Blocks and Chains

Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms

Synthesis Lectures on Information Security, Privacy, & Trust

Editors
Elisa Bertino, ***Purdue University***
Ravi Sandhu, ***University of Texas, San Antonio***

The Synthesis Lectures Series on Information Security, Privacy, and Trust publishes 50- to 100-page publications on topics pertaining to all aspects of the theory and practice of Information Security, Privacy, and Trust. The scope largely follows the purview of premier computer security research journals such as ACM Transactions on Information and System Security, IEEE Transactions on Dependable and Secure Computing and Journal of Cryptology, and premier research conferences, such as ACM CCS, ACM SACMAT, ACM AsiaCCS, ACM CODASPY, IEEE Security and Privacy, IEEE Computer Security Foundations, ACSAC, ESORICS, Crypto, EuroCrypt and AsiaCrypt. In addition to the research topics typically covered in such journals and conferences, the series also solicits lectures on legal, policy, social, business, and economic issues addressed to a technical audience of scientists and engineers. Lectures on significant industry developments by leading practitioners are also solicited.

Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms
Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl
2017

Digital Forensic Science: Issues, Methods, and Challenges
Vassil Roussev
2016

Differential Privacy: From Theory to Practice
Ninghui Li, Min Lyu, Dong Su, and Weining Yang
2016

Privacy Risk Analysis
Sourya Joyee De and Daniel Le Métayer
2016

Introduction to Secure Outsourcing Computation
Xiaofeng Chen
2016

Database Anonymization: Privacy Models, Data Utility, and Microaggregation-based Inter-model Connections
Josep Domingo-Ferrer, David Sánchez, and Jordi Soria-Comas
2016

Automated Software Diversity
Per Larsen, Stefan Brunthaler, Lucas Davi, Ahmad-Reza Sadeghi, and Michael Franz
2015

Trust in Social Media
Jiliang Tang and Huan Liu
2015

Physically Unclonable Functions (PUFs): Applications, Models, and Future Directions
Christian Wachsmann and Ahmad-Reza Sadeghi
2014

Usable Security: History, Themes, and Challenges
Simson Garfinkel and Heather Richter Lipford
2014

Reversible Digital Watermarking: Theory and Practices
Ruchira Naskar and Rajat Subhra Chakraborty
2014

Mobile Platform Security
N. Asokan, Lucas Davi, Alexandra Dmitrienko, Stephan Heuser, Kari Kostiainen, Elena Reshetova, and Ahmad-Reza Sadeghi
2013

Security and Trust in Online Social Networks
Barbara Carminati, Elena Ferrari, and Marco Viviani
2013

RFID Security and Privacy
Yingjiu Li, Robert H. Deng, and Elisa Bertino
2013

Hardware Malware
Christian Krieg, Adrian Dabrowski, Heidelinde Hobel, Katharina Krombholz, and Edgar Weippl
2013

Private Information Retrieval
Xun Yi, Russell Paulet, and Elisa Bertino
2013

Privacy for Location-based Services
Gabriel Ghinita
2013

Enhancing Information Security and Privacy by Combining Biometrics with Cryptography
Sanjay G. Kanade, Dijana Petrovska-Delacrétaz, and Bernadette Dorizzi
2012

Analysis Techniques for Information Security
Anupam Datta, Somesh Jha, Ninghui Li, David Melski, and Thomas Reps
2010

Operating System Security
Trent Jaeger
2008

Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms

Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl

www.morganclaypool.com

ISBN: 9781627057165 paperback
ISBN: 9781627057134 ebook

DOI 10.2200/S00773ED1V01Y201704SPT020

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON INFORMATION SECURITY, PRIVACY, & TRUST

Lecture #20
Series Editors: Elisa Bertino, *Purdue University*
Ravi Sandhu, *University of Texas, San Antonio*
Series ISSN
Print 1945-9742 Electronic 1945-9750

