital Signature Scheme

By retaining the mechanics of the Non Interactive Zero Knowledge Proof (NIZKP) used in the DSSs proposed by Yoo et al. [151] and Galbraith et al. [55], further compression of the signature is attainable. Note that the version by Yoo et al. [151] sends the compressed curves $E/\langle R \rangle$ and $E/\langle R, S \rangle$ of 12λ bits each, costing 24λ for each round of the NIZKP protocol as demonstrated in Equation 2.25. The version offered by Galbraith et al. [55] uses the j-invariant of 3λ bits, demonstrated in Equation 2.26. Instead, the RIDE uses a 256-bit hash of the j-invariant. This reduces the number of bits required for a signature to

$$2\lambda \left(256 + 256 + 1 + 3\lambda + \frac{3\lambda + 3\lambda}{2} \right) = \lambda \left(12\lambda + 1026 \right). \tag{5.8}$$

This reduction of bits requires a modification to the signing algorithm in Algorithm 2.5 and the verification algorithm in Algorithm 2.6. Algorithm 5.1 and Algorithm 5.2 illustrate these changes.

The Non Interactive Zero Knowledge Proof Stage Cache Algorithm 5.1 has two distinct stages; the *NIZKP* stage, and the *signing* stage. The NIZKP stage does not require the message, which means it is possible to precompute the commitments, challenge bits, and hashed responses before the signing stage. Alice, who wishes to authenticate a series of messages to Bob, precomputes a batch of NIZKP stages.

Algorithm 5.1: Compressed `Sign` algorithm

func `Sign(prv, message)`**:**
  for $i \leftarrow 0$ **to** $2\lambda - 1$ **do**
    Generate a random scalar value $k_B \in [0, 3^{e_B})$
    Compute the point $R \leftarrow P_B + [k_B]Q_B$
    Compute the isogeny $\psi : E \rightarrow E/\langle R \rangle$
    Compute the isogeny $\phi' : E/\langle R \rangle \rightarrow E/\langle R, S \rangle$
    Compute the isogeny $\psi' : E/\langle S \rangle \rightarrow E/\langle R, S \rangle$
    Assign commitment$[i][0] \leftarrow H(j(E/\langle R \rangle))$
    Assign commitment$[i][1] \leftarrow H(j(E/\langle R, S \rangle))$
    Generate a bit $b \xleftarrow{\text{rand}} \{0, 1\}$
    Assign challenge$[i] \leftarrow b$
    if challenge$[i] = 0$ **then**
      response$[i][0] \leftarrow (R, \phi(R))$
      response$[i][1] \leftarrow \psi(S)$
    else
      response$[i][0] \leftarrow \psi(S)$
      response$[i][1] \leftarrow (R, \phi(R))$
    hash$[i][0] \leftarrow G($response$[i][0])$
    hash$[i][1] \leftarrow G($response$[i][1])$
  end
  Generate a stream of bits bits $\leftarrow H($pub, message, commitment, challenge, hash$)$
  for $i \leftarrow 0$ **to** $2\lambda - 1$ **do**
    signature$[i] \leftarrow$ commitment$[i]$, challenge$[i]$, hash$[i]$, response$[i][$bits$[i]]$
  end
  return signature

Once she is ready to send the message, she selects a precomputed stage at random and completes the signature. A cache stores these batches until Alice requires them, indicative that the SIDSA performs the majority of isogeny calculations *before* to the signing stage. With this in mind, it reduces the time required to sign a message, as it has already completed the majority of the protocol. The practical implication of caching the NIZKP stage is that Alice can precompute a large batch of these stages, before having to sign any transactions; thus, signing a transaction is faster. Nonetheless, there is no similar approach to the Verify algorithm; thus, there is no time reduction when verifying a signature, even if it has use the NIZKP stage cache.

Algorithm 5.2: Compressed `Verify` algorithm

```
func Verify(pub, message, signature):
    bits ← H(pub, message, commitment, challenge, hash)
    for i ← 0 to 2λ − 1 do
        if hash[i][0] = H(response[i][0])  &  hash[i][1] = H(response[i][1]) then
            if challenge[i][0] = 0 then
                Check R has an order of 3^{e_B}
                Compute the isogeny ψ : E → E/⟨R⟩
                Check commitment[i][0] = H(j(E/⟨R⟩))
                Compute the isogeny ψ' : E/⟨S⟩ → E/⟨R, S⟩
                Check commitment[i][1] = H(j(E/⟨R, S⟩))
            else
                Check ψ(S) has the order 2^{e_A}
                Compute the isogeny φ : E → E/⟨ψ(S)⟩
                Check commitment[i][1] = H(j(E/⟨R, S⟩))
        else
            return false
    end
    return true
```

5.3 Results

This section explains the tests conducted to achieve the results for the Rail Industry Data Environment (RIDE). Section 5.3.1 describes the results of the modified Supersingular Isogeny Digital Signature Algorithm (SIDSA). Section 5.3.2 explains the tests to check the RIDE Classical Approach (RIDE-C) implementations, and provides average time-based results for uploading and retrieving a piece of data on the RIDE. Section 5.3.3 is identical to Section 5.3.2, but replaces the RIDE-C implementations with the RIDE Quantum Approach (RIDE-Q) implementations.

5.3.1 Quantum Resistant Encryption

The majority of blockchain platforms use the Go programming language in their main implementations. Because of this, the implementation of the SIDSA uses Go for ease of integration with existing blockchain platforms. The Digital Signature Scheme (DSS) uses the existing implementation of Supersingular Isogeny Diffie Hellman Key Ex-

change (SIDH) in Go developed by de Valence [45], who bases their open-source implementation upon the work of Jao and De Feo [72], Costello et al. [36], Jao et al. [74], and provides access to efficient algorithms to work with supersingular isogenies. The SIDSA required the internal package to access the internal isogeny arithmetic and static parameters. It made use of curve `p751` where the public parameters are as follows: the curve $E : y^2 = x^3 + x$; the prime $2^{372} \cdot 3^{239}$ of bit-length 751; and relevant generators P_a, P_b, Q_a, and Q_a. Go's `testing` package provides the necessary tools to create all tests and benchmarks. These benchmark tests ran on an Intel Core i5-6500 Skylake processor clocking at 3.2 GHz, with the compilation of the code on 64-bit Ubuntu 18.04.2 LTS.

Compression Given a security parameter $\lambda = 128$, Equations 2.25, 2.26, and 5.8 compute the average signature sizes for the SIDSA by Yoo et al. [151], the SIDSA by Galbraith et al. [55], and the SIDSA proposed in this chapter respectively. Table 5.2 summarises the results.

Table 5.2: Public key sizes, private key sizes, and signature sizes for different approaches of the SIDSA.

	Size (bytes)		
Scheme	Public Key	Private Key	Signature
Yoo et al. [151]	768	48	132,649
Galbraith et al. [55]	336	48	49,184
Proposed	336	48	40,992

Note that there is a 69.1% decrease in signature size from the SIDSA proposed by Yoo et al. [151], and a 16.7% decrease from the SIDSA proposed by Galbraith et al. [55], due to the hashing of the j-invariant.

General Performance In order to remove any discrepancies between programming languages, Table 5.3 provides performance results of the proposed SIDSA against the original SIDSAs. Running the Go benchmarks for 600 seconds obtained averages for the various DSSs. There is a negligible difference in computation time between the

Table 5.3: Performance of the signing and verification algorithms.

Algorithm	Threads	ns op^{-1}	clock cycles	B op^{-1}	allocs op^{-1}
Sign	1	7,576,259,344	24,244,029,901	1,524,894,830	6,334,954
(Galbraith)	2	3,945,928,327	12,626,970,646		
	4	2,125,308,930	6,800,988,576		
Sign	1	7,552,441,727	24,167,813,526	1,525,014,103	6,336,438
(Proposed)	2	3,945,071,743	12,624,229,578		
	4	2,126,280,111	6,804,096,355		
Verify	1	5,151,856,354	16,485,940,333	375,463,448	1,586,671
(Galbraith)	2	2,722,690,546	8,712,609,746		
	4	1,572,728,608	5,032,731,546		
Verify	1	5,135,660,374	16,434,113,198	367,124,831	1,551,955
(Proposed)	2	2,722,099,503	8,710,718,409		
	4	1,573,447,282	5,035,031,303		

algorithm of Galbraith et al. [55] and the proposed algorithms, suggesting a low space-time trade-off between the algorithms. The signing algorithm takes 0.05% additional time to complete and requires an average increase of 0.01% bytes per operation and 0.02% allocations per operation. Verification takes an additional 0.05% and requires an increase of 2.27% bytes per operation, and an increase if 2.24% of allocations per operation; this is also a negligible result.

Cached Performance Further to the results above, Table 5.4 provides performance results for the isolated Non Interactive Zero Knowledge Proof (NIZKP) stage within the signing algorithm. The scheme isolates the NIZKP round from the signing stage because the message is not required. The signing stage of the algorithm uses the precomputed batch of commitments, challenge bits, and hashed responses to sign the message as per Algorithm 5.1. In comparison to Table 5.3, the NIZKP stage takes an average 99.49% of the overall run-time of the signing algorithm. Furthermore, it takes 99.78% of the overall run-time of the compressed signing algorithm. These values demonstrate that a precomputed NIZKP round signs a message in 12,121,466 clock cycles.

Table 5.4: Performance of the NIZKP rounds of the signing algorithm.

Algorithm	Threads	ns op^{-1}	clock cycles	B op^{-1}	allocs op^{-1}
Galbraith	1	7,517,151,499	24,054,884,797	1,524,683,641	6,334,930
	2	3,938,615,053	12,603,568,170		
	4	2,122,714,680	6,792,686,976		
Proposed	1	7,526,826,659	24,085,845,309	1,524,937,156	6,336,419
	2	3,946,148,892	12,627,676,454		
	4	2,121,520,972	6,788,867,110		

5.3.2 RIDE Classical Approach

In order to test the various approaches to the RIDE, we investigated the processes with a number of test models on a private instance of the Ethereum blockchain platform, using the javascript Truffle suite to aid testing and deployment. For the sake of the test model, the plaintext $\mathcal{P}$ was data extracted from the Rail Delivery Group (RDG)'s fares, routing, and timetable data. Initially, we generated two Elliptic Curve Cryptography (ECC) keypairs to form the Ethereum accounts of Organisation A and Organisation B. Ethereum accounts hash their public keys to form an address. In order to retrieve the original ECC public key, one must receive a transaction from the account to obtain the values of r, s, and v to compute it. The `ethereum-utils` package provides the functionality to achieve this.

RIDE-1 Classical

Assume Organisation A wishes to sell a FLOW file type (a rail specific file), where Organisation B and Organisation C are both permitted to purchase this data. Organisation A proceeds to deploy a smart contract for the document, which includes a `mapping` named `whitelist` which maps the addresses to a boolean value that indicates whether that address is entitled to purchase the data. Organisation A invokes the `update` contract to legitimise Organisation B and Organisation C's addresses (and any additional addresses they approves of) within `whitelist`. The contract restricts this invocation to

Organisation A, as the owner of the contract. Organisation A then uploads metadata about $\mathcal{P}$ onto the InterPlanetary File System (IPFS). For the RIDE-1 Classical (RIDE-1C), the metadata is a JavaScript Object Notation (JSON) file that contains the title of the data, the author of the data, and Organisation A's Ethereum address. Organisation A also updates an index page, to enable other users to view the addresses of the metadata. Organisation B and Organisation C invoke the `request` method from the smart contract to request $\mathcal{P}$ from Organisation A by providing their addresses and signing the transactions. The smart contract receives this transaction and updates a `mapping` of addresses that have requested $\mathcal{P}$. Organisation A frequently checks the state of this mapping to determine whether they needs to issue any permits. The contract eases this process by updating the event log upon invocation, instantly notifying Organisation A of the alteration of state.

Organisation A proceeds to derive the shared secret x_B between them and Organisation B, and x_C between them and Organisation C. To achieve this, they derives their original ECC public key via the Ethereum address and the r, s, and v values of the signature of the request transaction to the smart contract. Then they computes the shared coordinates x via the Elliptic Curve Diffie Hellman Key Exchange (ECDH) protocol. Following this, Organisation A passes the shared secrets through the Hash-Based Message Authentication Code Key Derivation Function (HKDF) to derive two 256-bit keys k_B and k_C for the Galois/Counter Mode (GCM) of the Advanced Encryption Standard (AES). The GCM takes $\mathcal{P}$, the key k, and a uniformly-randomised initialisation vector to compute the ciphertext $\mathcal{C}$, and prepares the information for secure transit. Organisation A first uploads $\mathcal{C}$ to the IPFS at the respective `cid`. Once $\mathcal{C}$ is online, Organisation A invokes the `permit` method of the smart contract to inform Organisation B and Organisation C of the address of $\mathcal{C}$ on the IPFS. Organisation A signs this transaction to provide authentication of the document. Organisation B and Organisation C then proceed to mirror Organisation A's actions and derive the identical x and k by utilising their public key and the ECDH process. Once they have established k, they use the

AES to retrieve the original $\mathcal{P}$. Utilising a computer with an Intel i7-6700k processor, this sequence of events takes 4183 ms. (The sequence ran 100 times.) It is important to note that the transactions on this private network are instantaneous, as the mining parameters are modified to create new blocks for each transaction. The RIDE-1C was repeated with an array of files of various file types, and by sharding individual files into fragments, to protect the original data to a greater extent. Upon each run, Organisation B and Organisation C were both able to access the data, while Eve was unable to learn anything about the data transaction.

RIDE-2 Classical

Organisation A proceeds to deploy a smart contract for the document, which includes a `mapping` named `whitelist` which maps the addresses to a boolean value that indicates whether that address is entitled to obtain the document. Organisation A invokes the `update` contract to legitimise Organisation B's address (and any additional addresses they approve of) within `whitelist`.

Organisation A derives a random 256-bit key k_1 to compute the ciphertext $\mathcal{C}$ from the plaintext $\mathcal{P}$ by utilising the GCM of the AES. Organisation A appends the metadata of $\mathcal{P}$ to $\mathcal{C}$, and uploads $\mathcal{C}$ to the IPFS.

