

Blockchain Technology Overview

Dylan Yaga
Peter Mell
Nik Roby
Karen Scarfone

Blockchain Technology Overview

Dylan Yaga
Peter Mell
*Computer Security Division
Information Technology Laboratory*

Nik Roby
*G2, Inc.
Annapolis Junction, MD*

Karen Scarfone
*Scarfone Cybersecurity
Clifton, VA*

January 2018



U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

50
51

National Institute of Standards and Technology Internal Report 8202
59 pages (January 2018)

52
53
54
55

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

56
57
58
59
60
61

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

62
63
64

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at <https://csrc.nist.gov/publications>.

65

Public comment period: *January 24, 2018* through *February 23, 2018*

66
67
68
69

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: nistir8202-comments@nist.gov

70

All comments are subject to release under the Freedom of Information Act (FOIA).

71

72

Reports on Computer Systems Technology

73 The Information Technology Laboratory (ITL) at the National Institute of Standards and
74 Technology (NIST) promotes the U.S. economy and public welfare by providing technical
75 leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test
76 methods, reference data, proof of concept implementations, and technical analyses to advance
77 the development and productive use of information technology. ITL's responsibilities include the
78 development of management, administrative, technical, and physical standards and guidelines for
79 the cost-effective security and privacy of other than national security-related information in
80 federal information systems.

81

82

Abstract

83 Blockchains are immutable digital ledger systems implemented in a distributed fashion (i.e.,
84 without a central repository) and usually without a central authority. At its most basic level, they
85 enable a community of users to record transactions in a ledger public to that community such that
86 no transaction can be changed once published. This document provides a high-level technical
87 overview of blockchain technology. It discusses its application to electronic currency in depth,
88 but also shows its broader applications. The purpose is to help readers understand how
89 blockchains work, so that they can be appropriately and usefully applied to technology problems.
90 Additionally, this document explores some specific blockchain applications and some examples
91 of when a blockchain system should be considered for use.

92

93

94

Keywords

95 blockchain; consensus model; cryptocurrency; cryptographic hash; distributed ledger; mining

96

97

Acknowledgments

98 The authors wish to thank all contributors to this publication, and their colleagues who reviewed
99 drafts of this report and contributed technical and editorial additions. This includes James Dray,
100 Sandy Ressler, Rick Kuhn, Lee Badger, Eric Trapnell, and Mark Trapnell.

101

102

Audience

103 This publication is designed for readers with little or no knowledge of blockchain technology
104 who wish to understand at a high level how it works and for what it can be used. It is not
105 intended to be a technical guide; the discussion of the technology is abstracted to provide a
106 conceptual understanding. Note that some figures and tables are purposefully simplified to fit the
107 intended audience.

108

109

Trademark Information

110 All registered trademarks and trademarks belong to their respective organizations.

111

112 **Executive Summary**

113 Blockchains are immutable digital ledger systems implemented in a distributed fashion (i.e.,
114 without a central repository) and usually without a central authority. At their most basic level,
115 they enable a community of users to record transactions in a ledger that is public to that
116 community, such that no transaction can be changed once published. In 2008, the blockchain
117 idea was combined in an innovative way with several other technologies and computing concepts
118 to enable the creation of modern cryptocurrencies: electronic money protected through
119 cryptographic mechanisms instead of a central repository. The first such blockchain based
120 approach was Bitcoin. These currency blockchain systems are novel in that they store value, not
121 just information. The value is attached to a digital wallet—an electronic device (or software) that
122 allows an individual to make electronic transactions. The wallets are used to sign transactions
123 sent from one wallet to another, recording the transferred value publicly, allowing all participants
124 of the network to independently verify the validity of the transactions. Each participant can keep
125 a full record of all transactions, making the network resilient to attempts to alter that record (or
126 forge transactions) later.

127 Because there are countless news articles and videos describing the “magic” of the blockchain,
128 this paper aims to describe the method behind the magic (i.e., how a blockchain system works).
129 Arthur C. Clarke once wrote, “Any sufficiently advanced technology is indistinguishable from
130 magic” [1]. Clarke’s statement is a perfect representation for the emerging use cases for
131 blockchain technology. There is a high level of hype around the use of blockchains, yet the
132 technology is not well understood. It is not magical; it will not solve all problems. As with all
133 new technology, there is a tendency to want to apply it to every sector in every way imaginable.
134 This document attempts bring a high-level understanding of the technology so that it can be
135 applied effectively.

136 As stated above, blockchain technology is the foundation of modern cryptocurrencies, so named
137 because of blockchain’s heavy usage of cryptographic functions. Users utilize public and private
138 keys to digitally sign and securely transact within the system. Users of the blockchain may solve
139 puzzles using cryptographic hashing in hopes of being rewarded with a fixed amount of the
140 cryptocurrency. However, blockchain technology is more broadly applicable than its application
141 to cryptocurrencies. In this work, we try to show this broader applicability while still focusing to
142 a large extent on the cryptocurrency use case (since that is the primary use case today).

143 Organizations considering implementing blockchain technology need to understand important
144 aspects of the technology. For example, what happens when an organization implements a
145 blockchain system and then decides they need to make modifications to the data stored? When
146 using a database, this can be accomplished through a simple query (or major changes can be
147 made by updating the database schema or software). However, on a blockchain, it is much more
148 difficult to change data or update the ‘database’ software. Organizations need to understand the
149 extreme difficulty in changing anything that is already on the blockchain, and that changes to the
150 blockchain software may cause forking of the blockchain. Another critical aspect of blockchain
151 technology is how the participants agree that a transaction is valid. This is called “reaching
152 consensus”, and there are many models for doing so, each with positives and negatives for a
153 specific business case.

154 Some existing blockchain technologies focus on storing wealth, while others are a platform for
155 smart contracts (software which is deployed on the blockchain itself, and executed by the
156 computers running that blockchain). New blockchain technologies are being developed
157 constantly to enable new use cases and to improve the efficiency of existing systems. Some
158 blockchain implementations are permissionless, meaning anyone can read and write to them.
159 Other implementations limit participation to specific people or companies, allow finer-grained
160 controls, and may be managed by a central entity. Knowing these specifics allows an
161 organization to understand what will be most applicable to its needs.