Blocks and Chains

Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms

Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl
SBA Research

SYNTHESIS LECTURES ON INFORMATION SECURITY, PRIVACY, & TRUST #20

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

The new field of cryptographic currencies and consensus ledgers, commonly referred to as *blockchains*, is receiving increasing interest from various different communities. These communities are very diverse and amongst others include: technical enthusiasts, activist groups, researchers from various disciplines, start-ups, large enterprises, public authorities, banks, financial regulators, business men, investors, and also criminals. The scientific community adapted relatively slowly to this emerging and fast-moving field of cryptographic currencies and consensus ledgers. This was one reason that, for quite a while, the only resources available have been the Bitcoin source code, blog and forum posts, mailing lists, and other online publications. Also the original Bitcoin paper which initiated the hype was published online without any prior peer review. Following the original publication spirit of the Bitcoin paper, a lot of innovation in this field has repeatedly come from the community itself in the form of online publications and online conversations instead of established peer-reviewed scientific publishing. On the one side, this spirit of fast free software development, combined with the business aspects of cryptographic currencies, as well as the interests of today's time-to-market focused industry, produced a flood of publications, whitepapers, and prototypes. On the other side, this has led to deficits in systematization and a gap between practice and the theoretical understanding of this new field. This book aims to further close this gap and presents a well-structured overview of this broad field from a technical viewpoint. The archetype for modern cryptographic currencies and consensus ledgers is Bitcoin and its underlying Nakamoto consensus. Therefore we describe the inner workings of this protocol in great detail and discuss its relations to other derived systems.

KEYWORDS

block, chain, blockchain, Bitcoin, cryptographic currency, Proof-of-Work, Nakamoto consensus, consensus ledger

Contents

Acknowledgments

This research was funded by COMET K1, FFG–Austrian Research Promotion Agency, FFG Bridge Early Stage 846573 A2Bit and FFG Bridge 1 858561 SESC. We want to thank our reviewers, Foteini Baldimtsi, Patrick McCorry and Jong Ho Won, for useful feedback and discussions.

Aljosha Judmayer, Nicholas Stifter, Katharina Krombholz, and Edgar Weippl
May 2017

CHAPTER 1

Introduction

Since the introduction of Bitcoin [117] as a prototype for a decentralized cryptocurrency between 2008 and 2009, the field of cryptocurrency technologies has experienced a rapid growth in popularity. Those technologies that are based on the same or very similar fundamental principles as Bitcoin are commonly referred to as *blockchains*. The term *blockchain* itself was not directly introduced by Satoshi Nakamoto in the original paper [117], but used early on within the Bitcoin community to refer to certain concepts of the cryptocurrency. As a result, there are two common spellings of this term found throughout the literature, namely *blockchain* and *block chain*. Although, the later variant was used by Satoshi Nakamoto in a comment within the original source code,[1] the first one is used frequently in press articles as well as recent academic literature e.g., in publications such as [50], and has established itself as the de facto standard. Therefore, we will use the term *blockchain* throughout this book. Nowadays *blockchain* is used as a nebulous umbrella term to refer to various concepts that are related to cryptocurrency technologies. One goal of this book is to demystify this term and provide a solid introduction to the field it encompasses, i.e., distributed cryptocurrencies, their underlying technologies, as well as their governing consensus mechanisms.

To date, over 700 different cryptocurrencies have been created [1]. Some of those currencies only had a very short lifespan or were merely conceived for fraudulent purposes, while others brought additional innovations and still have vital and vibrant communities today.

The mechanisms and underlying principles of most of these cryptocurrencies are, to a greater or lesser extent, derived from the original Bitcoin protocol. Several of these incarnations may only differ from Bitcoin in their choice of certain constants such as the target block interval or maximum number of currency units that will eventually come into existence. Others have switched to alternative proof-of-work algorithms (e.g., Litecoin [129], Dogecoin [128]), have included additional features (e.g., Namecoin [2], Ethereum [66], Zcash [64]), or have used different distributed consensus approaches (e.g., PeerCoin [96], Ripple [133]).