Organisation A proceeds to derive the shared secrets x_B and x_C between their and Organisation B and their and Organisation C respectively, identical to that of the RIDE-1C. they derives Organisation B and Organisation C's original ECC public keys via their Ethereum address and the signature of the transaction to the smart contract. Following this, Organisation A passes this shared values through the HKDF to derive respective 256-bit keys k_{B2} and k_{C2} for the GCM of the AES. The GCM takes k_1, the respective k_2 keys, and a uniformly-randomised initialisation vector to compute the ciphered key k and prepares the information for secure transit. Organisation A issues a permit to Organisation B and Organisation C, which provides them with k; from this, they derive the k_1 by using their shared key k_2.

Utilising a computer with an Intel i7-6700k processor, this sequence of events takes an average of 4406*ms*. (The sequence ran 100 times.) As per the RIDE-1C, it is essential to note that the transactions on this private network are instantaneous, as the mining parameters are modified to create new blocks for each transaction. As with the RIDE-1C, the RIDE-2 Classical (RIDE-2C) was tested against a series of files, and fragmented files. Each test performed successfully.

5.3.3 RIDE Quantum Approach

The tests performed against the RIDE-Q implementations were identical to those performed against the RIDE-C implementations in Section 5.3.2. The obvious difference is that the keypairs were compatible with the Supersingular Isogeny Cryptography (SIC) schemes used in the RIDE-Q, and that public keys were explicitly declared in all blockchain transactions instead of the users retrieving keys through transaction data. Utilising a computer with an Intel i7-6700k processor, the sequence of events takes an average of 5755*ms* for the RIDE-1 Quantum (RIDE-1Q), and 5979*ms* for the RIDE-1Q. (The sequence ran 100 times.)

5.4 Discussion

This section discusses the implementations and results of the Rail Industry Data Environment (RIDE). Section 5.4.1 discusses the differences between the various implementations of the RIDE, and which protocol is superior. Section 5.4.2 discusses the impact of the modified Supersingular Isogeny Digital Signature Algorithm (SIDSA), and the benefit of using post-quantum cryptography in the model. Finally, Section 5.4.3 discusses the impact this research will have in the field.

5.4.1 Data Sharing Protocols

There are several advantages and disadvantages to the RIDE. Furthermore, each approach to the RIDE shares numerous properties that also hold advantages and disadvantages.

Advantages

The advantage of the RIDE One-Key Approach (RIDE-1) is that the process only involves one step of encryption when sharing data with a recipient. However, this leads to one of the key disadvantages of the RIDE-1; the fact that there is one shared symmetric key per pair of keypairs. Should Alice wish to share a piece of data with n recipients, she will have to establish n shared symmetric keys with and encrypt the data n times. Establishing n shared symmetric keys is a computationally demanding process. Furthermore, this requires extra spatial considerations of the distributed web, as Alice must upload n versions of the ciphertext to the InterPlanetary File System (IPFS).

The advantage the RIDE Two-Key Approach (RIDE-2) introduces is that each piece of data uses a unique randomly generated symmetric key which encrypts the plaintext before any requests. Suppose Alice and Bob agree to share data. In that case, they establish a shared symmetric key to encrypt the random symmetric key, maintaining one shared secret between the users, while protecting the symmetric key used to encrypt the data.

By using a blockchain platform, the Elliptic Curve Cryptography (ECC) key-pairs can be used in two ways; shared symmetric key generation for encryption of data, and digital signatures to ensure provenance and identity. Double usage provides a more flexible method of operation than the ephemeral Elliptic Curve Diffie Hellman Key Exchange (ECDH) used in Transport Layer Security (TLS). Existing platforms using the Elliptic Curve Digital Signature Algorithm (ECDSA) and smart contracts (such as Ethereum) can operate with the RIDE-1 Classical (RIDE-1C) and the RIDE-2

Classical (RIDE-2C) implementations 'out of the box'.

Two technologies implement the RIDE used in Section 5.3; the IPFS and Ethereum. Neither technology required modification for the framework to deliver the intended result. Hypothetically, any similar technologies can be used in place to suit the needs of the users. For example, any blockchain platform that utilises ECC and supports smart contracts can work in place of the Ethereum platform. Furthermore, any distributed web platform can work in place of the IPFS.

The RIDE Quantum Approach (RIDE-Q) implementations provide security against post-quantum attacks. Due to the permanence of data on a public distributed network, this is an increasingly important consideration. Though quantum attacks are only hypothetical, quantum computing technology continues to make rapid progress, and may one day pose a threat to the classical fields of cryptography that are so common with the centralised web.

Disadvantages

Once data is uploaded to the IPFS, it remains accessible for as long as the network is running. Revocation and deletion of data is impossible unless all nodes on the network agreed to remove all local copies; something that is infeasible to accomplish if members of the network do not trust one another. Should a breach of the encryption occur, or the symmetric key to become public, or the Ethereum account of a recipient, it becomes possible for anybody to access the data. Fragmenting the data into smaller pieces will reduce the chances of adversaries decrypting entire documents; however, as established in Section 5.3, this requires extra computational resources. Nonetheless, the tradeoff between this extra time required against the extra security provided is small; the importance of security far outweighs the inconvenience of a small amount of time.

Another disadvantage of this framework is the requirement of offline computation to perform encryption tasks. Due to the nature of the blockchain, all processes performed within a Solidity smart contract are visible to anybody who wishes to see them; even

private variables are accessible. For security purposes, no sensitive data should enter into a smart contract, i.e. an unencrypted symmetric key. This is not necessarily a problem; however, future research should look into providing a fully distributed system that does not require any offline computation, preventing any downtime.

5.4.2 Quantum Resistant Approach and Security

Galbraith et al. [54] provides proofs of the security of Supersingular Isogeny Cryptography (SIC). It is important to note that the protocols are based on two problems that are intractable for quantum computers; the Computational Supersingular Isogeny (CSSI) problem, and the Decisional Supersingular Product (DSSP) problem [72, 54]. The SIDSA follows the security proofs presented by Yoo et al. [151], Iovino and Visconti [70], Galbraith et al. [55].

Theorem 5.1. *The proposed Supersingular Isogeny Digital Signature Algorithm is still valid by replacing the field elements of the Zero Knowledge Proof (ZKP) round curves with their hashed j-invariants.*

Proof. Curves that are isomorphic to one another within the same field share the same j-invariant. The supersingular curves per round of the ZKP protocol are isomorphic, and thus satisfy the condition that the j-invariants are equivalent. This follows the security of the Supersingular Isogeny Diffie Hellman Key Exchange (SIDH) presented by Jao and De Feo [72]. Furthermore, this proposed SIDSA changes the commitments during the Sign algorithm. This information is already in the public domain; as such, the SIDSA follows the security proofs presented in Theorems 4 and 5 in Yoo et al. [151]. □

Tables 5.5 and 5.6 illustrate the spatial and timing performances of isogeny-based Digital Signature Schemes (DSSs) against other other post-quantum DSSs: hash-based DSSs proposed by Bernstein et al. [12], Chalkias et al. [27], van der Linde et al. [145]; a code-based DSS proposed by Courtois et al. [37]; a lattice-based DSS proposed by Ducas et al. [52]; and a ring-based DSS proposed by Barreto et al. [6].

Table 5.5: Key sizes and signature sizes of other post-quantum DSSs.

Scheme		Public Key	Private Key	Signature
Hash		1,056	1,088	41,000
Code		192,192	1,400,288	370
Lattice		7,168	2,048	5,120
Ring-LWE		7,168	4,608	3,488
Isogeny	Original	336	48	49,184
	Proposed	336	48	40,992

Table 5.6: Clock cycles of signing and verifying algorithms of other post-quantum DSSs.

Scheme		Sign	Verify
Hash		51,636,372	1,451,004
Code		20,000,000,000	1,000,000,000
Lattice		419,750	103,030
Ring-LWE		113,712	80,760
Isogeny	Original	6,800,988,576	5,032,731,546
	Proposed	6,804,096,355	5,035,031,303
	Isolated	12,121,466	5,035,031,303

One of the benefits of this scheme is precomputation and isolation. The Non Interactive Zero Knowledge Proof (NIZKP) rounds compute before they combine with the intended message. It is, therefore, possible to produce precomputed batches of signature responses for use when signing messages. It is also possible to compute these batches offline when signing is not required. The NIZKP stage is designed to operate on multiple threads, and (as Tables 5.3 and 5.4 infer) this provides a significant reduction in computational time.

Isogeny based cryptography maintains the smallest public and private keys of any of the post-quantum protocols. The proposed SIDSA reduces the signature size in line with conventional hash-based signatures, though the blockchain specific SIDSAs provide reduced signature sizes and signing and verification times. However, the construction of SIC and the nature of blockchains requires users to maintain two sets of keypairs.

Furthermore, the signature sizes themselves, being approximately 41 kb for any data, may prove to be a significant overhead.

Computation of isogenies is a hard problem. Table 5.6 indicates that both signing and verification still require considerable computation resources in comparison to the other post-quantum solutions. Furthermore, introducing SIC protocols into the domain of a blockchain will require an overhaul of any previous cryptosystems in use. For example, if a platform uses classical ECC, the original keys are not translational into the SIC. Moreover, the NIZKP rounds can only be used at most once; repetition of any random scalar values to generate R is likely to provide an adversary with responses to both possible challenge bits. The repetition allows the adversary to trace back the commutative process described in Section 2.3.3. With all of these aspects taken into consideration, we deem that the RIDE-Q is not currently an adequate solution with SIC as the underlying cryptosystem, and requires extra research into alternative post-quantum cryptosystems or improvements in SIC itself.

5.4.3 Impact

The RIDE offers a solution to distributing data between stakeholders over trustless distributed networks. Distribution is essential to achieve in order to remove the financial and logistical burdens of centralised networks. One particular industry that could make use of such a framework is the railway industry in the United Kingdom. With a highly privatised railway industry, there are many competing companies which leads to siloation; the segregation of entire Information Technology (IT) systems into silos. Siloation becomes a problem when a company may be willing to share data; companies must issue bespoke Application Programming Interfaces (APIs) in order to extract the data of interest while maintaining protection over the data that it wishes to keep secret. By storing data on a distributed network, the storage of data will be shared amongst the stakeholders and will make data accessibility easier. The RIDE will allow stakeholders to choose which data to share and sell, and with whom to share and

sell it with. Nonetheless, the RIDE requires further scrutinising, especially into the technical limitations of the current post-quantum implementations and distribited web approach.

5.5 Conclusion

This chapter proposes the Rail Industry Data Environment (RIDE), a framework for sharing sensitive data on distributed networks. We justify the decision to use blockchain technology via the Model of Blockchain Suitability (MoBS) developed as part of this thesis, of which the results suggested to utilise a hybrid approach of non-blockchain technologies and private blockchain technologies. With these results in mind, we selected the InterPlanetary File System (IPFS) and Hyperledger Fabric as the distributed storage and blockchain platforms respectively.

Four implementations exist; two using classical Elliptic Curve Cryptography (ECC) (the RIDE-1 Classical (RIDE-1C) and the RIDE-2 Classical (RIDE-2C)), and two using post-quantum Supersingular Isogeny Cryptography (SIC) (the RIDE-1 Quantum (RIDE-1Q) and the RIDE-2 Quantum (RIDE-2Q)). We have initiated research into using SIC with blockchain technology; something that has not appeared in the literature beforehand. The post-quantum implementations require a suitable post-quantum blockchain with associated keypairs to use for both key exchange and digital signatures. This chapter proposes the use of SIC. It develops upon existing implementations of the Supersingular Isogeny Digital Signature Algorithm (SIDSA); using a 256-bit hash of the j-invariant during the Non Interactive Zero Knowledge Proof (NIZKP) stage of signing. For quantum security of 128 bits, this reduces the size of the signature by 16.66%. Furthermore, it only requires a 0.01% increase in bytes per operation and a 0.02% in allocations per operation for signing from the previous SIDSAs. It is clear from the results in Section 5.3 that introducing this compression has a negligible effect on run-time. Nonetheless, the signature sizes for SIDSAs are still larger than other

post-quantum Digital Signature Scheme (DSS). With this in mind, and the requirement for two sets of keypairs, the RIDE Classical Approach (RIDE-C) will be most suitable for use for the time being, until alternative post-quantum approaches are investigated.

The architecture of the RIDE enables rail industry stakeholders to share and sell data on the platform; the associated blockchain and distributed web technologies handle the transactions, auditing, and storage of this data. This will help to control and share large quantities of sensitive data between stakeholders without having to rely upon centralised databases and bespoke Application Programming Interfaces (APIs) to access the data, with fully audited transactions.

Chapter 6

Towards a Distributed Rail Ticketing Platform

This chapter introduces the System for Ticketing Ubiquity with Blockchain (STUB), a Hyperledger Fabric-based platform for hosting a digital infrastructure for rail ticketing in Great Britain (GB)[1]. The STUB enables passengers to store rail tickets on their blockchain account. We carefully plan the architecture of the STUB in order to overcome the existing barriers to digital tickets as discussed in Section 2.4.3; mainly the various incompatible methods of digital tickets, such as m-Tickets and smartcards. The STUB accomplishes this by storing the tickets on the ledger, making them accessible to all stakeholders on the network. This is so that any Train Operating Company (TOC) or Ticket Vending Organisation (TVO) can access any passenger ticket, even if they did not sell them the ticket in the first place. We present STUB tickets as the closest analogy to the paper ticket, currently the most flexible version of a ticket on the GB rail network. Furthermore, we propose the STUB as the underlying data structure to the digital infrastructure; stakeholders may retain their applications for passengers to use. Section 6.1 serves as a reminder to the current approach to rail tickets in GB, regarding both the passenger experience and the digital solutions. We identify the

[1]This work (and therefore this chapter) is strongly related to that presented in one of the author's publications [109].

need for a platform like the STUB, and establish the aims of the platform. Section 6.2 presents the methodology, which considers the requirements for STUB and discusses the development of the architecture and protocols. Section 6.3 offers results from tests simulating interactions on STUB between governing organisations, TOCs and TVOs, and passengers. Section 6.4 discusses the benefits and limitations of STUB and the potential impact within the rail industry. Section 6.5 concludes the research.