162 Despite the many variations of blockchain systems and the rapid development of new
163 technologies, most blockchains use some common core concepts. Each transaction involves one
164 or more addresses and a recording of what happened, and it is digitally signed. Blockchains are
165 comprised of blocks, each block being a group of transactions. All the transactions in a block are
166 grouped together, along with a cryptographic hash of the previous block. Finally, a new hash is
167 created for the current block's header to be recorded within the block data itself as well as within
168 the next block. Over time, each block is then chained to the previous block in the chain by
169 adding the hash of the previous block to the header of the current block.

170 Each technology used in a blockchain system takes existing, proven concepts and merges them
171 together in a way that can address problems that were previously difficult. This document
172 explores the fundamentals of how blockchain technologies work, how the participants in the
173 network come to agree whether a transaction is valid, what happens when changes need to be
174 made to an existing blockchain deployment, and how permissions work. Additionally, this
175 document explores specific blockchain applications and examples of when to consider using a
176 blockchain system.

177 The use of blockchain technology is not a silver bullet, and there are issues that must be
178 considered such as how to deal with malicious users, how controls are applied, and the
179 limitations of any blockchain implementation. That said, blockchain technology is an important
180 concept that will be a basis for many new solutions.

Table of Contents

181

182 **Executive Summary iv**

183 **1 Introduction 9**

184 1.1 Background and History..... 9

185 1.2 Purpose and Scope 10

186 1.3 Notes on Terms 10

187 1.4 Document Structure 10

188 **2 Blockchain Architecture..... 12**

189 2.1 Hashes..... 12

190 2.2 Transactions 13

191 2.3 Asymmetric-Key Cryptography 13

192 2.4 Addresses and Address Derivation..... 14

193 2.4.1 Private Key Storage..... 14

194 2.5 Ledgers..... 15

195 2.6 Blocks 19

196 2.7 Chaining Blocks 23

197 **3 Blockchains in Operation..... 23**

198 **4 Consensus..... 26**

199 4.1 Proof of Work Consensus Model 26

200 4.2 Proof of Stake Consensus Model 29

201 4.3 Round Robin Consensus Model 30

202 4.4 Ledger Conflicts and Resolutions 30

203 **5 Forking..... 33**

204 5.1 Soft Forks 33

205 5.2 Hard Forks 33

206 5.3 Cryptographic Changes and Forks 34

207 **6 Smart Contracts 35**

208 **7 Blockchain Categorization 36**

209 7.1 Permissioned 36

210 7.1.1 Application Considerations for Permissioned Blockchains 36

211 7.1.2 Use Case Examples 37

212 7.2 Permissionless..... 38

213 7.2.1 Application Considerations for Permissionless Blockchains..... 38

214 7.2.2 Use Case Examples 38

215 **8 Blockchain Platforms 40**

216 8.1 Cryptocurrencies..... 40

217 8.1.1 Bitcoin (BTC) 40

218 8.1.2 Bitcoin Cash (BCC) 41

219 8.1.3 Litecoin (LTC)..... 41

220 8.1.4 Ethereum (ETH) 41

221 8.1.5 Ethereum Classic (ETC)..... 41

222 8.1.6 Dash (DASH)..... 42

223 8.1.7 Ripple (XRP) 42

224 8.2 Hyperledger 42

225 8.2.1 Hyperledger Fabric..... 42

226 8.2.2 Hyperledger Sawtooth..... 43

227 8.2.3 Hyperledger Iroha..... 43

228 8.2.4 Hyperledger Burrow..... 43

229 8.2.5 Hyperledger Indy 43

230 8.3 MultiChain..... 43

231 **9 Blockchain Limitations and Misconceptions..... 44**

232 9.1 Blockchain Control..... 44

233 9.2 Malicious Users..... 44

234 9.3 No Trust..... 45

235 9.4 Resource Usage 45

236 9.5 Transfer of Burden of Credential Storage to Users..... 46

237 9.6 Private/Public Key Infrastructure and Identity 46

238 **10 Conclusions..... 47**

List of Appendices

241 **Appendix A— Acronyms 48**

242 **Appendix B— Glossary 50**

243 **Appendix C— References 55**

List of Tables and Figures

246 Table 1: Examples of Inputs and SHA-256 Digest Values 12

247 Table 2: Example Transaction..... 13

248 Figure 1 - A simple network maintaining a copy of a ledger across nodes..... 16

249 Figure 2 - Submitting a Transaction to a Node, waiting in the Pending Transaction List

250 17

251 Figure 3 - Transaction 4 information transmitted from node to node..... 18

252 Figure 4 - Transaction 4 has been included into a block, nodes are transmitting the

253 information; the final node has not yet received the latest information..... 19

254 Figure 5: Example of a Merkle Tree 21

255 Figure 6: Blockchain with Merkle Tree 22

256 Figure 7: Generic Chain of Blocks..... 23

257 Figure 8: Transaction Being Added to Unspent Transaction Pool..... 24

258 Figure 9: Finalized Block (Generalized) 25

259 Figure 10: Distributed Network in Conflict 31

260 Figure 11: Blockchains in Conflict 31

261 Figure 12: Chain B Adds the Next Block 32

262 Table 3: Impact of Quantum Computing on Common Cryptographic Algorithms 34

263

264 **1 Introduction**

265 Blockchains are immutable digital ledger systems implemented in a distributed fashion (i.e.,
266 without a central repository) and usually without a central authority. At their most basic level,
267 they enable a community of users to record transactions in a ledger that is public to that
268 community, such that no transaction can be changed once published. This technology became
269 widely known starting in 2008 when it was applied to enable the emergence of electronic
270 currencies where digital transfers of money take place in distributed systems. It has enabled the
271 success of e-commerce systems such as Bitcoin, Ethereum, Ripple, and Litecoin. Because of this,
272 blockchains are often viewed as bound to Bitcoin or possibly e-currency solutions in general.
273 However, the technology is more broadly useful and is available for a variety of applications.

274 The numerous components of blockchain technology along with its reliance on cryptographic
275 primitives and distributed systems can make it challenging to understand. However, each
276 component can be described simply and used as a building block to understand the larger
277 complex system. We provide an informal concise description of blockchain technology:

278 Blockchains are distributed digital ledgers of cryptographically signed transactions that are
279 grouped into blocks. Each block is cryptographically linked to the previous one after
280 validation and undergoing a consensus decision. As new blocks are added, older blocks
281 become more difficult to modify. New blocks are replicated across all copies of the ledger
282 within the network, and any conflicts are resolved automatically using established rules.