In the few years since the launch of Bitcoin, the decentralized cryptocurrency has grown to remarkable economic value and currently has a market capitalization of around 17 billion USD.[2]

[1] https://github.com/trottier/original-bitcoin/blob/master/src/main.h#L795-L803

[2] This marked rise in valuation, but also the high volatility of the currency, has made it difficult to provide an estimate that is not quickly superseded and appears hopelessly outdated.

This has not only led to extensive news coverage but also to an increased interest from different communities reaching from technical enthusiasts to business people and investors to criminals and law enforcement agencies.

Mainstream media coverage of security incidents and popular myths around Bitcoin show that its fundamentals are hard to understand for non-expert users and cannot be reconciled with the mental models of traditional currency systems.

Bitcoin was designed to be a decentralized cryptographic currency that does not rely on trusted third parties. It achieves this by combining clever incentive engineering and the right cryptographic primitives with a novel probabilistic distributed consensus approach. This combination and the practical demonstration of its feasibility are proving to be a significant contribution that has the potential to profoundly impact other domains beyond cryptocurrencies. These implications are increasingly gaining attention from the scientific community and relate to other security problems of distributed systems, such as distributed name spaces, secure timestamping, and many more.

All these circumstances make the deployment of Bitcoin as a financial instrument an exciting experiment for researchers in many fields. As stated by Bonneau et al. [27], *"Bitcoin is a rare case where practice seems to be ahead of theory. We consider that a tremendous opportunity for the research community to tackle the many open questions about Bitcoin ...."*

Hence, the use of the underlying technologies, commonly referred to as *blockchain*, has been progressively covered in scientific literature and is more and more finding its way to consumer applications. Despite the rising interest within academia as well as the private sector, many open problems remain in terms of finding a balance between performance, scalability, security, decentralization, and anonymity in such systems.

1.1 ASPECTS OF CRYPTOCURRENCIES

Cryptocurrencies have many different aspects, and can therefore be viewed from various angles, including the *financial and economic perspective*, *legal perspective*, *political and sociological perspective*, as well as *technical* and *socio-technical perspectives*. These very different viewpoints can be separated even further; for example, the technical aspects can be divided into the following non-exhaustive list of fields: *cryptography*, *network and distributed systems*, *game theory*, *data science*, and *software and language security*. In this book, the focus is placed on the **technical perspectives** that are necessary to understand this broad field. In doing so, we also discuss aspects of *human-computer interaction* and *usable security*, which are vital for the adoption of a cryptographic currency and, therefore, also related to the overall level of security a cryptographic currency can offer.

1.2 CRYPTOCURRENCY COMMUNITY

The cryptographic currency community is as diverse as the possible viewpoints on the topic. Cryptocurrencies are, as the name suggests, intended to be used as currencies. Therefore, they attract a variety of different people, including *technology enthusiasts*, *businesses and investors*, *ideologists*, *researchers*, *cypherpunks*, *libertarians*, *public authorities and policy makers*, *financial regulators*, *banks*, and also *criminals*, who exploit anonymity measures and make use of the fact that criminal investigation and de-anonymization techniques are lagging behind. In contrast to that, the distributed nature of Bitcoin-like cryptocurrencies also attracts activists and individuals living in oppressive regimes, as these enable them to manage their digital assets despite political sanctions. This highlights the important role that decentralized currencies can play for inhabitants of such countries.

This composition of the broader Bitcoin community as well as its loose structure, combined with a strong mindset of avoiding trusted single points of failure, might also be one reason why it is sometimes hard to reach consensus regarding the direction of Bitcoin's technological development, as interests might diverge. This book aims to not engage in currently ongoing debates (e.g., regarding the maximum block size) but rather to present a neutral, fact-based introduction to this broad topic.