6.1 Introduction

Railways across the globe are undergoing a twenty-first-century renaissance. At the forefront of this period of revitalisation is the pursuit of digitalisation, whereby the industry is upgrading the numerous legacy systems our railways rely upon to conform to twenty-first-century expectations [95]. There are a plethora of fields in which digitalisation has benefited the railway industry, and one such topic is railway ticketing. Passengers in Great Britain (GB) have seen an increasing number of options to purchase digital tickets in the over the recent decade; Section 2.4.3 describes the variety of methods in detail. National Rail indicates that passengers in GB may download an *m-Ticket*, which is "just like the standard orange rail ticket, but it's held on your mobile device" [96]. Furthermore, as of October 2019, a selection of Train Operating Companies (TOCs) and certain vendors will permit the use of Integrated Transport Smartcard Organisation (ITSO) standard smartcards that utilise Quick Response (QR) and Near Field Communication (NFC) technologies to validate tickets. Nonetheless, these methods of purchasing and storing tickets have numerous disadvantages; not least that each method is completely isolated to one another; for example, if a passenger purchases a ticket for a smart card, it is only accessible via that smartcard in future. Section 2.4.3 identifies these issue in depth.

The System for Ticketing Ubiquity with Blockchain (STUB) is a platform designed to overcome these barriers and to encompass the relative ticketing infrastructure within the

Rail Settlement Plan (RSP), including Latest Earnings Network Nationally Overnight (LENNON), to ensure a distributed shared approach to railway ticketing while retaining the key advantages that current approaches hold. The aims of the STUB are:

1. Use a blockchain platform to distribute access to ticketing data.

2. Allow passengers to purchase rail tickets on a single platform, to store all of their tickets in one place.

3. Allow passengers to access their account and tickets through a variety of devices and applications, to allow flexibility in terms of personal restrictions and preferences.

4. Allow organisations to vend and validate tickets where necessary.

5. Allow organisations to access tickets sold by other organisations.

6. Allow organisations to retain their own applications and finances, to preserve the industry competition.

6.2 Methodology

This section describes the development of the System for Ticketing Ubiquity with Blockchain (STUB). Section 6.2.1 determines the suitability of blockchain technology for the underlying use case via the Model of Blockchain Suitability (MoBS). It also considers prerequisites for any implementation of the use case. Section 6.2.2 describes the development of platform architecture through the use of Hyperledger Fabric. Finally, Section 6.2.3 describes the development of the mechanics and protocols that operate on the architecture.

6.2.1 Suitability and Preliminary Decisions

The STUB considers several requirements concerning both passengers, and organisations (such as Train Operating Companies (TOCs) and Ticket Vending Organisations (TVOs)) alike. These requirements retain the advantages of existing methods of railway ticketing and eradicate any disadvantageous or surplus aspects.

Suitability

As per in Section 5.2.1, we looked to validate the use of blockchain technology for the hypothetical ticketing platform proposed in Section 6.1. The intention of the STUB is to allow stakeholders to dispense and verify tickets to passengers. These passengers will also engage with the network, though to a lesser extent than the main stakeholders. With these factors in mind, and taking into account the objectives discussed in Section 6.1, we provided the aptest statements into the MoBS to determine the results. Table 6.1 illustrates the scores.

By analysing the scores presented in Table 6.1, we see a clear endorsement for private blockchain technology. This is derived from a combination of the ISA and the DSA. The resultant ISA value of 100% provides a unanimous indication that a private blockchain is the best approach over non-blockchain and a public blockchain technologies. The DSA of 100% reiterates this and extends upon the result by confirming that this use case is a highly suited use case for blockchain technology in general. According to Table 4.13, this puts private blockchain at a "high level of suitability", of which we should "strongly consider using over the other categories". From the five Use Case categories presented in Section 4.2.3, this use case falls within the Business and Industry category. The supplementary suggestion table for this category (Table 4.14) provides a Transportation subcategory, and suggests the Hyperledger projects for the STUB. As such, this chapter forms a proof-of-concept using Hyperledger Fabric, one of the most established Hyperledger projects that provides several key technologies required of the

Table 6.1: The Individual Score Array (ISA) and Domain Score Array (DSA) of the response to the MoBS for the STUB use case.

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S2	0	0	1	1	-1	-1
A1P2S2	-1	-1	1	2	0	-1
A1P3S2	-1	-2	1	3	-1	-2
A1P4S2	0	-2	1	4	-1	-3
A2P1S1	-1	-3	1	5	1	-2
A2P2S2	1	-2	1	6	-1	-3
A2P3S1	-1	-3	1	7	1	-2
A3P1S2	1	-2	1	8	-1	-3
A3P2S1	1	-1	1	9	1	-2
A3P3S1	-1	-2	1	10	1	-1
Total Score	-2		10		-1	
Adjusted Score	0		10		0	
ISA (%)	0		100		0	
DSA (%)	0		100		0	

STUB.

Prerequisites

There are several prerequisites to consider before designing the architecture and protocols of the STUB. These prerequisites concern technical considerations, passengers, organisations, and other external factors.

Technical There are several considerations regarding the specificity of the tickets. First, a variety of types of ticket must exist: *advanced single* and *advanced return* journeys, whereby the tickets restrict passengers to specific trains; *open single* and *open return* journeys, restricting passengers to a specific route; and *seasonal* tickets, where the ticket is valid on a particular route for an extended period. Within these types, vendors offer various fares: *first-class* and *standard* to differentiate the class of travel taken; *peak* and *off-peak* fares, restricting travel during specific times of the day; and *discounted* tickets

due to age or the validity of an accompanying railcard. On top of this, passengers book *seat reservations* alongside advanced tickets. Thus, the STUB must:

- Handle all existing types and fares of tickets as listed above;
- Handle seat reservations;
- Protect passengers against loss or theft of identity.

Passengers Section 2.4.3 describes the experience for passengers purchasing tickets in Great Britain (GB). With these aspects taken into consideration, the passengers using the STUB must:

- Have the option of station purchase and online purchase, as to not discriminate against non-online users;
- Have the option to purchase with cash or bank card;
- Have access to tickets without requiring online access.

Organisations On privatised railway networks, it is crucial to maintain competition between organisations acting as ticket vendors. As such, passengers should be able to choose whom they purchase their ticket from (when using online purchase). TVOs should offer identical base tickets and prices, with allowances for sales. In GB, Latest Earnings Network Nationally Overnight (LENNON) captures data from the TVOs and other TOCs on a daily basis. LENNON combines this data with relevant non-LENNON data to calculate passenger usage and split the fares between the TOCs in a fair manner. As such, TVOs using the STUB must:

- Offer identical base tickets and prices, with allowances for sales;
- Have access to tickets sold by other companies, without sacrificing security and privacy of their systems.

External Factors A pressing issue of many organisations is the need to be conscious of environmental impacts. Paper tickets and plastic railcards are still in use in GB. As such, the STUB should:

- Eradicate the need for paper tickets in order to reduce waste;
- Eradicate the need for surplus plastic railcards in order to reduce waste.

6.2.2 Development of Platform Architecture

As per Section 6.2.1 and according to the MoBS, the channel and hierarchical features that the private Hyperledger platform possesses suits the use case of the STUB. With further investigation, we deem the architecture of the Hyperledger Fabric platform to be most suited. Figure 6.1 illustrates the proposed network architecture of STUB as a Hyperledger Fabric network to achieve distributed ticket data sharing while maintaining individual rights and security for participating organisations.

Firstly, one of the participating organisations must initiate the network. The most likely candidate to act as the *initiating organisation* is the organisation responsible for the governance of the railway network. (In the case of GB, this is the Rail Delivery Group (RDG).) Within Figure 6.1, the colour red represents this organisation. Upon initiation, the *network* (represented by the bold outer line in Figure 6.1) consists of three artefacts: a *network policy*, which provides the initiating organisation with administrative rights for the network in the form of a Membership Service Provider (MSP); a *certificate authority*, which dispenses digital identities to permitted users within the initiating organisation; and an *orderer* node[2] responsible for ordering future transactions into blocks before consensus.

Once the initiating organisation establishes the network, they create a *channel* to add new organisations. (Figure 6.1 illustrates this channel with a dashed line and the new organisations in yellow and blue.) These organisations do not require network

[2]There may be more than a single orderer node in a Fabric network. However, for simplicity, the STUB proof-of-concept restricts the network to just a single orderer.

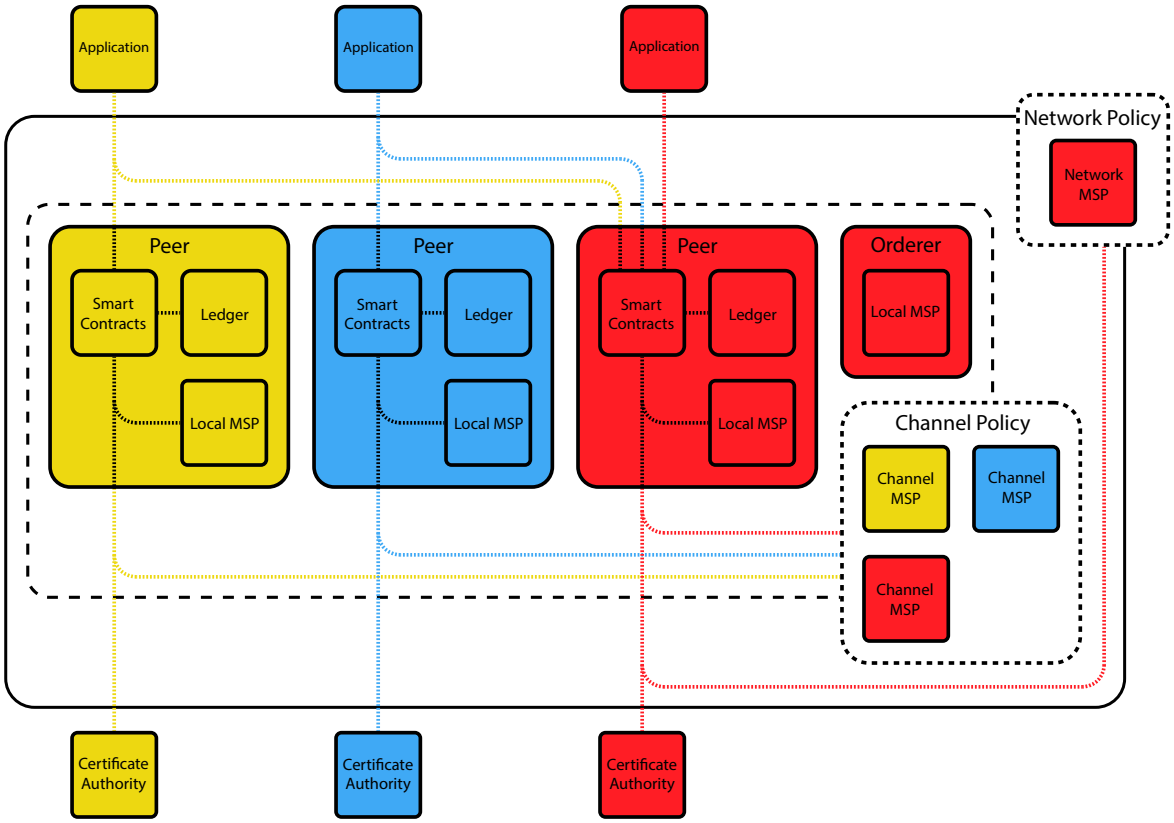


Figure 6.1: The STUB architecture. The different colours represent different organisations.

administrative privileges. Hence they exist on the channel instead. Channels are sub-networks that possess self-contained ledgers and smart contracts for members of the channel to share between one another. Upon creation, the initiating organisation establishes a channel policy, granting MSPs to each organisation participating in the channel permit shared channel administrative privileges. Furthermore, the initiating organisation forms a unique sub-organisation to maintain the identities of passengers seeking to use the platform. This sub-organisation allows organisations on the channel to share the passenger identities, extirpating the need for passengers to create a separate identity for each organisation with whom they interact.

The other organisations host *peer* nodes on the channel to store a copy of the channel ledger, execute smart contracts, and provide consent to transactions initiated by other organisations on the channel. Furthermore, these organisations provide applications

to allow passengers to purchase tickets. The organisations construct the applications around the various methods identified in Section 2.4.3; for example, mobile applications, websites, or fixed station ticket terminals. Applications are responsible for verifying the identity of a passenger before requesting a transaction to issue a ticket (as a digital asset on the channel ledger).

6.2.3 Development of Platform Protocols and Mechanics

Figure 6.2 illustrates the platform mechanics that operate on the architecture of Figure 6.1. Any passenger who wishes to purchase a ticket must first create a platform identity. The STUB stores the identity as an X.509 digital certificate which contains appropriate information about the passenger, such as their age (to determine permissible fares) and any railcards and other discounts the passenger is subject. The platform restricts the creation of an identity to a smart contract built and stored by the initiating organisation. Nonetheless, applications from the other organisations (the TVOs) have access to the passenger identities by invoking the contract via a request to the initiating organisation peer node. Figure 6.1 illustrates this. The channel MSP of the initiating organisation stores the identity, thus it is accessible to all organisations on the channel. Storing identities on the channel results in organisations using the same identities, preventing data duplication and redundancy.