283 **1.1 Background and History**

284 The core ideas behind blockchain technology emerged in 1991 when a signed chain of
285 information was used as an electronic ledger for digitally signing documents in a way that could
286 easily show none of the signed documents in the collection had been changed [2]. It was first
287 applied to digital cash in 2008 in the initial paper describing the Bitcoin electronic cash solution,
288 *Bitcoin: A Peer to Peer Electronic Cash System* [3], which was published pseudonymously by
289 Satoshi Nakamoto. The actual author(s) and owner of the first Bitcoins remain a mystery. Since
290 then, blockchain technology has become tightly linked to Bitcoin and is often assumed to be used
291 for monetary transactions (although it is not restricted to simple fund transfers). Nakamoto's
292 paper contained the blueprint that most modern digital cash schemes follow, with many
293 variations. Bitcoin is in fact the first of many applications or use cases for a blockchain.

294 Many electronic cash schemes existed prior to Bitcoin, but none of them achieved widespread
295 use. By adopting blockchain technology, Bitcoin achieved compelling capabilities that promoted
296 its use. The use of a blockchain enabled Bitcoin to be implemented in a distributed fashion so
297 that no single user controlled the currency and no single point of failure existed. Its primary
298 benefit was to enable direct electronic financial transactions between users without the need for a
299 third party. It also enabled the issuance of new currency in a fair fashion to those users
300 (sometimes called *miners* or *minters*) maintaining the blockchain that, among other factors,
301 enabled lower transaction costs for using the system. The payment of the mining nodes enabled
302 distributed administration of the system without the need to organize those maintaining the
303 system. By using a distributed blockchain and consensus-based maintenance, a self-policing
304 mechanism was created that ensured that only valid transactions were added to the blockchain.

305 Also, the blockchain enabled users to be *pseudonymous*, meaning that users are anonymous but
306 their accounts are not – all their transactions are publicly observable. This has effectively
307 enabled Bitcoin to offer pseudo-anonymity because accounts can be created without any
308 identification or authorization process. Finally, the distributed maintenance of the blockchain
309 created a system with complete transparency, which promoted trust in its use. Since all
310 transactions are transparent within the system, and must be verified before being included, it
311 greatly reduces the ability for users to *double spend* (sending the same digital asset to more than
312 one user) their digital assets. One of the most valuable aspects of applications built on
313 blockchains is that they can enable business to be conducted with untrusted and unknown users.

314 **1.2 Purpose and Scope**

315 This document provides a high-level technical overview of blockchain technology. It discusses
316 its application for electronic currency in depth, but also shows its broader uses. It looks at
317 different categories of approaches since many blockchain platforms exist, each subtly different.
318 This document is intended to help readers to understand the technologies which comprise
319 blockchain systems and to understand how blockchains can be appropriately and usefully applied
320 to technology problems.

321 **1.3 Notes on Terms**

322 The terminology for blockchain technology varies from one implementation to the next – in
323 order to talk about the technology as a whole, generic terms will be used. Throughout this
324 document the terms *user* and *node* are used to describe aspects of blockchain components. For
325 the purposes of this document, a *user* is a generic term to describe any person, organization,
326 entity, business, government, etc. which is utilizing the blockchain system. A *node* is an
327 individual system within a blockchain system, and can further be refined to *full node* (stores the
328 entire blockchain), *mining node* (full node that also maintains the blockchain by publishing new
329 blocks), and *lightweight node* (node that does not maintain a history of the entire blockchain).

330 **1.4 Document Structure**

331 The rest of this document consists of the following sections and appendices:

- 332 • Section 2 defines the high-level components of a blockchain system architecture,
333 including hashes, transactions, ledgers, blocks, and blockchains.
- 334 • Section 3 discusses how a blockchain is expanded through the addition of new blocks
335 representing sets of transactions.
- 336 • Section 4 examines the need for consensus models to resolve conflicts among blockchain
337 mining nodes.
- 338 • Section 5 introduces the concept of forking.
- 339 • Section 6 defines and discusses smart contracts.
- 340 • Section 7 looks at blockchain permission models, discusses their application
341 considerations, and provides use case examples for each model.
- 342 • Section 8 provides several examples of blockchain platforms in use today to indicate the
343 variations from one platform to another.
- 344 • Section 9 highlights some of the limitations of blockchain technology.

- 345
- Section 10 gives a short conclusion for the document.
- 346
- Appendix A contains a glossary for selected terms defined in the document.
- 347
- Appendix B provides a list of acronyms and abbreviations used in the document.
- 348
- Appendix C defines the references used throughout the document.

349

350 2 Blockchain Architecture

351 Blockchain systems can seem complex; however, they can be easily understood by examining
 352 each component technology individually. At a high level, blockchains utilize well-known
 353 computer science mechanisms (linked lists, distributed networking) as well as cryptographic
 354 primitives (hashing, digital signatures, public/private keys) mixed with financial concepts (such
 355 as ledgers).

356 2.1 Hashes

357 An important component of the blockchain technology is the use of cryptographic hash functions
 358 for many operations, such as hashing the content of a block. *Hashing* is a method of calculating a
 359 relatively unique fixed-size output (called a *message digest*, or just *digest*) for an input of nearly
 360 any size (e.g., a file, some text, or an image). Even the smallest change of input (e.g., a single bit)
 361 will result in a completely different output digest. Table 1 shows simple examples of this. Hash
 362 algorithms are designed to be one-way (known as being preimage resistant): it is computationally
 363 infeasible to find any input that maps to any pre-specified output. If a particular output is desired,
 364 many inputs must be tried by passing them through the hash function until an input is found that
 365 produces the desired result. Hash algorithms are also designed to be collision resistant (known as
 366 second preimage resistant): it is computationally infeasible to find two or more inputs that
 367 produce the same output.

368 A hashing algorithm used in many blockchain technologies is the Secure Hash Algorithm (SHA)
 369 with an output size of 256 bits (SHA-256). Many computers support this algorithm in hardware,
 370 making it fast to compute. This algorithm has an output of 32 (8-bit) characters (shown below, in
 371 Table 1, as a 64-character hexadecimal string), meaning that there are $2^{256} \approx 10^{77}$, or
 372 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936
 373 possible digest values. The algorithm for SHA-256, as well as others, is specified in Federal
 374 Information Processing Standard (FIPS) 180-4 [4]. The NIST Secure Hashing website [5]
 375 contains FIPS specifications for all NIST-approved hashing algorithms.