Following the traditional publication spirit of Satoshi Nakamoto, many papers in this field are self-published or made available online as pre-prints prior to their acceptance at scientific journals or conferences. Therefore, we opted to also reference online resources and pre-prints that have not yet been published in peer reviewed venues. The authors are furthermore maintaining a public bibliography[3] where all references that are made in this book can be found.

1.3 FROM CRYPTOCURRENCY TO BLOCKCHAIN

Early works in the area of cryptographic currencies or *cryptocurrencies* mostly focused on required cryptographic primitives as well as the privacy guarantees that could be achieved in such systems [41, 42, 43]. Thereby, these systems themselves still had to rely on trusted third parties (TTPs) to be able to guarantee correct operation. This necessity changed in 2009 when Bitcoin was launched as the first *decentralized* distributed currency [117] that removed the dependency on TTPs. Bitcoin achieves this through a novel combination of well known primitives and techniques, such as, for example, proof-of-work (PoW), to eventually establish agreement (or *consensus*) amongst all nodes on the state of the underlying transaction ledger. The resulting consensus approach, termed *Nakamoto consensus* [27], allows for permissionless participation [147] by potentially *anonymous* actors.

One core element of Bitcoin and Nakamoto consensus is the *blockchain*. Originally the term *blockchain* was used to refer to the aggregation and agreement on transactions in an immutable ledger. Now *blockchain* is used as an umbrella term to refer to all kinds of cryptocurrency

[3] Bibliography: `https://allquantor.at/blockchainbib`.

technologies. This set of technologies and techniques is also commonly referred to as *blockchain technologies* [32]. Although the term *blockchain* is often not well defined, a rough distinction can be made between *permissionless blockchains*, where participation in the consensus algorithm, at least in principle, is not restricted, and *permissioned blockchain*, where there is a closed set of nodes amongst which consensus has to be reached. For a more detailed definition of the term *blockchain* as used in this book see Section 4.2.2.

1.4 THE ANALOG STONE-BLOCK-CHAIN

Capturing and effectively conveying the basic principles of Bitcoin and other blockchain-based cryptocurrencies to novices, especially those without a technical background, can be a difficult task. When trying to explain the technological innovation and novel approach presented by Bitcoin, you are quickly faced with the problem of having to refer to complex elements such as consensus algorithms and cryptography.

This section provides a completely *analog* example that may be helpful when trying to explain the fundamental mechanisms of blockchain technologies to people without the necessary technological background knowledge. The example of the *stone-block-chain* replaces Bitcoin's complex components with simple, real-world analogies, and while it is, of course, not able to accurately cover all the details, it should capture the basic ideas. Practicality aside, the described system should help illustrate the basic principles of blockchain-based cryptocurrencies.

Nakamotopia: In a land far away, there is a stone age village called *Nakamotopia* whose inhabitants are famous for their stone carvers and general obsession with stone blocks. Up until recently, the Nakamotopians relied on small, round, intricately carved rocks as their currency and medium of exchange. However, crafty individuals found a process that allowed them to easily and quickly carve new rocks and subsequently both the value and trust in the currency was quickly lost in the wake of hyperinflation. In dire need of a new currency, the village elders called for an emergency meeting to discuss the future of the Nakamotopian financial system. Their solution was an ingenious idea for a stone-block-chain that combines the Nakamotopians' obsession with stone blocks and their attraction toward lottery systems. The following three-step scheme was devised, which the Nakamotopians called the block creation ceremony:

Miner selection: Every day, all Nakamotopians meet in the village square. In the first part of the block creation ceremony, every villager puts one small stone, engraved with their (unique) name, into a big wooden box. Thereby, the other villagers oversee the process and check that every villager acts honestly.

This box is then placed on a geyser next to the village. During the selection ceremony, all villagers wait for the geyser to erupt and eject steam so that the box containing all the stones is propelled high up into the air and scatters its contents. The villager whose stone lands closest to the geyser wins the lottery and is elected as the *miner* of the next block.