The STUB entitles passengers with digital identities to purchase tickets. As the Channel Policy maintains passenger identities, the passenger may use an application provided by any organisation on the channel. The passenger submits a request of the tickets they wish to purchase (along with a valid signature corresponding to their digital certificate) to the application, which proceeds to validate this request via a smart contract stored within the peer node of the dispensing organisation. The channel's distributed ledger records the ticket as a transaction, whereby the vendor's identity transfers the underlying asset (the ticket) to the passenger's identity. The transaction contains the necessary information about the ticket. The initiating organisation permits

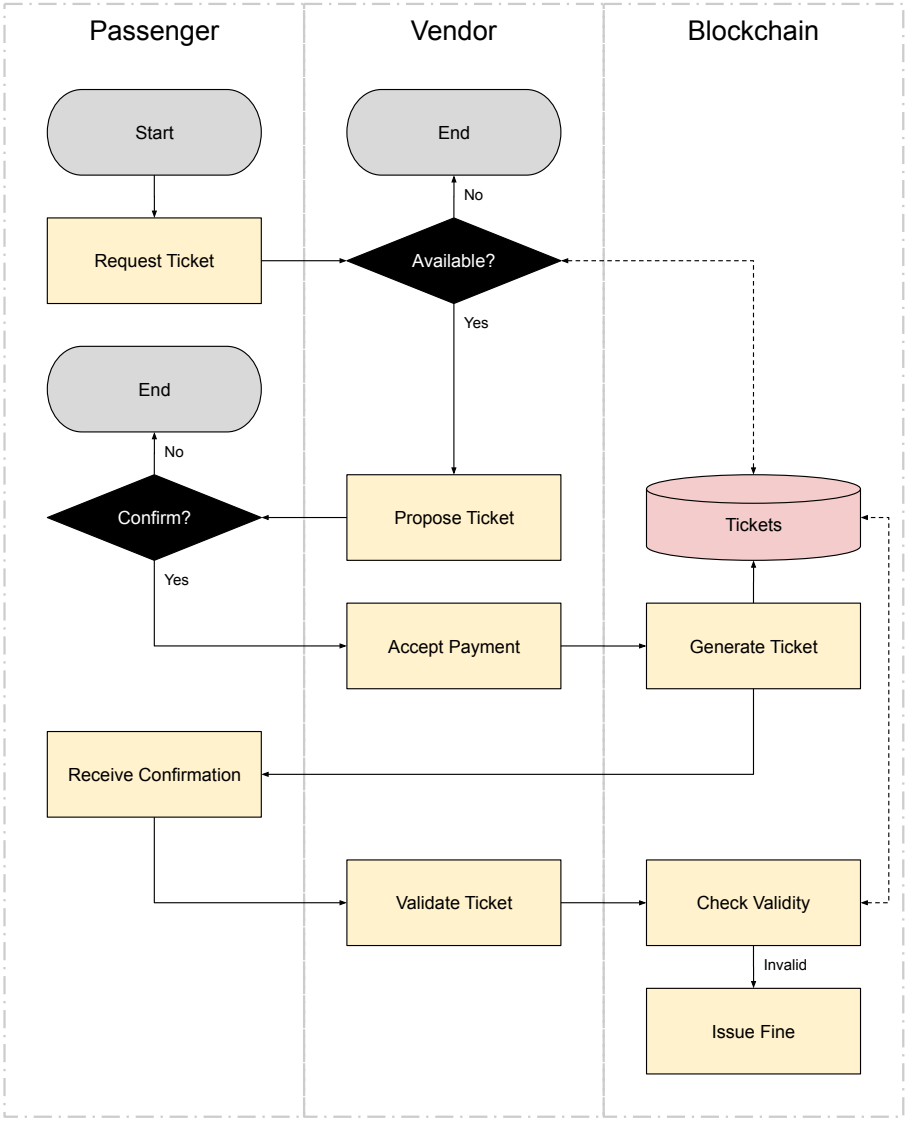


Figure 6.2: The process flow of the platform.

specific channel organisations to validate tickets via consensus. Once more, it is the role of an application to obtains passenger credentials and query the ledger via a smart contract to establish whether a valid ticket exists for the passenger at that particular time.

6.3 Results

In order to test the capabilities of the System for Ticketing Ubiquity with Blockchain (STUB), we performed a series of tests based on case studies concerning various users.

The perspectives of these users are crucial to forming these case studies: Section 6.3.1 looks at the organisation responsible for the governance of the platform; Section 6.3.2 looks at the organisations seeking to vend and verify tickets; and Section 6.3.3 looks at the passengers seeking to purchase a ticket. We conducted the tests in the form of mock testing, whereby mock data (historical ticket transactions provided by the Rail Delivery Group (RDG)) simulated the inputs and outputs required. Each test passed successfully. Note that the order provided in this section does not reflect the order of tests conducted; instead, they are grouped by the users to provide clarity. Each test has a number to indicate the chronology of the tests.

6.3.1 Administrative Organisations

There are two significant administrative organisations required for the platform; an initiating organisation to set the platform rules and determine governance, and an organisation to analyse ticketing statistics and to oversee the Rail Settlement Plan (RSP). For simplicity, the following tests combine these responsibilities into one organisation. However, it is feasible to split these organisations into two separate organisations.

Test 1 Organisation R (denoted by red in Figure 6.1) is responsible for initiating the network and adding organisations to the channel. The first test for Organisation R is as follows:

1. Network Initiation

    (a) Organisation R initiates the network, providing a certificate authority, an orderer node, and a network Membership Service Provider (MSP).

2. Channel Initiation

    (a) Organisation R initiates a channel within the network, providing a peer node and a channel MSP.

(b) Organisation R adds Organisation B and Organisation Y to the channel, defining their responsibilities.

3. Add peer Nodes

(a) Organisation R provides smart contracts on their peer node to allow applications to add passenger identities to the channel.

Test 4 Further to their initiating responsibilities, Organisation R requires access to the entire database to provide a statistical overview of the ticketing data. Moreover, access allows Organisation R to evaluate the fares of the passengers and distribute the profit between the stakeholders as per the agreement of Latest Earnings Network Nationally Overnight (LENNON). The second test for Organisation R is as follows:

1. Network Initiation

(a) Organisation R initiates the network, providing a certificate authority, an orderer node, and a network MSP.

2. Channel Initiation

(a) Organisation R initiates a channel within the network, providing a peer node and a channel MSP.

(b) Organisation R adds Organisation B and Organisation Y to the channel, defining their responsibilities.

3. Add peer Nodes

(a) Organisation R provides smart contracts on their peer node to allow applications to add passenger identities to the channel.

(b) Organisation B adds one peer node to the channel.

(c) Organisation Y adds one peer node to the channel.

4. Issue Applications

(a) Organisation B issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.

(b) Organisation Y issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.

5. Dispense Tickets

   (a) Organisation B dispenses tickets to passengers.

   (b) Organisation Y dispenses tickets to passengers.

6. Transaction Consensus

   (a) Organisation B provides consensus for other transactions on the channel.

7. Ticket Validation

   (a) Organisation Y validates the passenger tickets.

8. Data Extraction

   (a) Organisation R gathers this data in a suitable format for further analysis.

This test uses suitable passenger data, provided by the RDG.

6.3.2 Ticket Vending Organisations

The researcher creates two hypothetical Ticket Vending Organisations (TVOs) to test the platform against the passenger considerations of Section 6.2.1.

Test 2 Organisation B's (denoted by blue in Figure 6.1) sole responsibility is selling tickets to passengers. To achieve this, Organisation B must host one or more peers on the channel for two reasons; to consent to other transactions within the channel, and to

issue new tickets to passengers using their application. Thus, the test for Organisation B is as follows:

1. Network Initiation
    (a) Organisation R initiates the network, providing a certificate authority, an orderer node, and a network MSP.
2. Channel Initiation
    (a) Organisation R initiates a channel within the network, providing a peer node and a channel MSP.
    (b) Organisation R adds Organisation B and Organisation Y to the channel, defining their responsibilities.
3. Add peer Nodes
    (a) Organisation R provides smart contracts on their peer node to allow applications to add passenger identities to the channel.
    (b) Organisation B adds one peer node to the channel.
    (c) Organisation Y adds one peer node to the channel.
4. Issue Applications
    (a) Organisation B issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.
    (b) Organisation Y issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.
5. Dispense Tickets
    (a) Organisation B dispenses tickets to passengers.

(b) Organisation Y dispenses tickets to passengers.

6. Transaction Consensus

   (a) Organisation B provides consensus for other transactions on the channel.

Test 3 Organisation Y (denoted by yellow in Figure 6.1) is also a Train Operating Company (TOC), and is therefore responsible for both issuing and validating tickets. Validation occurs upon request. Aside from this additional responsibility, the test is identical to that of Organisation B's.

1. Network Initiation

   (a) Organisation R initiates the network, providing a certificate authority, an orderer node, and a network MSP.

2. Channel Initiation

   (a) Organisation R initiates a channel within the network, providing a peer node and a channel MSP.

   (b) Organisation R adds Organisation B and Organisation Y to the channel, defining their responsibilities.

3. Add peer Nodes

   (a) Organisation R provides smart contracts on their peer node to allow applications to add passenger identities to the channel.

   (b) Organisation B adds one peer node to the channel.

   (c) Organisation Y adds one peer node to the channel.

4. Issue Applications

   (a) Organisation B issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.

(b) Organisation Y issues an application for passengers to purchase tickets. This application will connect to the channel to identify the passengers from the passenger organisation.

5. Dispense Tickets

   (a) Organisation B dispenses tickets to passengers.

   (b) Organisation Y dispenses tickets to passengers.

6. Transaction Consensus

   (a) Organisation B provides consensus for other transactions on the channel.

7. Ticket Validation

   (a) Organisation Y validates the passenger tickets.

6.3.3 Passengers

Three hypothetical passengers (henceforth named Alice, Bob, and Charlie) exist to test the platform against the passenger considerations of Section 6.2.1. For each test, assume that a STUB network is fully running as per the previous tests.

Test 5 Alice is twenty-four years of age and thus qualifies for the Young Persons Railcard, which offers a discount of 25% on the usual fare. She uses Organisation B's application on her mobile phone to create her identity, procure her railcard, and purchase an open return, standard-class ticket. The test for Alice is as follows:

1. Alice creates an identity and purchases a Young Persons Railcard with Organisation B's application.

2. Alice purchases an open return ticket from Organisation B's mobile application.

3. Alice validates the ticket with Organisation Y.

This tests against the feature of built-in railcards, removing the separation between the ticket and the railcard. Furthermore, the test proves the ability of two competing companies to interact with the same underlying ticket.

Test 6 Bob is under sixteen years-of-age, thus qualifying for a child's fare. Bob sets up his account online at home under the supervision of his parents. Using their accounts, they send Bob an advance single ticket with corresponding seat reservations via the website provided by Organisation Y. The test for Bob is as follows:

1. Bob's parents create an account.
2. Bob creates an account, providing his age.
3. Bob's parents purchase an advance single ticket from Organisation Y using their credentials.
4. Bob's parents send him the ticket via the application provided by Organisation Y.
5. Bob validates the ticket with Organisation Y.

This test checks the automatic qualification of cheaper fares and the transfer of tickets between passenger accounts. Once again, the underlying ticket is accessible by competing companies.

Test 7 Charlie creates an account and purchases a season ticket using station facilities. He loses his card and wishes to cancel the digital identity to prevent anybody who finds his card from using it. The test for Charlie is as follows:

1. Charlie creates an account at the station.
2. Charlie purchases a season ticket from Organisation Y using cash at a station.
3. Charlie validates the ticket with Organisation Y.

4. Charlie loses the physical card provided and cancels the identity using Organisation B's application.

This test analyses the ability to create seasonal tickets that require validation between an extensive range of dates. Moreover, it tests the revocation of identity should that be the passenger's desire.

6.4 Discussion

The architecture and the mechanics of the System for Ticketing Ubiquity with Blockchain (STUB) combine the requirements of railway ticketing with state-of-the-art technologies. This section discusses how the symbiotic development of the STUB results in several beneficial aspects. Section 6.4.1 discusses the benefits of the STUB, whereas Section 6.4.2 discusses the limitations. Finally, Section 6.4.3 explains the impact the STUB will have on research and industry.

6.4.1 Benefits

The design of the STUB on Hyperledger Fabric - with channels and a separate organisation for handling passenger identities - maintains competition between Ticket Vending Organisations (TVOs). This grants TVOs with the freedom to design applications to their standards. The core of the application will connect and run with the STUB, whereas the interface is open to design. Because the applications provided are external to the network, they interact with other networks, databases, and Application Programming Interfaces (APIs). For example, external databases store timetabling data and vehicle information required for seat reservations. Applications can access the data to inform decisions before dispensing tickets.

The use of blockchain technology provides a secure, distributed platform, resulting in no centralisation of the data. This prevents the possibility of network downtime

and the redundancy of ticket duplication. If one TVO experiences any downtime, a passenger's ticket is accessible through any other TVO. This is because all peer nodes store a local copy of the distributed ledger, which shows all ticket transactions that have occurred on the network. Storing the ledger local to the peer node reduces the time required to transmit requests and responses across the network via an API. Transparent data and automated settlement contracts on the blockchain will enable full automation of the Rail Settlement Plan (RSP), enabling the overhaul of legacy systems such as Latest Earnings Network Nationally Overnight (LENNON).

Paper tickets and surplus plastic railcards become redundant with the use of the STUB. Passenger identities are cryptographic keypairs, accessible via a variety of methods such as mobile phones and reusable, multi-purpose smart cards that have access to Near Field Communication (NFC) and Quick Response (QR) technologies. Because the public identities are X.509 certificates, they can store additional information such as important personal details and any valid railcards. This wraps up a passenger's ticket and relevant discount cards into one single identity.

The STUB does not require a paradigm shift in ticketing from the perspective of a passenger. The STUB still offers passengers a variety of methods to purchase tickets, and online access is not a prerequisite for using the STUB, avoiding requiring passengers without access to the World Wide Web (WWW) to obtain access. Once a passenger has created a STUB identity, they may reuse this with any TVO participating on the STUB. Because the STUB uses a distributed ledger to store and access tickets, and passengers have multiple avenues of accessing their STUB identity, the STUB reduces points-of-failure for the passenger. For example, should a passengers phone run out of battery - rendering any phone applications inaccessible - they may use a smartcard linked with their identity instead to access the same identity and therefore the same ticket.

6.4.2 Limitations

The limitations of the STUB concern the organisations on the network. Furthermore, the existing infrastructure used in railway ticketing must adapt to support the STUB. In order to make use of the STUB's mechanics, TVOs must modify existing applications to interact with the network. As said in Section 6.4.1, there are many advantages to applications; nonetheless, adaptation will take time and effort.

Infrastructure must also adapt to support the platform. To validate any ticket, an organisation queries the latest copy of the ledger. Though possible to perform offline (from a local copy of the ledger at any peer node), the nodes require internet access to update the ledger to the newest state. Any machine that validates a ticket - whether an electronic gate or a handheld device - must have access to the internet in order to update the ledger. The STUB overcomes this by delayed validation, where it caches validation requests until an internet connection becomes available.

6.4.3 Impact

This research will promote interest in the area of digital ticketing using distributed solutions. As of the time of writing this thesis, the majority of digital ticketing takes the form of QRs on mobile applications and restrictive smartcards, both of which have limitations that might result in loss of proof-of-validity for the passenger. The STUB is the first step towards a system inclusive for all passengers while providing numerous other technological and environmental benefits. Industrial stakeholders will be keen to adopt a platform like the STUB, as it does not make their existence redundant; instead, it replaces the core of the current model for stakeholders to feed in and out.