376 **Table 1: Examples of Inputs and SHA-256 Digest Values**

Input Text	SHA-256 Digest Value
1	0x6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdf6d6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

377
 378 Since there is an extremely large number of possible input values and a finite number of possible
 379 output digest values, it is possible to have a collision where $\text{hash}(x) = \text{hash}(y)$ (i.e., the hash of
 380 two different inputs produces the same digest). However, it is highly unlikely for any such input
 381 x and y that produce the same digest to both be valid in the context of the blockchain system (in
 382 this case, both being valid blockchain transactions) as well as be computed reasonably close to
 383 each other in time. The hashing algorithm used (SHA-256) is said to be collision resistant, since
 384 to find a collision in SHA-256, one would have to execute the algorithm, on average, about 2^{128}

385 times. Blockchain technologies take a list of transactions and create a hash “fingerprint” (the
 386 digest is the fingerprint) for the list. Anyone with the same list of transactions can generate the
 387 exact same fingerprint. If a single value in a transaction within the list changes, the digest for that
 388 block changes, making it easy to discover even minor one bit changes.

389 2.2 Transactions

390 A *transaction* is a recording of a transfer of assets (digital currency, units of inventory, etc.)
 391 between parties. An analog to this would be a record in a checking account for each time money
 392 was deposited or withdrawn. Table 2 shows a notional example of a transaction. Each block in a
 393 blockchain contains multiple transactions. A single transaction typically requires at least the
 394 following information fields, but can contain more:

- 395 • **Amount** – The total amount of the digital asset to transfer.
- 396 • **Inputs** – A list of the digital assets to be transferred (their total value equals the amount).
 397 Note that each digital asset is uniquely identified and may have different values from
 398 other assets. However, assets cannot be added or removed from existing digital assets.
 399 Instead, digital assets can be split into multiple new digital assets (each with lesser value)
 400 or combined to form fewer new digital assets (each with a correspondingly greater value).
- 401 • **Outputs** – The accounts that will be the recipients of the digital assets. Each output
 402 specifies the value to be transferred to the new owner(s), the identity of the new owner(s),
 403 and a set of conditions the new owners must meet to receive that value. If the digital
 404 assets provided are more than required, the extra funds are returned to the sender (this is a
 405 mechanism to “make change”).
- 406 • **Transaction ID/Hash** – A unique identifier for each transaction. Some blockchains use
 407 an ID, and others take a hash of the specific transaction as a unique identifier.

408 **Table 2: Example Transaction**

	Input	Output	Amount	Total
Transaction ID: 0xa1b2c3	Account A	Account B	0.0321	
		Account C	2.5000	
				2.5321

409

410 Determining the validity of a transaction is important. Just because someone claims a transaction
 411 took place does not mean it really happened. Transactions are signed and can be verified with
 412 public/private key pairs at any time.

413 2.3 Asymmetric-Key Cryptography

414 A fundamental technology utilized by blockchain technologies is asymmetric-key cryptography¹
 415 (also referred to as public/private key cryptography). Asymmetric-key cryptography uses a pair

¹ FIPS Publication 186-4, Digital Signature Standard [6] specifies a common algorithm for digital signing used in blockchain technologies: Elliptic Curve Digital Signature Algorithm (ECDSA).

416 of keys: a public key and a private key that are mathematically related to each other. The public
 417 key may be made public without reducing the security of the process, but the private key must
 418 remain secret if the data is to retain its cryptographic protection. Even though there is a
 419 relationship between the two keys, the private key cannot efficiently be determined based on
 420 knowledge of the public key.

421
 422 Asymmetric key cryptography uses the different keys of the key pair for specific functions,
 423 dependent on which service is to be provided. For example, when digitally signing data, the
 424 cryptographic algorithm utilizes the private key to sign. The signature can then be verified using
 425 the corresponding public key.

426 Asymmetric-Key Cryptography Utilization in Blockchain Systems:

- 427 • Private keys are used to digitally sign transactions.
- 428 • Public keys are used to derive addresses, allowing for a one-to-many approach for
 429 pseudonymity (one public key pair can yield multiple addresses; in some cases, multiple
 430 public key pairs are utilized to create multiple addresses).
- 431 • Public keys are used to verify signatures generated with private keys.
- 432 • Asymmetric-key cryptography provides the ability to verify that the user transferring
 433 value to another user is in possession of the private key capable of signing the value.

434 2.4 Addresses and Address Derivation

435 A user's *address* is a short, alphanumeric string derived from the user's public key using a hash
 436 function, along with some additional data (used to detect errors). Addresses are used to send and
 437 receive digital assets. Most blockchain systems make use of addresses as the "to" and "from"
 438 endpoints in a transaction. Addresses are shorter than the public keys and are not secret. To
 439 generate an address, it typically means taking a public key, hashing it, and converting the hash to
 440 text:

441
$$\text{public key} \rightarrow \text{hash function} \rightarrow \text{address}$$

442 Users can generate as many private/public key pairs, and therefore addresses as desired, allowing
 443 for a varying degree of pseudo-anonymity. Addresses act as the public-facing "identity" on a
 444 blockchain for a user, and oftentimes an address will be converted into a QR code for easier use.

445 When a blockchain distributes digital assets, it does so by assigning them to an address. To spend
 446 that digital asset, a user must prove possession of the address's corresponding private key. By
 447 digitally signing a transaction with the private key, the transaction can be verified with the public
 448 key.

449 2.4.1 Private Key Storage

450 Most users of a blockchain system do not record their private keys manually, rather, software
 451 commonly called a *wallet* securely stores them. The wallet can store private keys, public keys,
 452 and associated addresses. The wallet software can also calculate the total number of assets a user
 453 may have.

454 A private key is usually generated using a secure random function, meaning that reconstructing it
455 is difficult, if not impossible. If a user loses a private key, then any asset associated with that key
456 is lost. If a private key is stolen, the attacker will have full access to all assets controlled by that
457 private key. The security of private keys is so important that many users use special secure
458 hardware to store it.