Figure 1.1: Nakamotopian random miner selection by geyser.

Transaction processing: After a villager has been selected as miner for that day, she has the duty to collect all transactions from the villagers that have not yet been recorded. The villagers who want to perform transactions queue up in front of the miner to inform her about transactions that should be included in the stone-block-chain. A transaction transfers ownership of a certain number of currency units from one name to another and is only valid if the sender actually has at least as many units as he wants to transfer to the receiver. The only exception to this rule is the first transaction that is engraved into the block, which credits the miner with a predetermined number of units as a reward for her efforts. This special miner transaction is also the only way in which new currency units can be created. At the end of this session, the stone block will contain all the transactions the miner has decided to include. The remaining space of the stone block will be filled with the holy termination symbol 0x00 so that no additional transactions can be added, i.e., engraved, later on without being detected. If someone were to polish the entire surface of the stone block to engrave a completely new set of transactions, this would be detectable, since

all blocks must have exactly the same dimensions. During this whole process, the chosen miner is allowed to not include a particular transaction. If this happens, the person who wants the transaction to be included into a stone block has to wait until the next day and hope that the next miner will include the transaction.

Figure 1.2: Transaction processing by engraving transactions into empty stone blocks.

Chaining: After the miner has prepared the current stone block, it is heaved toward the town center. Because of the tremendous size and weight of such a stone block, it takes the combined effort of a large number of villagers to move it at all. If a miner were to engrave invalid transactions or otherwise create a stone block that does not obey the rules that were set out by the elders, no honest villager would help the miner move the block. This ensures that the miner sticks to the rules and does not forfeit her chance to receive the mining reward.

Once a valid stone block has been moved by the villagers into the town center, they lift it on top of the towering stack of previous blocks. Only once a block is placed onto this stack is it considered valid by the Nakamotopians.

Stacking the stone blocks has several advantages: Not only does it establish a logical order of transactions, it also makes it much more difficult to change blocks that are further down in the past. An attacker would need to persuade a large number of villagers to start taking off blocks from the top, each requiring a significant amount of time and effort to be removed, which would not remain unnoticed by honest villagers for very long. On the other hand, if a large number of villagers come to the conclusion that one or several blocks should not belong on top of the chain, they can collectively remove these blocks and replace them, thereby ensuring that the majority always agrees upon the contents of their stone-block-chain.

1.4.1 SECURITY MODEL OF THE STONE-BLOCK-CHAIN

We will now look at the security guarantees such a stone-block-chain can offer and how this analogy relates to the properties current cryptographic currency technologies aim to provide.

Public transaction ledger As with Bitcoin, all transactions that take place in Nakamotopia are recorded in a publicly accessible chain of blocks. The key difference here is that Bitcoin is a pseudonymous system, whereas the Nakamotopians use their real identities in their transactions.

Proof-of-Work The basic requirement for a proof-of-work (PoW) should be that it is hard to produce but easy to verify. In Bitcoin, the PoW also functions as a leader election mechanism that randomly selects a new leader, i.e., creator of a valid PoW, on every new block.

In the stone-block-chain analogy, the properties of the proof-of-work are split into three parts. (I) The work that has been put into crafting the blank blocks beforehand and placing the current one at the top of the chain on town square aims to fulfill the "hard to produce" criterion. (II) Once a block has been placed onto the stone-block-chain, it is still easy to verify by reading the transactions engraved onto it and measuring its dimensions to verify that it complies with the rules defining a valid block layout. (III) The geyser in our example works as a random leader-election mechanism on every new stone-block. In Bitcoin, this is achieved through the probabilistic properties of computing a valid PoW for blocks.

Immutability Since every stone block is huge and has precisely defined dimensions, it is unlikely that the effort required for changing a previous stone block in the chain will go unnoticed by several honest Nakamotopians. Even if someone manages to craft a completely new stone block that includes malicious transactions, the effort of replacing an older block in the chain will be detected by some villagers living next to the town square and would also require the collaboration of many dishonest Nakamotopians to be feasible.