6.5 Conclusion

This chapter has presented a novel approach to digital ticketing using blockchain technology. As per the underlying idea of thesis, we utilised the Model of Blockchain Suitability (MoBS) to cyclically justify the decision to approach the use case with blockchain technology, whilst refining and validating the MoBS itself. From the results of the MoBS, we established a different response to that of the Rail Industry Data Environment (RIDE) in Chapter 5; that private blockchain technology was by far the most appropriate technology to use. As such, we do not use the RIDE as the underlying platform, opting instead to build a new system that utilised the security and decentralisation of the technology to provide a distributed system that benefits both passengers and organisations. This demonstrates that the MoBS is capable of providing nuanced and varied results, even within the same industry.

The System for Ticketing Ubiquity with Blockchain (STUB) utilises the existing technology of Hyperledger Fabric to build shared passenger identities between organisations. Channels separate organisations into relevant groups that control the accessibility to particular tickets. Organisations host peer nodes; these nodes hold a copy of the ledger and the smart contracts to interact with the ledger. Peer nodes are responsible for endorsing transactions to add to the ledger. This design gives rise to a platform whereby participating organisations may still compete against one another to issue tickets to passengers. However, the underlying ticket data is precisely the same[3]. We note the platform does not force a paradigm shift in the approach to ticketing, as the platform preserves a range of inclusive methods of ticket purchase for the passengers. In a business context, the blockchain will enable the sharing of core data while preserving competition.

The STUB offers an exciting new approach to ticketing in the privatised landscape of Great Britain (GB) railways. The STUB stores tickets as assets on a Hyperledger Fabric

[3]This is analogous to a paper ticket purchased with one vending organisation and verified by another.

blockchain, making them accessible to all Train Operating Companies (TOCs) and Ticket Vending Organisations (TVOs) on the network in order for them to validate them. This has two significant benefits. The first is that organisations are still responsible for their own applications and management, preserving the competition and enterprising nature of the industry. The second is that passengers will finally have a direct digital equivalent to the paper ticket; something that they may purchase anywhere, use anywhere, and access in multiple locations.

Part III

Concluding Remarks and Appendices

Chapter 7

Conclusion

This chapter summarises the research conducted in this thesis. Section 7.1 responds to the research questions posed in Section 3.3. Section 7.2 draws upon the impact of the researched conducted, and discusses the impact of the thesis as a whole. Section 7.3 identifies the key avenues of research that will follow this thesis.

7.1 Response to Research Questions

We respond to the questions posed in Section 3.3.

Question 1: Provided that a significant barrier to adoption of blockchain technology is the misinterpretation of when to deploy it appropriately and that the current models in the literature have certain flaws, how can a new model become more robust and provide the users with more in-depth feedback? Chapter 2 establishes that the current flow chart models in the literature fail to provide flexibility in their decision processes. As such, Chapter 4 introduces the Model of Blockchain Suitability (MoBS); a brand new model to aid the decision process. This model builds on the previous models in terms of the topics of the questions but provides the user with quantitative feedback to enable a more in-depth analysis of which particular technology the user requires. Furthermore, the model identifies real platforms to suggest to the users.

Question 2: Given that data siloation is a common occurrence in the rail industry, can blockchain technology overcome the underlying issues of data centralisation and the lack of trust between stakeholders? We establish the concepts of the Rail Industry Data Environment (RIDE) in Chapter 5, stating that using a blockchain has the potential to distribute the trust amongst stakeholders and enable automated marketisation of rail industry data.

Question 3: Provided that blockchains themselves are not suitable for native storage of large quantities of data and that current literature suggests secure, centralised storage or insecure distributed storage as a solution, how can the auditing and distributed properties of blockchain technology integrate with a secure distributed storage solution? Because the of the large quantities of data in the rail industry, Chapter 5 investigates a new architecture called the RIDE that symbiotically links blockchain and distributed web technologies to provide a platform that stores this sensitive data securely. Using a distributed web in this context has both advantages and disadvantages. It means that the data is available at all times. However, data revocation remains an open problem with distributed web technologies, as it requires the consensus of the entire network.

Question 4: If such a platform exists as an answer to Question 2, which post-quantum cryptosystem is a suitable replacement for the classical cryptosystems commonly used in blockchain technology? Traditional platforms use Elliptic Curve Cryptography (ECC) due to the benefit of small key sizes in a restricted size domain. As stated in Chapter 2, Casino et al. [25] imply that lattice and code-based post-quantum systems will be suitable. Furthermore, researchers have conducted work into blockchains with hash-based signatures. However, Supersingular Isogeny Cryptography (SIC) - which has some of the smallest key sizes in post-quantum cryptography for similar levels of security - had not been thoroughly investigated before this thesis. Thus, Chapter 5 investigates SIC as "natural successor" to ECC, as both are based on the mathematics of

elliptic curves and use small key sizes. We establish that SIC is appropriate for the RIDE as it has both a Digital Signature Scheme (DSS) and a Key Exchange Scheme (KES); however, we establish that SIC currently requires two sets of keypairs to operate. Thus, we intend that future research be conducted into alternative post-quantum solutions to the RIDE.

Question 5: Provided that the current data-driven approach to ticketing is fragmented and has numerous flaws, can the distributed mechanics of blockchain technology provide a solution? Chapter 6 designs a blockchain-based architecture to support distributed rail ticketing in Great Britain (GB), that enables competing Ticket Vending Organisations (TVOs) and Train Operating Companies (TOCs) to share the underlying ticket data while maintaining their own applications and frameworks. Each passenger using the System for Ticketing Ubiquity with Blockchain (STUB) has their own identity, which is possible to access on multiple devices to improve the flexibility of digital tickets.

Question 6: How can such a system help in overhauling the Rail Settlement Plan (RSP)? With the fundamental ticketing data open on the ledger, real-time statistical analysis is possible. Furthermore, we introduce the concept of automated settlements via smart contracts on the blockchain, overhauling the legacy systems currently used in the RSP.

Question 7: Assuming answers for all previous questions, will blockchain technology disrupt the rail industry as it has with other industries? We believe this research shall pave the way towards the adoption of blockchain technology in the rail industry, and in overhauling the Digital Rail Framework (DRF). Though future work and industrial scrutiny are required to develop the concepts into industrial standards, this thesis provides the underlying ideas to solve data issues that have plagued the rail industry for years.

7.2 Impact

This thesis has taken steps towards improving upon previous blockchain decision frameworks. First identified in Section 2.2.2, we established both the advantages and disadvantages of prior models of Lo et al. [84], Wust and Gervais [150], and Maull et al. [86]. We use these factors as a springboard to build the MoBS in Chapter 4. The philosophy of the MoBS was to bring in improved quantitative metrics and new qualitative suggestions to enhance the user feedback, whilst maintaining the intuitive nature that the previous models achieved. The MoBS is developed in distinct stages, with this core philosophy shaping the framework each step of the way.

We use a combination of simple tests (conducted within Chapter 4) and two larger, rail-related use cases to validate the MoBS. These use cases were extracted from reviewing the literature in Section 2.4, establishing flawed or missing data platforms within the industry. From this, there were two key areas of investigation; stakeholder data sharing, and ticketing, addressed by Chapters 5 and 6 respectively. These chapters have held two significant purposes; to validate the work conducted in Chapter 4, and to establish viable solutions to the issues presented. The research conducted throughout this thesis presents exciting novel solutions to existing issues.

7.3 Future Research

As industrial and academic research into blockchain technology grows, so to will the MoBS. The MoBS is designed with expansion and flexibility in mind and will improve as we discover new ideas within the field. As part of future research, we would like to see the MoBS validated with real blockchain developers, just as in the model of Maull et al. [86]. Despite the numerous advantages the MoBS introduces over older decision models, there are opportunities for future research to expand upon the research conducted in this thesis. As mentioned in Section 4.4.2, there are three areas of potential

improvement: the exclusion of *infeasible responses* to streamline the model; greater use of the attribute tags to provide a higher level of granularity; and the introduction of real implementation suggestions for responses that select databases as the most suitable platform. Though beyond the scope of this thesis, these topics will form the basis of future research projects and will supplement the general development of the MoBS as it evolves with the field of blockchain technology. We also see a validation of the MoBS by developers of blockchain platforms, as per the model of Maull et al. [86].

The RIDE is a step towards moving technical data away from a centralised context is theorised and implemented. There are several areas of research identified from the results of the RIDE. One area of concern is the computation of symmetric keys, and whether this can shift into a distributed context. This would alleviate the need to compute the symmetric keys offline, resulting in a framework where users can request and access data with no need for the data creator to be online. This research will look into modifications of the virtual machines and script languages of the blockchain platforms in use. Furthermore, researchers should look into how such a framework can integrate with data processing. As of the writing of this thesis, the framework can store sensitive documents in a public domain, but all other manipulation of the documents must perform offline. We expect to conduct an industrial review as part of future research, determining the interest from stakeholders and the specifications and standardisation of data.

Moreover, there are limitations to the post-quantum approaches of the RIDE; namely, the need for two sets of keypairs. Future research will look into combining these keypairs into one single account. Furthermore, research will investigate how to process these large datasets; as of the writing of this thesis, the RIDE benefits data *storage*. Projects are beginning to research how best to process data on the blockchain. As such, they may conglomerate the protocols from the RIDE into their designs. In the future, there will be additional research into improving the computation of the isogenies to reduce the time required for the signing and verification. De Feo and Galbraith [43] in-

troduces an isogeny-based Supersingular Isogeny Digital Signature Algorithm (SIDSA) by the name of SeaSign that utilises class group actions. Signature sizes are smaller than those presented in this thesis, though signing and verification currently take much longer than the current SIDSAs offer. Should this scheme see improvements, it may become applicable to argue its place in the blockchain context. In addition, we anticipate research into consensual revocation of data from distributed web technologies.

The research of Chapter 6 will kindle the process of overhauling the RSP. In the future, we anticipate the use of the TrustOS framework to replace the channel-wide organisation for sharing passenger identities. We also anticipate a localised industrial demonstrator to validate this approach to ticketing. The University of Birmingham are considering funding applications to build such a demonstrator for the city of Birmingham, to bring an interoperable ticketing system between the various types of transport on offer in the city.

Appendix A

Chapter 2 Supplements

A.1 Advanced Encryption Standard

The Advanced Encryption Standard (AES) is a symmetric key algorithm specified as an encryption standard by National Institute of Standards and Technology (NIST). Proposed by Daemen and Rijmen [41] in 1998, AES uses a 128 bit length block cipher with keys of 128 bits, 192 bits, or 256 bits. A substitution-permutation network performs over ten, twelve, or fourteen rounds determined by the size of the key used. The AES utilises column-major arrays A representative of a number of bytes depending on the key size. There is a state array a that represents the sixteen bytes within the block which has Nb columns, and a key array represented with Nk columns. Table A.1 provides the parameters for Nb, Nk, and the number of associated rounds Nr for each key size. For

Table A.1: Parameters of number of columns for the state array Nb, number of columns for the key array Nk, and number of rounds Nr

	Parameters		
Key Size (bits)	Nb	Nk	Nr
128	4	4	10
192	4	6	12
256	4	8	14

example, the representation of a block is an array a of sixteen bits $a_0, ..., a_{15}$ such that

$$a = \begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix}. \tag{A.1}$$

AES combines five substeps to achieve the full algorithm: `KeyExpansion`; `AddRoundKey`; `SubBytes`; `ShiftRows`; and `MixColumns`.

`KeyExpansion` expands the original 128-bit, 192-bit, or 256-bit cipher key. Because the AES uses a 128-bit block and runs over ten, twelve, or fourteen rounds, the amount of key material needed is $128 \times 11 = 1408$-bits, $128 \times 13 = 1664$-bits, or $128 \times 15 = 1920$-bits respectively. Thus, the original key requires expansion to prevent reuse and potential attacks. Assume a 128-bit key k represented by a $Nk = 4$ column array, where the columns (bytes) are $k_0, ..., k_3$. This key must expand to 1408-bits, which will result in expansion to several columns equal to $Nr \times Nk$. For k, this means an expansion such that there are forty-four columns $k_0, ..., k43$. The AES derives the round keys by splitting the expanded key into Nr separate round keys. Table A.2 demonstrates this.

Table A.2: Illustration of how a 128-bit key k expands and splits into eleven round keys [39].

Expanded Key	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	...	k_{40}	k_{41}	k_{42}	k_{43}
Round Key	0				1				...	10			

`AddRoundKey` combines the state array a with subkey k derived from the main key using a key schedule. Both a and k are 4x4 column-major arrays. The round constructs a new state b such that

$$b_{i,j} = a_{i,j} \oplus k_{i,j}. \tag{A.2}$$

Figure A.2 demonstrates this.

`SubBytes` forms a new state b by

$$b_{i,j} = S(a_{i,j}), \tag{A.3}$$

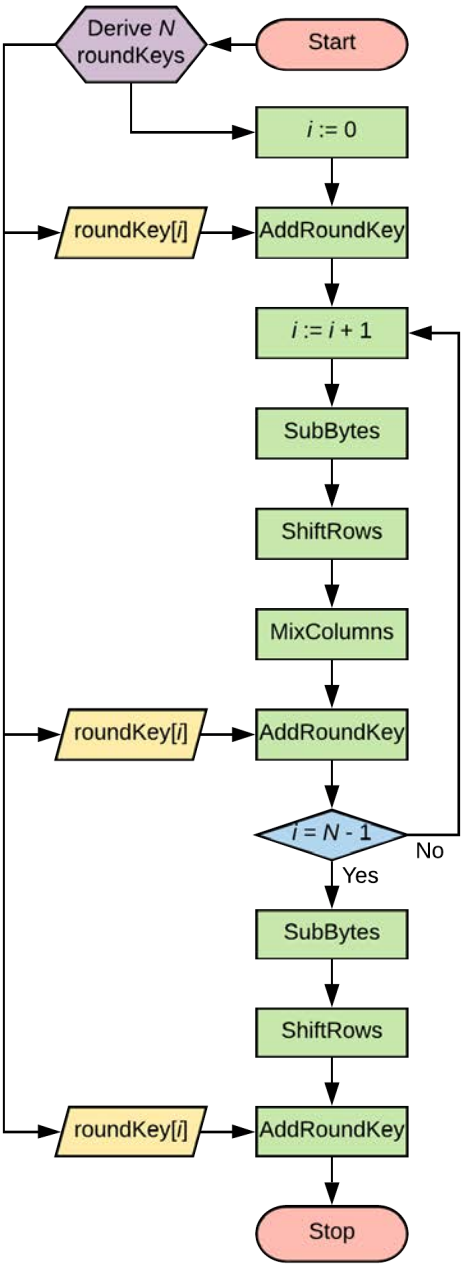


Figure A.1: The process of the AES.