459 Private key storage is an extremely important aspect of blockchain technology. When it is
460 reported in the news that “Bitcoin was stolen from...”, it almost certainly means the private keys
461 were found and used to sign a transaction sending the money to a new account, not that the
462 system was compromised. Note that because blockchain data cannot generally be changed, once
463 a criminal steals a private key and publicly moves the associated funds to another account, it
464 cannot be undone.

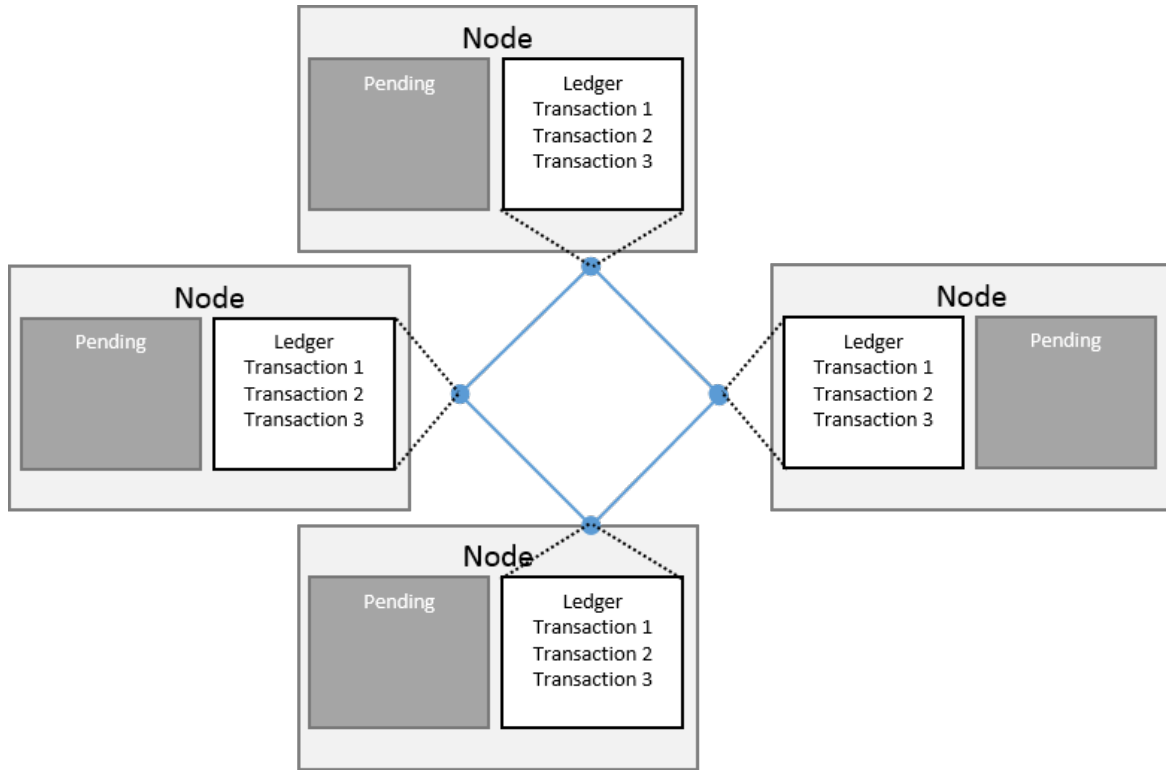
465 **2.5 Ledgers**

466 A *ledger* is a collection of transactions. Throughout history, pen and paper ledgers have been
467 used to keep track of the exchange of goods and services. More recently, ledgers have been
468 stored digitally, often in large databases owned and operated solely by centralized “trusted” third
469 parties on behalf of a community of users (i.e., the third party is the owner of the ledger).
470 Centralized ledgers may have shortcomings, such as:

- 471 • They may be lost or destroyed; a user must trust that the owner is properly backing up the
472 system.
- 473 • The transactions may not be valid; a user must trust that the owner is validating each
474 received transaction.
- 475 • The transaction list may not be complete; a user must trust that the owner is including all
476 valid transactions that have been received.
- 477 • The transaction data may have been altered; a user must trust that the owner is not
478 altering past transactions.

479 Of course, it is in the best interest of any centralized ledger to backup data, validate transactions,
480 include all valid transactions, and not to alter history.

481 A ledger implemented using a blockchain can mitigate these issues through the use of a
482 distributed consensus mechanism. One aspect of this is that the blockchain ledger will be copied
483 and distributed amongst every node within the system. Figure 1 depicts a simple network with
484 four nodes, where each has a copy of a ledger of transactions.



485

486

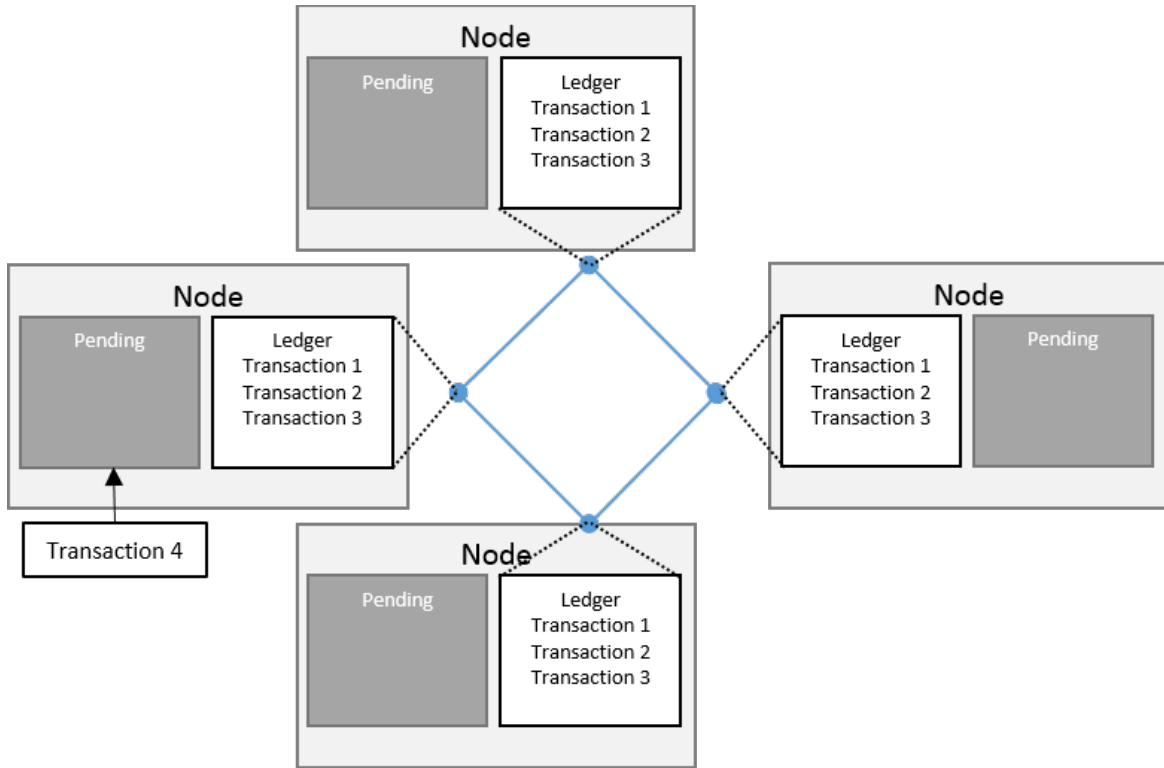
Figure 1 - A simple network maintaining a copy of a ledger across nodes