In Bitcoin, the blocks are chained together by cryptographic hash functions.

Honest majority Assuming that the majority of villagers are honest, a large portion of the stacked chain of blocks comes from honest villagers and will eventually cease to be in danger of being changed by malicious villagers. Initially there is a slight chance that some of the topmost blocks that have been added to the chain came from malicious villagers while the larger portion of honest Nakamotopians were occupied with other, more pressing issues. Once they return, this honest majority can set about removing the invalid blocks and start replacing them. On the other hand, it takes time for the minority of dishonest villagers to remove or add blocks and both can be quickly detected by any honest villager. If there are enough new stone blocks stacked upon a particular block, it would take the dishonest villagers many days to remove them, making such an attack very unlikely to succeed. Therefore, stone blocks that have been included far enough in the past (i.e., lower in the chain) can be considered agreed upon.

Bitcoin blocks that have a high number of confirmations, i.e., blocks appended after them, are unlikely to change and can, therefore, be considered agreed upon. Although the number of confirmation blocks depends on the value of the transaction in question, common wisdom is that six confirmation blocks are enough to consider a past transaction secure [69].

1.5 STRUCTURE OF THIS BOOK

The remainder of this book is structured as follows: Following a brief introduction of notations and definitions in Chapter 2, Chapter 3 provides a brief overview of the history of cryptocurrencies that led to the invention of Bitcoin. Chapter 4 discusses Bitcoin as the archetype of modern distributed proof-of-work-based cryptocurrencies and highlights the basic properties of blockchain and distributed ledger technologies. Chapter 5 provides an overview of human interactions with cryptocurrency ecosystems on the example of Bitcoin. This highlights the challenges in the area of digital assets management and presents a discussion of Bitcoin usability, privacy, and security challenges from the user's perspective. Chapter 6 addresses the Nakamoto consensus in the context of distributed fault-tolerant computing and highlights the developments toward modeling this new consensus approach. Chapter 7, finally, provides an outlook on future developments of cryptocurrencies and other applications of blockchain technology. For further studies we point the reader to our public bibliography[4] that holds additional references that go beyond the scope of this book.

[4]Bibliography: `https://allquantor.at/blockchainbib`.

CHAPTER 2

Background

This chapter provides a high-level overview of the cryptographic primitives required in the domain of cryptocurrency technologies, as well as explanations of the symbols and notations that are used throughout the book. For the background on distributed fault tolerant computing see Chapter 6.

2.1 CRYPTOGRAPHIC PRIMITIVES

In this section we outline the cryptographic primitives that are required to understand the principles of current PoW-based cryptocurrencies. On a high level the two basic buildings blocks in this context are *cryptographic hash functions* and *asymmetric cryptography*.

2.1.1 CRYPTOGRAPHIC HASH FUNCTIONS

The most important primitive in the context of PoW-based cryptocurrencies are *cryptographic hash functions*. Therefore, we focus on the properties required from such functions as well as the constructions that can be based on it, e.g., *Merkle trees*. While describing the basic properties, we will not go into much detail regarding the security guaranties of the discussed schemes.

Hash function: A hash function H takes a message x of arbitrary but finite size and outputs a fixed size hash h (also called digest). When not explicitly stated differently, we refer to a *cryptographic hash function* whenever the term *hash function* is used in this book.

Cryptographic hash function: There are four additional properties of a hash function that have to be fulfilled so that the function qualifies as a *cryptographic* hash function [106].