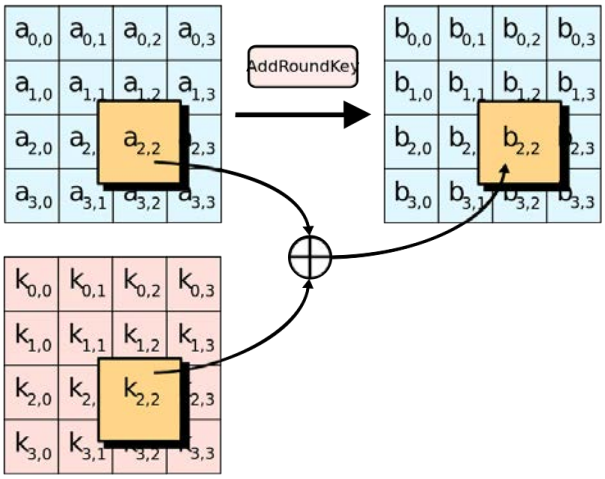


Figure A.2: An illustration of the `AddRoundKey` step of AES [39].

where $S(c)$ is a substitution box operative that maps an 8-bit input a to an 8-bit output b. The round maps he input a to its multiplicative inverse α within the Galois field GF(2^8) employing the modulo polynomial

$$p(x) = x^8 + x^4 + x^3 + x + 1. \tag{A.4}$$

An affine transformation determines the output b such that

$$b = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{A.5}$$

This round uses this particular substitution such that $S(a) \neq a$ and $S(a) \oplus c \neq \text{FF}_{16}$, and provides the cipher with non-linearity. Figure A.3 illustrates this process.

`ShiftRows` shifts certain bytes by an offset within their row. Row zero remains unchanged. Row one shifts each byte one place to the left, with the leftmost byte cycling

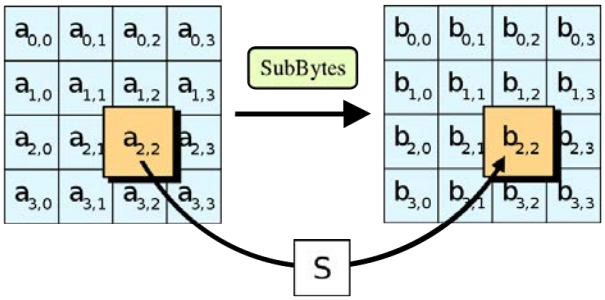


Figure A.3: An illustration of the `SubBytes` step of AES [39].

around to become the rightmost byte. Row two switches the bytes in even columns with one another and repeats this for the odd columns. Finally, row three shifts each byte one place to the right in the cyclical fashion used by row one. Figure A.4 illustrates these manipulations. The purpose of this step is to avoid encryption of each independent

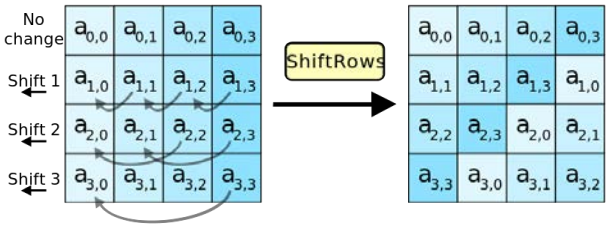


Figure A.4: An illustration of the `ShiftRows` step of AES [39].

column, preventing the case where the algorithm becomes four independent block ciphers.

`MixColumns` takes each column and represents it as a polynomial

$$p(x) = b_3x^3 + b_2x^2 + b_1x + b_0 \tag{A.6}$$

over GF(2^8), and then multiples it modulo $x^4 + 1$ with a fixed polynomial

$$a(x) = 3x^3 + x^2 + x + 2. \tag{A.7}$$

The purpose of this step is identical to the purpose of `ShiftRows`; to provide cryptographic diffusion. Figure A.5 illustrates this. The AES initiates by deriving several round keys from the main key; ten, twelve, or fourteen round keys for the main key of 128-bits, 192-bits, or 256-bits respectively.

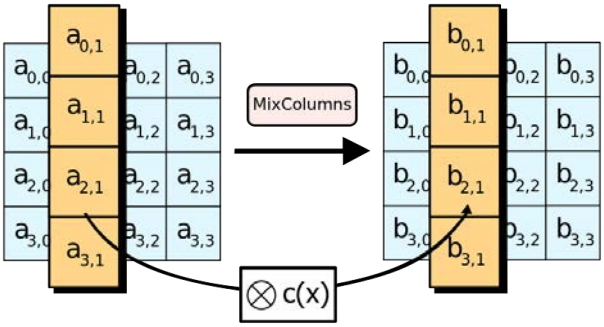


Figure A.5: An illustration of the `MixColumns` step of AES [39].

The AES has a strong performance in regards to both software and hardware. Processing speed is high, and RAM requirements are low and were suitable for the computers of 1999 when NIST made their selection; on an Intel Pentium Pro of 200MHz, the encryption process required 18 cpb resulting in a throughput of 11 MB/s.

The keyspace of the AES for an n-bit key is 2^n, meaning a brute force attack requires a time complexity of $O(2^n)$. Nonetheless, there have been several attacks against the AES published, though none are feasible with current computing power. In 2011, Bogdanov et al. [16] published a key-recovery attack using a biclique attack method, resulting in maximal operations of $O(2^{126.2})$, $O(2^{190.2})$, and $O(2^{254.6})$ for 128-bit, 192-bit, and 256-bit keys respectively. Despite this reduction by a factor of four, and contributions made by Tao and Wu [138] to improve the efficiency to $O(2^{126.0})$, $O(2^{189.9})$, and $O(2^{254.3})$, these remain theoretical.

Gilbert and Peyrin [57] identify a known-key distinguishing attack that uses a reduced eight-round implementation of the AES, by combining by consecutive rounds as one and using a "super SBox", an increased SBox as described previously in this section. The attack has a complexity of $O(2^{48})$; this is infeasible, and only works on the reduced version of the AES.

Since the only attacks against the AES are theoretical, and the fact that NIST selected it as the national standard due to its security and efficiency, the AES remains the most popular choice of symmetric cryptography algorithm to this day. There are no post-quantum attacks that exist to break the AES, which likely means a prosperous future

for the algorithm.

A.2 Rail Data Process Maps

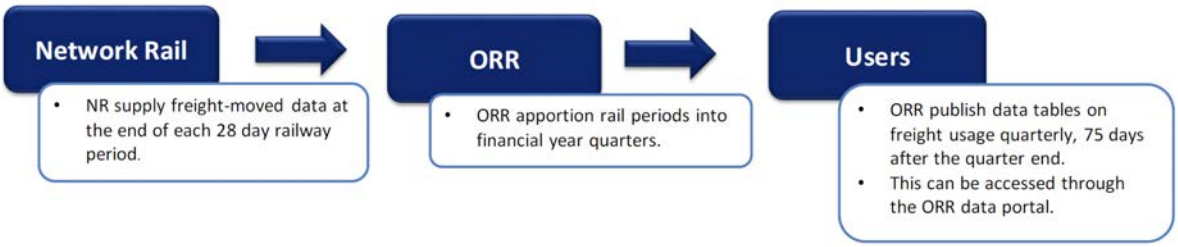


Figure A.6: Process map showing the collection of freight rail transport administrative data [114].

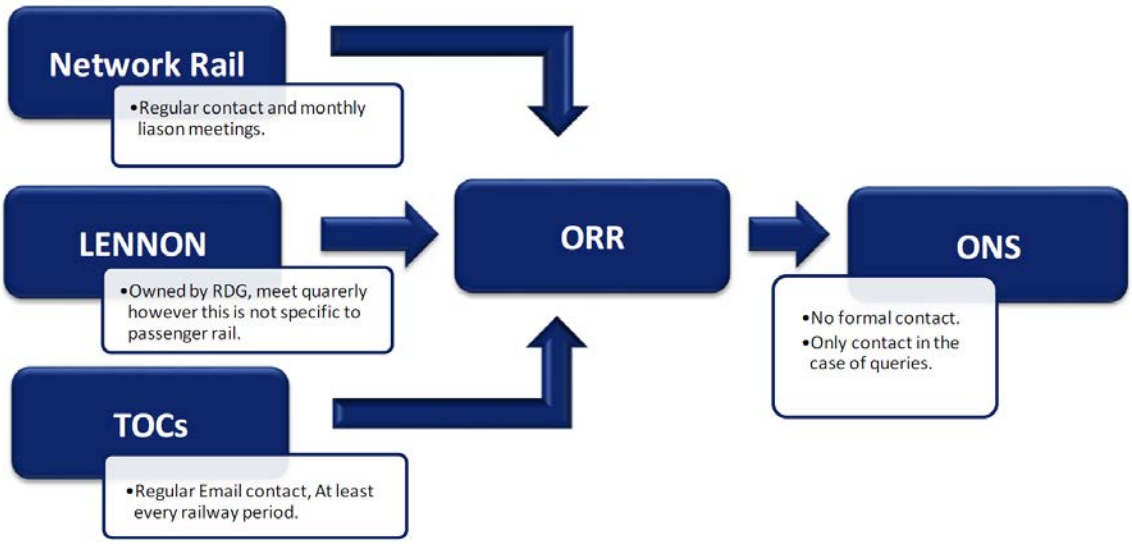


Figure A.7: Process map showing the flow of communication between data suppliers to output producers [114].

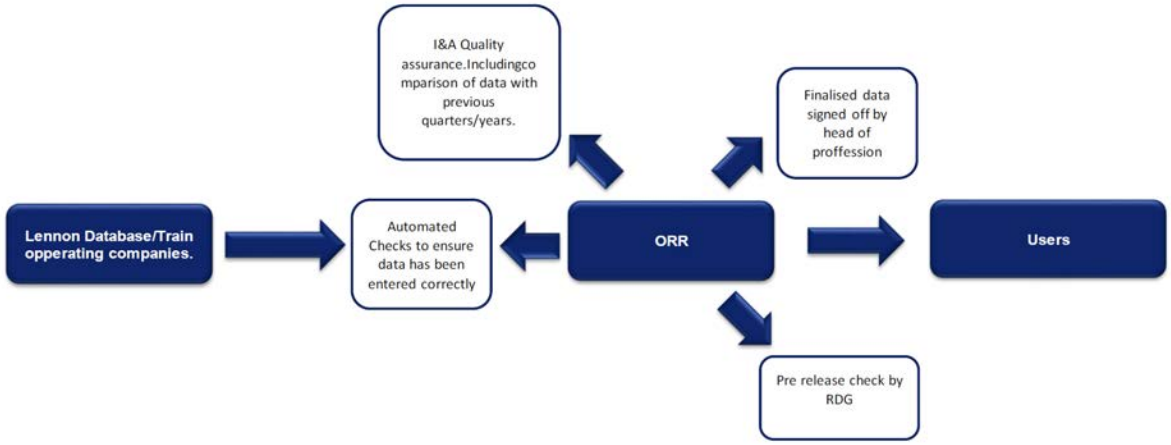


Figure A.8: Process map showing the passenger rail quality assurance checks by data suppliers [114].

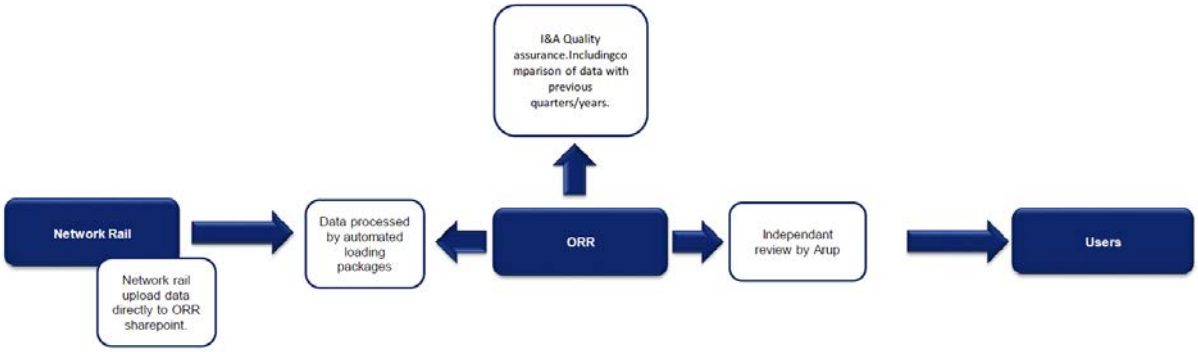


Figure A.9: Process map showing the freight rail quality assurance checks by data suppliers [114].

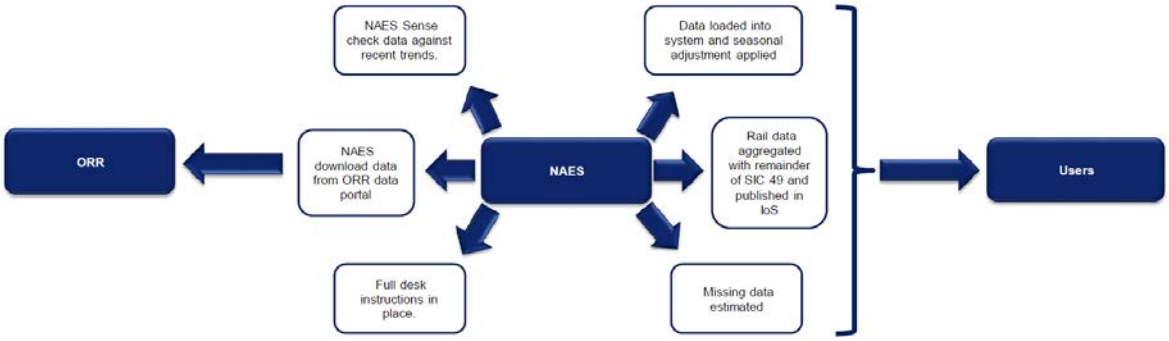


Figure A.10: Process map showing the quality assurance method from the Office for National Statistics (ONS) to users [114].