487

New transactions are submitted to a node (as seen in Figure 2), which will then alert the rest of

488

the network that a new transaction has arrived (as seen in Figure 3).



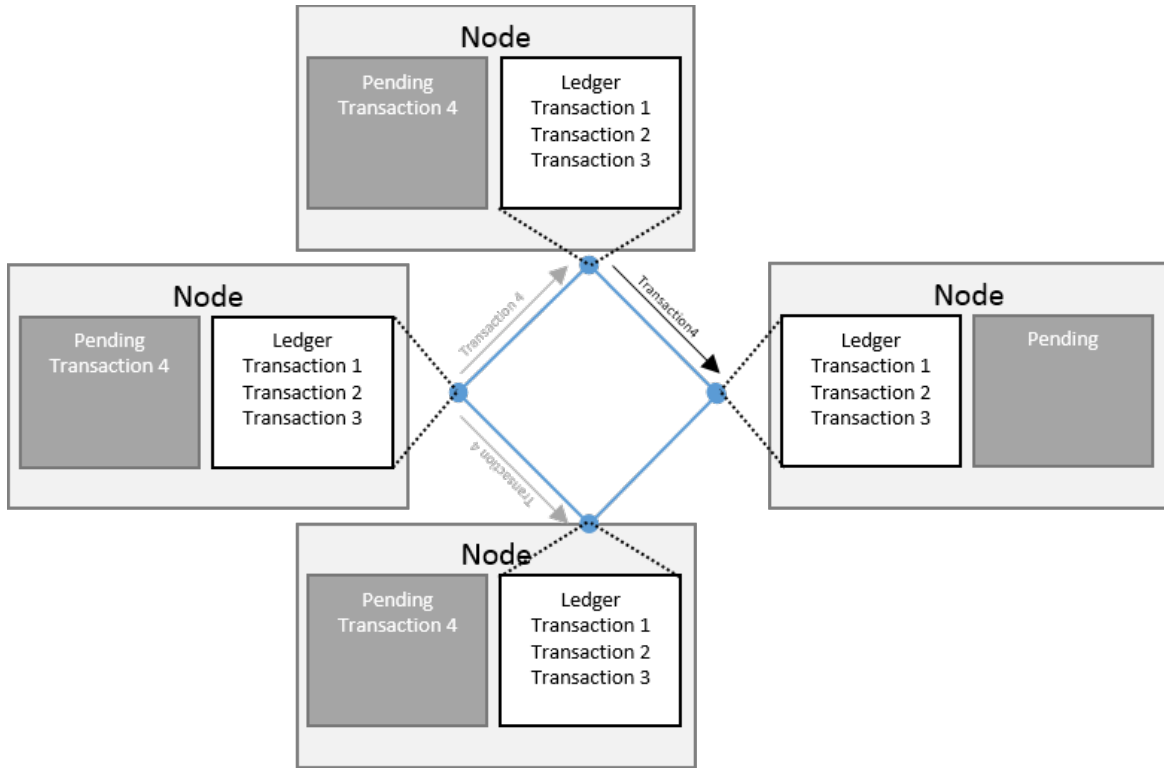
489

490

Figure 2 - Submitting a Transaction to a Node, waiting in the Pending Transaction List

491

At this point, it is a pending transaction, and not included in a block within the ledger.



492

493

Figure 3 - Transaction 4 information transmitted from node to node

494

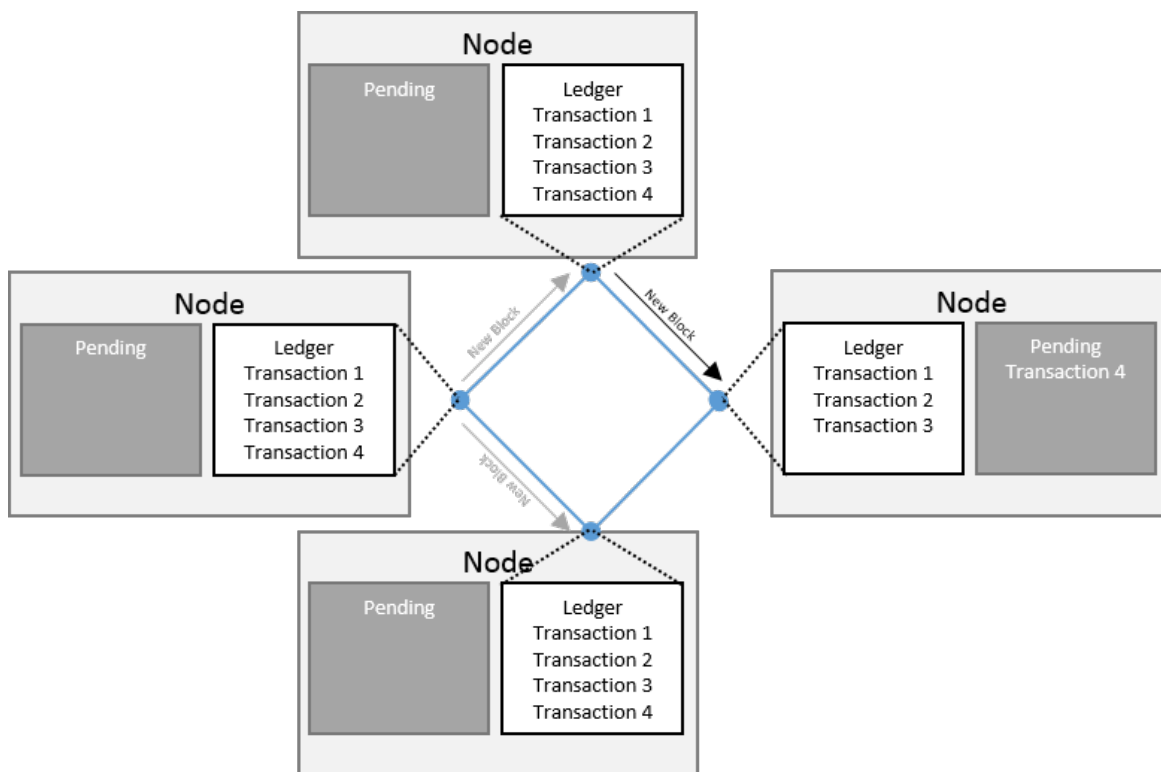
Eventually, a node will include this new transaction within a block and complete the system's