1. **Easy to compute:** It is computationally easy to calculate the hash of any given finite message.

$$h = H(x), \text{ Where } h \text{ is of fixed length.} \tag{2.1}$$

2. **Pre-image resistance:** It is infeasible to generate a message that has a given hash value. Infeasible in this context means it cannot be achieved by an adversary as long as the security of the message is important. In terms of complexity theory, this is defined as not being possible in polynomial time. Because of this property, cryptographic hash functions are also called one-way functions.

$$\text{Given a hash } h \text{ it is infeasible to find any message } x \text{ such that } h = H(x). \tag{2.2}$$

3. **Second pre-image resistance:** It is infeasible to find two different messages which produce identical outputs, i.e., a collision, when given as input to the hash function.

$$\text{Given a message } m \text{ it is infeasible to find another message } m' \text{ such that } m \neq m' \text{ and } H(m) = H(m'). \tag{2.3}$$

4. **Collision resistance:** It is infeasible to find *any* two different messages which produce identical outputs, i.e., a collision, when given as input to the hash function.

$$\text{It is infeasible to find any two messages } m, m' \text{ where } m \neq m' \text{ and } H(m) = H(m'). \tag{2.4}$$

Merkle tree: In the paper [107] Merkle introduced the concept of a one-time signature scheme that relies on a *"infinite tree of one-time signatures."* This underlying concept later became known as a *Merkle tree*, *hash tree*, or *authentication tree* [106]. Merkle trees are binary trees in which the leaf nodes are labeled with the values that need to be authenticated and each non-leaf node is labeled with the hash of the labels or values of its child nodes. Figure 2.1 shows an example Merkle tree with $n = 4$ values and the resulting *root hash* or Merkle tree root r. To authenticate a value v_1 and prove that it was part of a Merkle tree with root hash r, the values h_2 and h_6 are required. For more information on Merkle trees see [14].

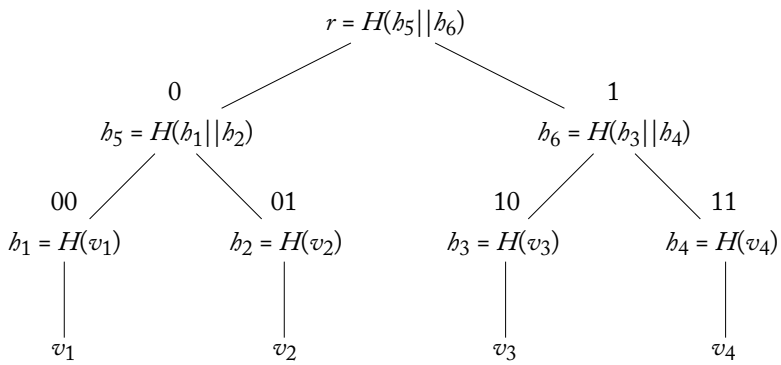

Figure 2.1: Example Merkle tree with $n = 4$ values. Nodes are referenced with a binary string representing their position, e.g., node 01 is labeled h_2.

Some properties of such a tree structure are:

- The length of the path from any leaf to the root of a (balanced) binary tree with n leafs is approximated by $log_2(n)$.
- Given a root hash r and a value v, it requires approximately $log_2(n)$ hash computations to prove that v is indeed a leaf of a (balanced) binary tree.

2.1.2 ASYMMETRIC CRYPTOGRAPHY

The second most important primitive on which cryptographic currencies are based is *asymmetric cryptography*. Since cryptographic currency technologies mostly rely on well researched algorithms and parameters in this context (e.g., Bitcoin uses Secp256k1 [38]), we will not go into detail regarding the aspects concerning this broad field of research.

For further details as well as the mathematical foundations of the topics mentioned in this section please refer to [6, 26, 28, 46, 86, 89, 91].