Figure A.11: Process map showing the overall flow of rail transport statistics from suppliers to the ONS [114].

Appendix B

Chapter 4 Supplements

B.1 Clarification on Notation

Assume a generalised array A, consisting of n distinct elements e. The notation of Chapter 4 follows that

$$A = [e_0, e_1, \ldots, e_n], e_i \equiv A_{[i]}, \tag{B.1}$$

where i is any given index of A. The forms e_i and $A_{[i]}$ (where $e_i \equiv A_{[i]}$) appears interchangably.

B.2 Extended Results

The following tables provide detailed results from Chapter 4.

Table B.1: Frank's response to the MoBS for his approach to currency. The simplified string identity is "1111111321".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S1	0	0	0	0	1	1
A1P2S1	-1	-1	-1	-1	1	2
A1P3S1	-1	-2	-1	-2	1	3
A1P4S1	0	-2	0	-2	1	4
A2P1S1	-1	-3	1	-1	1	5
A2P2S1	1	-2	1	0	1	6
A2P3S1	-1	-3	1	1	1	7
A3P1S3	1	-2	-1	0	-1	6
A3P2S2	1	-1	1	1	0	6
A3P3S1	-1	-2	1	2	1	7
Total Score	-2		2		7	
Adjusted Score	0		2		7	
ISA (%)	0		22		78	
DSA (%)	0		20		70	

Table B.2: Susan's response to the MoBS for her approach to currency. The simplified string identity is "2322121321".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S2	0	0	1	1	-1	-1
A1P2S3	1	1	0	1	0	-1
A1P3S2	-1	0	1	2	-1	-2
A1P4S2	0	0	1	3	-1	-3
A2P1S1	-1	-1	1	4	1	-2
A2P2S2	1	0	1	5	-1	-3
A2P3S1	-1	-1	1	6	1	-2
A3P1S3	1	0	-1	5	-1	-3
A3P2S2	1	1	1	6	0	-3
A3P3S1	-1	0	1	7	1	-2
Total Score	0		7		-2	
Adjusted Score	0		7		0	
ISA (%)	0		100		0	
DSA (%)	0		70		0	

Table B.3: Frank's response to the MoBS for his approach to supply chain. The simplified string identity is "2222121221".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S2	0	0	1	1	-1	-1
A1P2S2	-1	-1	1	2	0	-1
A1P3S2	-1	-2	1	3	-1	-2
A1P4S2	0	-2	1	4	-1	-3
A2P1S1	-1	-3	1	5	1	-2
A2P2S2	1	-2	1	6	-1	-3
A2P3S1	-1	-3	1	7	1	-2
A3P1S2	1	-2	1	8	-1	-3
A3P2S2	1	-1	1	9	0	-3
A3P3S1	-1	-2	1	10	1	-2
Total Score	-2		10		-2	
Adjusted Score	0		10		0	
ISA (%)	0		100		0	
DSA (%)	0		100		0	

Table B.4: Susan's response to the MoBS for his approach to supply chain. The simplified string identity is "2233121221".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S2	0	0	1	1	-1	-1
A1P2S2	-1	-1	1	2	0	-1
A1P3S3	1	0	-1	1	-1	-2
A1P4S3	1	1	-1	0	-1	-3
A2P1S1	-1	0	1	1	1	-2
A2P2S2	1	1	1	2	-1	-3
A2P3S1	-1	0	1	3	1	-2
A3P1S2	1	1	1	4	-1	-3
A3P2S2	1	2	1	5	0	-3
A3P3S1	-1	1	1	6	1	-2
Total Score	1		6		-2	
Adjusted Score	1		6		0	
ISA (%)	14		86		0	
DSA (%)	10		60		0	

Table B.5: Frank's response to the MoBS for his approach to data storage. The simplified string identity is "4344222212".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S4	1	1	0	0	-1	-1
A1P2S3	1	2	0	0	0	-1
A1P3S4	1	3	-1	-1	-1	-2
A1P4S4	1	4	-1	-2	-1	-3
A2P1S2	1	5	-1	-3	-1	-4
A2P2S2	1	6	1	-2	-1	-5
A2P3S2	1	7	-1	-3	-1	-6
A3P1S2	1	8	1	-2	-1	-7
A3P2S1	1	9	1	-1	1	-6
A3P3S2	1	10	-1	-2	-1	-7
Total Score	10		-2		-7	
Adjusted Score	10		0		0	
ISA (%)	100		0		0	
DSA (%)	100		0		0	

Table B.6: Susan's response to the MoBS for his approach to data storage. The simplified string identity is "2322222332".

	Non-Blockchain		Private Blockchain		Public Blockchain	
Statement ID	Score	Cumulative	Score	Cumulative	Score	Cumulative
A1P1S2	0	0	1	1	-1	-1
A1P2S3	1	1	0	1	0	-1
A1P3S2	-1	0	1	2	-1	-2
A1P4S2	0	0	1	3	-1	-3
A2P1S2	1	1	-1	2	-1	-4
A2P2S2	1	2	1	3	-1	-5
A2P3S2	1	3	-1	2	-1	-6
A3P1S3	1	4	-1	1	-1	-7
A3P2S3	1	5	0	1	-1	-8
A3P3S2	1	6	-1	0	-1	-9
Total Score	6		0		-9	
Adjusted Score	6		0		0	
ISA (%)	100		0		0	
DSA (%)	60		0		0	

Bibliography

[1] Sareh Aghaei, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. *International Journal of Web & Semantic Technology*, 3(1):1–10, 2012.

[2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolic, Sharon Weed Cocco, and Jason Yellick. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.

[3] Andreas Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, Inc., 1st edition, 2015.

[4] Frederik Armknecht, Ghassan Karame, Avikarsha Mandal, Franck Youssef, and Erik Zenner. Ripple: Overview and Outlook. In *Trust and Trustworthy Computing*, volume 9229 of *Lecture Notes in Computer Science*, pages 163–180, Greece, 2015. Springer.

[5] Paul Baran. On Distributed Communications Networks. *IEEE transactions on Communications Systems*, 12(1):1–9, 1964. Publisher: IEEE.

[6] Paulo Barreto, Patrick Longa, Michael Naehrig, Jefferson Ricardini, and Gustavo

Zanon. Sharper Ring-LWE Signatures. *IACR Cryptology ePrint Archive*, 2016:1026, 2016.

[7] Georg Becker. Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis. Technical report, Ruhr-University Bochum, 2008.

[8] Juan Benet. IPFS - Content Addressed, Versioned, P2P File System. *arXiv preprint arXiv:1407.3561*, 2014.

[9] Tim Berners-Lee, James Hendler, and Ora Lassila. The Sematic Web. *Scientific American*, 2001.

[10] Daniel Bernstein. Cost Analysis of Hash Collisions: Will Quantum Computers Make SHARCS Obsolete. Unpublished, 2009.

[11] Daniel Bernstein, Johannes Buchmann, and Erik Dahmen. *Post-Quantum Cryptography*. Springer-Verlag Berlin Heidelberg, 1st edition, 2009.

[12] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: Practical Stateless Hash-Based Signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 368–397. Springer, 2015.

[13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the Indifferentiability of the Sponge Construction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 181–197. Springer, 2008.

[14] Jean-Francois Biasse and David Jao. A Quantum Algorithm for Computing Isogenies between Supersingular Elliptic Curves. In *Progress in Cryptology*, volume 8885 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2014.

[15] Grant Blank, William Dutton, and Julia Lefkowitz. Perceived Threats to Privacy Online: The Internet in Britain. Technical report, Oxford Internet Institution, 2019.

[16] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. In *Advances in Cryptology*, volume 7073 of *Lecture Notes in Computer Science*, Seoul, South Korea, December 2011. Springer.

[17] Philip Boucher. What If Blockchain Technology Revolutionised Voting? Technical report, European Parliamentary Research Service, September 2016.

[18] Gilles Brassard, Peter Hoyer, and Alain Tapp. Quantum Algorithm for the Collision Problem. *arXiv preprint quant-ph/9705002*, 1997.

[19] Jerry Brito and Andrea Castillo. Bitcoin: A Primer for Policymakers. Technical report, Mercatus Centre, George Mason University, 2013.

[20] Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera. Overcoming Limits of Blockchain for IoT Applications. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 1–6, 2017.

[21] Johannes Buchmann, Erik Dahmen, and Andreas Hulsing. XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. In *International Workshop on Post-Quantum Cryptography*, pages 117–129. Springer, 2011.

[22] Vitalik Buterin. A Next Generation Smart Contract and Decentralized Application Platform. Technical report, Ethereum Foundation, 2013.

[23] Vitalik Buterin. On Public and Private Blockchains. *Ethereum Blog*, 7, 2015.

[24] Vitalik Buterin. The Meaning of Decentralization, 2017. URL `https://tinyurl.com/y7oblw5r`.

[25] Fran Casino, Thomas K. Dasaklis, and Constantinos Patsakis. A Systematic Literature Review of Blockchain-Based Applications: Current Status, Classification and Open Issues. *Telematics and Informatics*, 36:55 – 81, 2019.

[26] Florent Chabaud and Antoine Joux. Differential Collisions: An Explanation for SHA-1. In *Advances in Cryptology*, Lecture Notes in Computer Science, pages 56–71. Springer Berlin Heidelberg, 1998.

[27] Konstantinos Chalkias, James Brown, Mike Hearn, Tommy Lillehagen, Igor Nitto, and Thomas Schroeter. Blockchained Post-Quantum Signatures. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1196–1203. IEEE, 2018.

[28] Kai Chen. Decentralization Analysis Based on Blockchain Technology Related Projects. In *Mediterranean Conference on Information Systems*, 2019.

[29] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on Post-Quantum Cryptography. Technical report, US Department of Commerce, National Institute of Standards and Technology, 2016.

[30] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On Security Analysis of Proof-of-Elapsed-Time (PoET). In Paul Spirakis and Philippas Tsigas, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 10616 of *Lecture Notes in Computer Science*, pages 282–297, Cham, 2017. Springer International Publishing.

[31] Yongle Chen, Hui Li, Kejiao Li, and Jiyang Zhang. An Improved P2P File System Scheme Based on IPFS and Blockchain. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2652–2657. IEEE, 2017.

[32] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016.

[33] David Lee Kuo Chuen. *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*. Academic Press, 2015.

[34] Bram Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, volume 6, pages 68–72, 2003.

[35] Andrea Corbellini. Elliptic Curve Cryptography: A Gentle Introduction, May 2015. URL `https://tinyurl.com/yybhntq5`.

[36] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient Algorithms for Supersingular Isogeny Diffie-Hellman. In *Annual International Cryptology Conference*, pages 572–601. Springer, 2016.

[37] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174. Springer, 2001.

[38] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, and others. Blockchain Technology: Beyond Bitcoin. *Applied Innovation*, 2(6-10):71, 2016.

[39] Matt Crypto. AES, 2006. Diagram.

[40] Jan Czajkowski, Leon Groot Bruinderink, Andreas Hülsing, Christian Schaffner, and Dominique Unruh. Post-Quantum Security of the Sponge Construction. In *International Conference on Post-Quantum Cryptography*, pages 185–204. Springer, 2018.

[41] J Daemen and V Rijmen. AES Proposal: Rijndael. Technical report, Multiple Institutions, 1998.

[42] Chris Dannen. *Introducing Ethereum and Solidity*, volume 1st. Springer, 2017.

[43] Luca De Feo and Steven D Galbraith. Actions, SeaSign: Compact Isogeny Signatures from Class Group Actions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 759–789. Springer, 2019.

[44] Joost De Kruijff and Hans Weigand. Understanding the Blockchain Using Enterprise Ontology. In *International Conference on Advanced Information Systems Engineering*, pages 29–43. Springer, 2017.

[45] Henry de Valence. SIDH Golang Package, 2018.

[46] Bert Den Boer and Antoon Bosselaers. An Attack on the Last Two Rounds of MD4. In *Advances in Cryptology*, volume 576 of *Lecture Notes in Computer Science*, pages 194–203. Springer Berlin Heidelberg, 1991.

[47] Department for Transport. Joint Rail Data Action Plan: Addressing Barriers to Make Better Use of Rail Data. Policy Paper, Department for Transport, August 2018.

[48] Whitfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[49] Digiconimist. Bitcoin Energy Consumption Index, 2019. URL `https://tinyurl.com/y2yvw3b9`.

[50] Hans Dobbertin. Cryptanalysis of MD4. *Journal of Cryptology*, 11(4):253–271, 1998.

[51] Hans Dobbertin. The First Two Rounds of MD4 Are Not One-Way. In *Fast Software Encryption*, volume 1372 of *Lecture Notes in Computer Science*, pages 284–292. Springer Berlin Heidelberg, 1998.

[52] Leo Ducas, Alain Durmus, Tancrede Lepoint, and Vadim Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *Annual Cryptology Conference*, pages 40–56. Springer, 2013.

[53] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[54] Steven Galbraith, Christopher Petit, Barak Shani, and Bo Ti. On the Security of Supersingular Isogeny Cryptosystems. In *Advances in Cryptology*, volume 10031 of *Lecture Notes in Computer Science*, pages 63–91. Springer, 2016.

[55] Steven Galbraith, Christopher Petit, and Javier Silva. Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. In *Advances in Cryptology*, volume 10624 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2017.

[56] Valentina Gatteschi, Fabrizio Lamberti, Claudio Demartini, Chiara Pranteda, and Victor Santamaria. To Blockchain or Not to Blockchain: That Is the Question. *IT Professional*, 20(2):62–74, 2018. Publisher: IEEE.

[57] H Gilbert and T Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In T Iwata S. Jong, editor, *Fast Software Encryption*, volume 6147 of *Lecture Notes in Computer Science*, Seoul, South Korea, 2010. Springer.

[58] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *Journal of the Association for Computing Machinery*, 38(3):690–728, 1991.