495

required consensus method (explained later). This new block will be distributed across the

496

system and all ledgers will be updated to include the new transaction (as seen in Figure 4).



497

498 **Figure 4 - Transaction 4 has been included into a block, nodes are transmitting the information; the final**
 499 **node has not yet received the latest information**

500 Whenever new users join the system, they receive a full copy of the blockchain, making loss or
 501 destruction of the ledger difficult. All transactions are stored in blocks within the blockchain
 502 (transactions discussed in Section 2.2).

503

504 **2.6 Blocks**

505 Users may submit candidate transactions to the ledger by sending these transactions to some of
 506 the nodes participating in the blockchain. Submitted transactions are propagated to the other
 507 nodes in the network (but this by itself does not include the transaction in the blockchain). The
 508 distributed transactions then wait in a queue, or *transaction pool*, until they are added to the
 509 blockchain by a mining node.

510 Mining nodes are the subset of nodes that maintain the blockchain by publishing new blocks.
 511 Transaction are added to the blockchain when a mining node publishes a block. A *block* contains
 512 a set of validated transactions. ‘Validity’ is ensured by checking that the providers of funds in
 513 each transaction (listed in the transaction’s ‘input’ values) have each cryptographically signed
 514 the transaction. This verifies that the providers of funds for a transaction had access to the private
 515 key which could sign over the available funds. The other mining nodes will check the validity of
 516 all transactions in a published block and will not accept a block if it contains any invalid
 517 transactions.

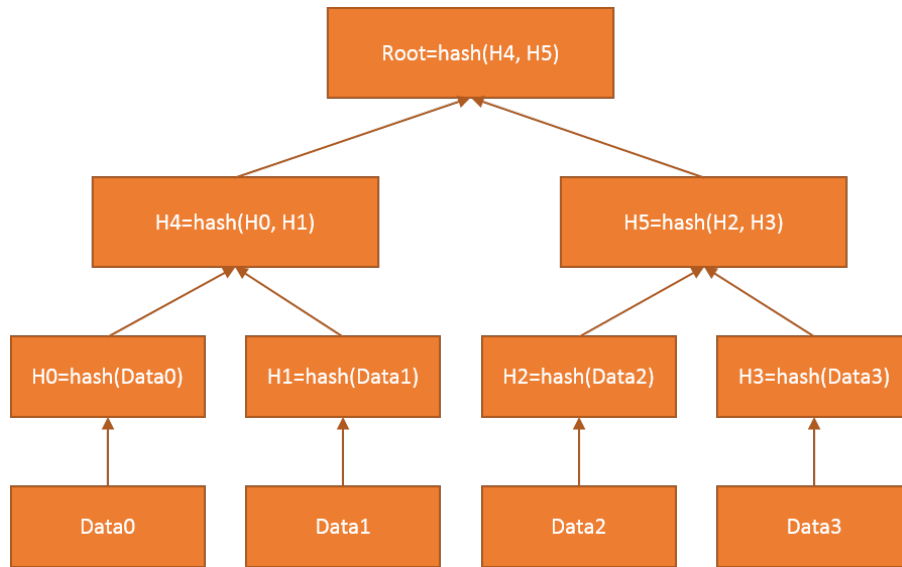
518 After creation, each block is hashed thereby creating a digest that represents the block. The
519 change of even a single bit in the block would completely change the hash value. The block's
520 hash digest is used to help protect the block from change since all nodes will have a copy of the
521 block's hash and can then check to make sure that the block has not been changed.

522 The actual construction of a block is slightly more complicated. The data fields comprising a
523 block typically consist of the following:

- 524 • The block number, also known as block height
- 525 • The current block hash value
- 526 • The previous block hash value
- 527 • The Merkle tree root hash (defined below)
- 528 • A timestamp
- 529 • The size of the block
- 530 • The *nonce value*, which is a number manipulated by the mining node to solve the hash
531 puzzle that gives them the right to publish the block (see Section 4.1 for details)
- 532 • A list of transactions included within the block
- 533

534 Rather than storing the hash of every transaction within the header of a block, a data structure
535 known as a *Merkle tree* is utilized. A Merkle tree combines the hash values of data together until
536 there is a singular root (a *Merkle tree root hash*). The root is an efficient mechanism used to
537 summarize the transactions in a block and verify the presence of a transaction within a block.
538 This structure ensures that the data sent in a distributed network is valid, since any alteration to
539 the underlying data would be detected and can be discarded. Figure 5 shows an example of a
540 Merkle tree:

- 541 • The bottom row represents the data to be summarized, in the case of blockchains this is
542 the transaction data.
- 543 • The second to bottom row shows that data being hashed.
- 544 • The hashed data from the second row is then combined and then hashed on the third to
545 bottom row.
- 546 • Finally, the top row shows the Root hash, which combines and hashes H4 and H5. The
547 root hash is a hash of all previous combinations and hashes made.