Public-key encryption: A *public-key encryption* scheme is defined as a triple of efficient algorithms $\mathscr{E} = (G, E, D)$ where,

- G is a key generation algorithm that takes no input and outputs a key pair (pk, sk), where pk is called public key, which can be shared publicly, and sk is called secret key, which should be kept private.
$$(pk, sk) \leftarrow G(). \tag{2.5}$$
- E is a encryption algorithm that takes as input a public-key pk as well as a message $m \in \mathcal{M}$ and outputs a cipher text $c \in \mathcal{C}$ encrypted under the public-key pk associated with the public/secret key pair (pk, sk) of the intended recipient.
$$c \leftarrow E(pk, m). \tag{2.6}$$
- D is a (deterministic) decryption algorithm that takes as input a secret-key sk as well as a cipher text $c \in \mathcal{C}$ and outputs the message $m \in \mathcal{M}$, that was encrypted under the public-key pk associated with sk, or $\perp$ if the wrong keys have been used.
$$m \leftarrow D(sk, c). \tag{2.7}$$

It follows that if the respective operations are reversible $\forall (pk, sk)$ of G it holds that:

$$\forall m \in \mathcal{M} : D(sk, E(pk, m)) = m. \tag{2.8}$$

Digital signatures: A *digital signature* scheme is defined as a triple of efficient algorithms $\mathscr{S} = (G, S, V)$ where,

- G is a key generation algorithm that takes no input and outputs a key pair (pk, sk), where pk is called public key, which can be shared publicly, and sk is called secret key, which should be kept private.
$$(pk, sk) \leftarrow G(). \tag{2.9}$$
- S is a signing algorithm that takes as input a secret key sk as well as a message $m \in \mathcal{M}$ and outputs a signature $\sigma \in \Sigma$ that can be communicated publicly together with the message. S is invoked as
$$S : \sigma \leftarrow E(sk, m). \tag{2.10}$$

- V is a (deterministic) algorithm that takes as input a public-key pk a message $m \in \mathcal{M}$ as well as a signature $\sigma \in \Sigma$ and outputs either `accept` or `reject` depending on the validity of the signature σ on message m.

$$\{\text{accept}, \text{reject}\} \leftarrow V(pk, m, \sigma). \tag{2.11}$$

If follows that a signature generated by S is accepted by V iff (pk, sk) is a valid public/secret key pair. So $\forall(pk, sk)$ of G it holds that:

$$\forall m \in \mathcal{M} : V(pk, m, S(sk, m)) = \text{accept}. \tag{2.12}$$

2.2 NOTATION, SYMBOLS, AND DEFINITIONS

This section provides an overview of the notations and symbols used throughout the book (Table 2.1).

Table 2.1: Notations, symbols, and definitions used in this book

Symbols	Description	Sections
0*xff*	The prefix 0x denotes a hexadecimal representation. In this case the hexadecimal representation of the decimal number 255.	4
$\|\|$	String concatination.	-
$x[251 : 255]$	Refers to the bits from 251 to 255 of variable x.	-
$H(\,)$	Cryptographically secure hash function.	2.1; 4.3
$H^x(\,)$	Chained use of function x times e.g., $H^2(i) = H(H(i))$.	-
SHA 256 ( )	The cryptographic hash function SHA256 as defined in [119].	-
T	The target defines the validity and hardness of a proof-of-work. In Bitcoin a valid PoW is defined as: $\text{SHA256}^2(\text{block header}) \leq T$.	4
z	Number of leading *zero bits* of the 256 bit number T.	4
$Pr(x)$	Probability of x, $0 \leq Pr(x) \leq 1$.	-
$m(p)$	Number of attempts a process p can make when searching for a PoW solution in a unit of time.	-
Π	Set of processes $\{p_1, p_2, \ldots, p_n\} = \Pi$.	-
$\mathcal{B}(t)$	Subset of Byzantine nodes $\mathcal{B} \subseteq \Pi$ at time t.	-
f	Number of faulty processes, $0 \leq f \leq n$ where n denotes the total number of processes.	-
$\Diamond\mathcal{W}$, $\mathcal{P}$, $\Diamond\mathcal{S}(bz)$, $\Diamond\mathcal{M}_{\mathcal{A}}$	Different classes of Failure Detectors.	6.2.2