[59] Christian Gorenflo, Stephen Lee, Lukasz Golab, and S Keshav. FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second. Unpublished, 2019.

[60] Sri Nikhil Gupta Gourisetti, Michael Mylrea, and Hirak Patangia. Evaluation and Demonstration of Blockchain Applicability Framework. *IEEE Transactions on Engineering Management*, 67(4):1142–1156, 2019.

[61] Gideon Greenspan. Avoiding the Pointless Blockchain Project, 2015. URL `https://tinyurl.com/sbsx3nh`.

[62] T R Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[63] Trading Strategy Guides. VeChain Cryptocurrency Strategy–The Power of Three, 2018.

[64] Martin Haferkorn and Josué Manuel Quintana Diaz. Seasonality and Interconnectivity Within Cryptocurrencies-an Analysis on the Basis of Bitcoin, Litecoin and Namecoin. In *International Workshop on Enterprise Applications and Services in the Finance Industry*, pages 106–120. Springer, 2014.

[65] Helena Handschuh. SHA-0, SHA-1, SHA-2 (Secure Hash Algorithm). In Henk van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*. Springer United States, Boston, Massachusetts, second edition, 2011.

[66] Ingo Hansen. *Timetable Planning and Information Quality*. WIT Press, 2010.

[67] Da Hongfei and Erik Zhang. NEO White Paper. Technical report, NEO Foundation, 2013.

[68] Jen-Ho Hsiao, Raylin Tso, Chien-Ming Chen, and Mu-En Wu. Decentralized E-Voting Systems Based on the Blockchain Technology. In *Advances in Computer Science and Ubiquitous Computing*, pages 305–309. Springer, 2017.

[69] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating Multi-Target Attacks in Hash-Based Signatures. In *Public-Key Cryptography–PKC 2016*, pages 387–416. Springer, 2016.

[70] Vincenzo Iovino and Ivan Visconti. Non-Interactive Zero Knowledge Proofs in the Random Oracle Model. In *Codes, Cryptology, and Information Security*, volume 11445 of *Lecture Notes in Computer Science*, pages 118–141. Springer, 2019.

[71] Ori Jacobovitz. Blockchain for Identity Management. *The Lynne and William*

Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva, 2016.

[72] David Jao and Luca De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In *Post-Quantum Cryptography*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011.

[73] David Jao and Vladimir Soukharev. Isogeny-Based Quantum-Resistant Undeniable Signatures. In *Post-Quantum Cryptography*, volume 8772 of *Lecture Notes in Computer Science*, pages 160–179. Springer, 2014.

[74] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Infosec Global, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular Isogeny Key Encapsulation. Technical report, University of Waterloo, 2017.

[75] Burt Kaliski. The MD2 Message-Digest Algorithm. *Internet Privacy Task Force*, RFC1319, 1992.

[76] Burt Kaliski. Symmetric Cryptosystem. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 1271–1271. Springer US, Boston, MA, 2011. doi: 10.1007/978-1-4419-5906-5_438.

[77] Henry M Kim and Marek Laskowski. Toward an Ontology-Driven Blockchain Design for Supply-Chain Provenance. *Intelligent Systems in Accounting, Finance and Management*, 25(1):18–27, 2018. Publisher: Wiley Online Library.

[78] P Kravchenko. OK, I Need a Blockchain, But Which One. *Medium*, 2016. URL `https://tinyurl.com/y54wdn3o`.

[79] Ivo Kubjas. *Using Blockchain for Enabling Internet Voting*. Janeiro, 2017.

[80] Helene Köpf. InteGrail – Publishable Final Activity Report. Technical report, Integrail, 2010.

[81] Power Ledger. Power Ledger White Paper. *Power Ledger Pty Ltd, Australia, White paper*, pages 16–21, 2017.

[82] Gaëtan Leurent and Thomas Peyrin. SHA-1 Is a Shambles - First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust. *IACR Cryptol. ePrint Arch.*, 2020:14, 2020.

[83] Antony Lewis. Avoiding Blockchain for Blockchain's Sake: Three Real Use Case Criteria, 2017. URL `https://tinyurl.com/y947auhz`.

[84] Sin Kuang Lo, Xiwei Xu, Yin Kia Chiam, and Qinghua Lu. Evaluating Suitability of Applying Blockchain. In *Engineering of Complex Computer Systems*, pages 158–161. IEEE, November 2017.

[85] I MacLarty, L Langevine, M Vanden Bossche, and P Ross. Using SWRL for Rule-Driven Applications. Technical report, ODASE Ontologies, 2009.

[86] Roger Maull, Phil Godsiff, Catherine Mulligan, Alan Brown, and Beth Kewell. Distributed Ledger Technology: Applications and Implications. *Strategic Change*, 26(5):481–489, 2017. Publisher: Wiley Online Library.

[87] James S. Milne. *Elliptic Curves*. BookSurge, 2006.

[88] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A Review on Consensus Algorithm of Blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572. IEEE, 2017.

[89] Multiple Authors. Rail Settlement Plan, 2020. URL `https://tinyurl.com/y9cxmdz4`.

[90] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report, Bitcoin Foundation, 2009.

[91] Svetlin Nakov. ECIES Hybrid Encryption Scheme, 2020. URL https://tinyurl.com/59v9pdtz.

[92] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin as a Platform. In *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, Princeton, New Jersey, first edition, 2016.

[93] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Introduction to Cryptography and Cryptocurrencies. In *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, Princeton, New Jersey, first edition, 2016.

[94] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Mechanics of Bitcoin. In *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, Princeton, New Jersey, first edition, 2016.

[95] Network Rail. Digital Railway Programme Strategic Plan. Technical report, Network Rail, 2018.

[96] Network Rail. m-Tickets, 2019. URL https://tinyurl.com/y5l6bgb2.

[97] Network Rail. Smart Tickets, 2019. URL https://tinyurl.com/tvne7jj.

[98] Kien Nguyen, Thinh Nguyen, and Yevgeniy Kovchegov. A P2P Video Delivery Network. In *Proceedings of the International Conference on Computer Communications and Networks*, pages 1–7. Institute of Electrical and Electronics Engineers, 2009.

[99] NIST. Federal Information Processing Standards Publication 202. Technical report, National Institute of Standards and Technology, 2015.

[100] NIST. Federal Information Processing Standards Publication 180-4. Technical report, National Institute of Standards and Technology, 2015.

[101] Shen Noether. Ring SIgnature Confidential Transactions for Monero. *IACR Cryptol. ePrint Arch.*, 2015:1098, 2015.

[102] Stephen O'Neal. Who Scales It Best? Inside Blockchains' Ongoing Transactions-Per-Second Race, January 2019. URL `https://tinyurl.com/y5lvz2de`.

[103] Marc Pilkington. Blockchain Technology: Principles and Applications. In *Research Handbook on Digital Transformations*. Edward Elgar Publishing, 2016.

[104] Oleg Pokusaev, Vasily Kupriyanovsky, Dmitry Namiot, Varvara Lazutkina, Aleksandr Zazhigalkin, and Pavel Kupriyanovsky. Blockchain on the Digital Railway in Germany. *International Journal of Open Information Technologies*, 6(2):43–53, 2018.

[105] Porterbrook. Digital, 2020. URL `https://tinyurl.com/y6674ata`.

[106] Joseph Preece and John Easton. Blockchain Technology as a Mechanism for Digital Ticketing. In *Data in the Rail Industry*, September 2018.

[107] Joseph Preece and John Easton. A Review of Prospective Applications of Blockchain Technology in the Railway Industry. Unpublished, September 2018.

[108] Joseph Preece and John Easton. Towards Encrypting Industrial Data on Public Distributed Networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4540–4544. IEEE, 2018.

[109] Joseph Preece and John Easton. Blockchain Technology as a Mechanism for Digital Railway Tickets. In *2019 IEEE International Conference on Big Data*, pages 4540–4544. IEEE, 2019.

[110] Bart Preneel. Hash Functions. In Henk van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 543–553. Springer United States, Boston, Massachusetts, second edition, 2011.

[111] QRL Group. The Quantum-Resistant Ledger. Technical report, QRL Group, 2018.

[112] Raiann Rahman. Elliptic Curves, the Group Law, and the J Invariant. Unpublished, 2016.

[113] Christian Rahmig. railML 3. In *30th railML Conference*, Paris, 2016.

[114] Rail Delivery Group. LENNON Architecture, 2015.

[115] RailTopoModel. Home, 2019. URL `https://tinyurl.com/y629e72x`.

[116] Eric Rescorla and Tim Dierks. The Transport Layer Security (TLS) Protocol Version 1.3. Technical Report 8446, RFC, 2018.

[117] Ronald Rivest. The MD4 Message-Digest Algorithm. *Internet Privacy Task Force*, RFC 1320, 1992.

[118] Ronald Rivest. The MD5 Message-Digest Algorithm. *Internet Privacy Task Force*, RFC 1321, 1992.

[119] Ronald Rivest. The MD6 Message-Digest Algorithm. *Computer Science and Artificial Intelligence Laboratory*, April 2009.

[120] P Rizzo. Nasdaq and Citi Announce Pioneering Blockchain and Global Banking Integration, 2017.

[121] Martin Roetteler and Michael Naerig. Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. *IACR Cryptology ePrint Archive*, 2017.

[122] Phillip Rogaway and Thomas Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[123] Nathalie Rogier and Pascal Chauvaud. MD2 is Not Secure Without the Checksum Byte. *Designs, Codes, and Cryptography*, 12(3):245–251, 1997.

[124] P Sajana, M Sindhu, and M Sethumadhavan. On Blockchain Applications: Hyperledger Fabric and Ethereum. *International Journal of Pure and Applied Mathematics*, 118(18):2965–2970, 2018.

[125] Siemens. Railigent, 2020. URL `https://tinyurl.com/y2ue6prw`.

[126] Joseph H Silverman. *The Arithmetic of Elliptic Curves*. Springer Science & Business Media, 106 edition, 2009.

[127] J Slone. *Local Area Network Handbook*. Auerback Publications, sixth edition, 1999.

[128] Patrícia R Sousa, Luís Antunes, and Rolando Martins. The Present and Future of Privacy-Preserving Computation in Fog Computing. In *Fog Computing in the Internet of Things*, pages 51–69. Springer, 2018.

[129] M Seshadri Srinath and Venkatachalam Chandrasekaran. Isogeny-Based Quantum-Resistant Undeniable Blind Signature Scheme. *IACR Cryptology ePrint Archive*, 2016:148, 2016.

[130] SSL Shop. Symmetric vs Asymmetric Encryption – Which Is More Secure?, 2019. URL `https://tinyurl.com/y5ode3uf`.

[131] Xi Sun, Haibo Tian, and Yumin Wang. Toward Quantum-Resistant Strong Designated Verifier Signature from Isogenies. In *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*, pages 292–296. IEEE, 2012.

[132] Melanie Swan. *Blockchain: Blueprint for a New Economy*. " O'Reilly Media, Inc.", 2015.

[133] Tim Swanson. Consensus-as-a-Service: A Brief Report on the Emergence of Permissioned, Distributed Ledger Systems. Technical report, RCEV, 2015.

[134] Nick Szabo. Smart Contracts. *Virtual School*, 1994.

[135] Nick Szabo. The Idea of Smart Contracts. *Nick Szabo's Papers and Concise Tutorials*, 6, 1997.

[136] Michael Szydlo. Merkle Tree Traversal in Log Space and Time. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 541–554. Springer, 2004.

[137] Huimin Tang, Yong Shi, and Peiwu Dong. Public Blockchain Evaluation Using Entropy and TOPSIS. *Expert Systems with Applications*, 117:204–210, 2019. Publisher: Elsevier.

[138] B Tao and H Wu. Improving the Biclique Cryptanalysis of AES. In E Foo and D Stebila, editors, *Information Security and Privacy*, volume 9144 of *Lecture Notes in Computer Science*. Springer, 2015.

[139] Paolo Tasca, Thayabaran Thanabalasingham, and Claudio J Tessone. Ontology of Blockchain Technologies. Principles of Identification and Classification. *SSRN Electronic Journal*, 10, 2017.

[140] Viacoin Dev Team. Viacoin Whitepaper, 2017.

[141] Theymos. Blockchain Formation, May 2015. URL `https://tinyurl.com/aq68q43`.

[142] J Tutcher, J M Easton, and C Roberts. Enabling Data Integration in the Rail Industry Using RDF and OWL: The RaCoOn Ontology. *Journal of Risk and Uncertainty in Engineering Systems*, 3(2), 2015.

[143] Jonathan Tutcher. *Development of a Semantic Data Model to Support Data Interoperability in the Rail Industry*. PhD Thesis, University of Birmingham, 2015.

[144] Jude Umeh. Blockchain: Double Bubble or Double Trouble? *Itnow*, 58(1):58–61, 2016.

[145] Wouter van der Linde, Peter Schwabe, Andreas Hulsing, Yuval Yarom, and Lejla Batina. *Post-Quantum Blockchain Using One-Time Signature Chains*. PhD Thesis, Radboud University, 2018.

[146] Jaques Velu. Isogenies Entre Courbes Elliptiques. *Comptes Rendus de l'Académie des Sciences*, 273(Series A):305–347, 1971.

[147] Han Vinck. Diffie-Hellman Key Exchange, 2011. URL https://tinyurl.com/y3wkur5j.

[148] Marko Vukolić. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer, 2015.

[149] World Wide Web Consortium. Semantic Web, 2015. URL https://tinyurl.com/yef7fv4.

[150] Karl Wust and Arthur Gervais. Do you Need a Blockchain? In *Blockchain Technology*, pages 45–54, 2018.

[151] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A Post-Quantum Digital Signature Scheme Based on Supersingular Isogenies. In *Financial Cryptography and Data Security*, volume 10322 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2017.

[152] Mark Zhandry. A Note on the Quantum Collision and Set Equality Problems. *arXiv preprint arXiv:1312.1027*, 2013.

[153] J Leon Zhao, Shaokun Fan, and Jiaqi Yan. *Overview of Business Innovations and Research Opportunities in Blockchain*. SpringerOpen, 2016.

[154] Qiuhong Zheng, Yi Li, Ping Chen, and Xinghua Dong. An Innovative IPFS-Based Storage Model for Blockchain. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 704–708. IEEE, 2018.

[155] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018.