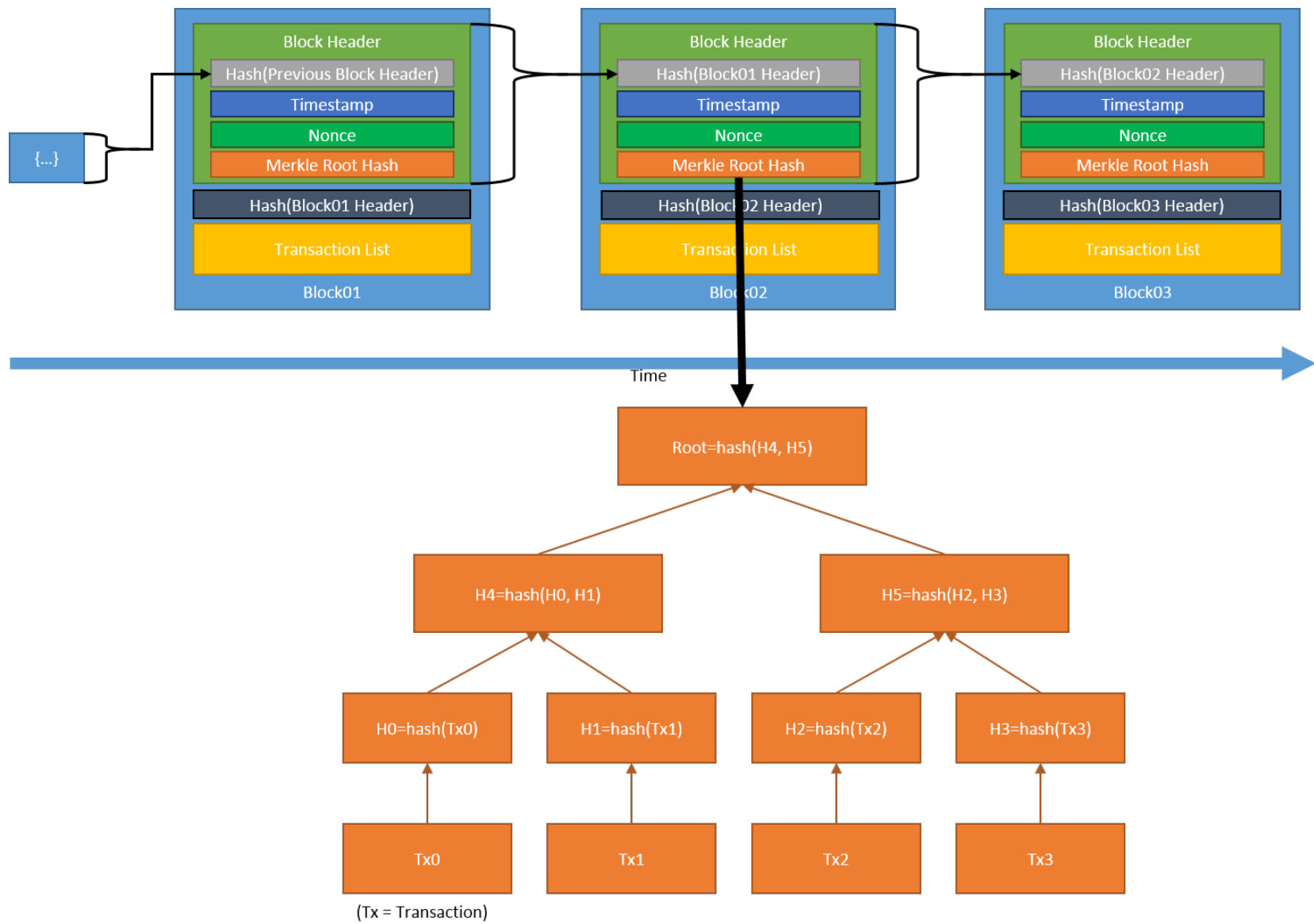
548

549

Figure 5: Example of a Merkle Tree

550 Figure 6 shows the relationship between a Merkle tree and a block. The bottom row of the tree
 551 contains blockchain transactions Tx0 through Tx3. The Merkle root is stored within the block
 552 header.

553 The entire block header is hashed; the block header hash value is stored within the block itself, as
 554 well as within in the next block, and this helps provide the immutability of transactions since the
 555 Merkle root hash will not match if any change is made to the transactions.



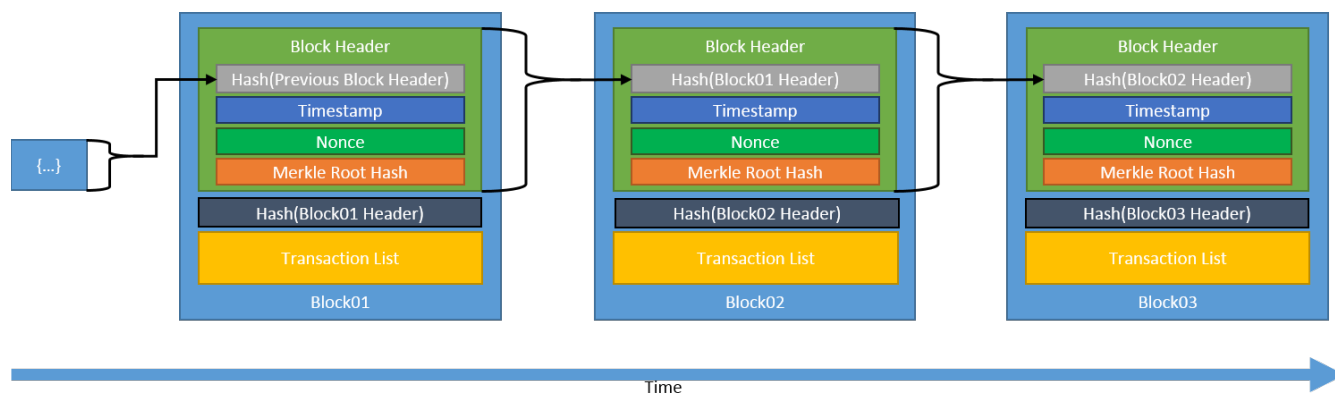
556

557

Figure 6: Blockchain with Merkle Tree

558 2.7 Chaining Blocks

559 Blocks are chained together through each block containing the hash of the previous block's
 560 header, thus forming the *blockchain*. If a previously published block were changed, it would
 561 have a different hash. This in turn would cause all subsequent blocks to also have different
 562 hashes since they include the hash of the previous block. This makes it possible to easily detect
 563 and reject any changes to previously published blocks. Figure 7 shows a generic chain of blocks.



564

565

Figure 7: Generic Chain of Blocks

566 3 Blockchains in Operation

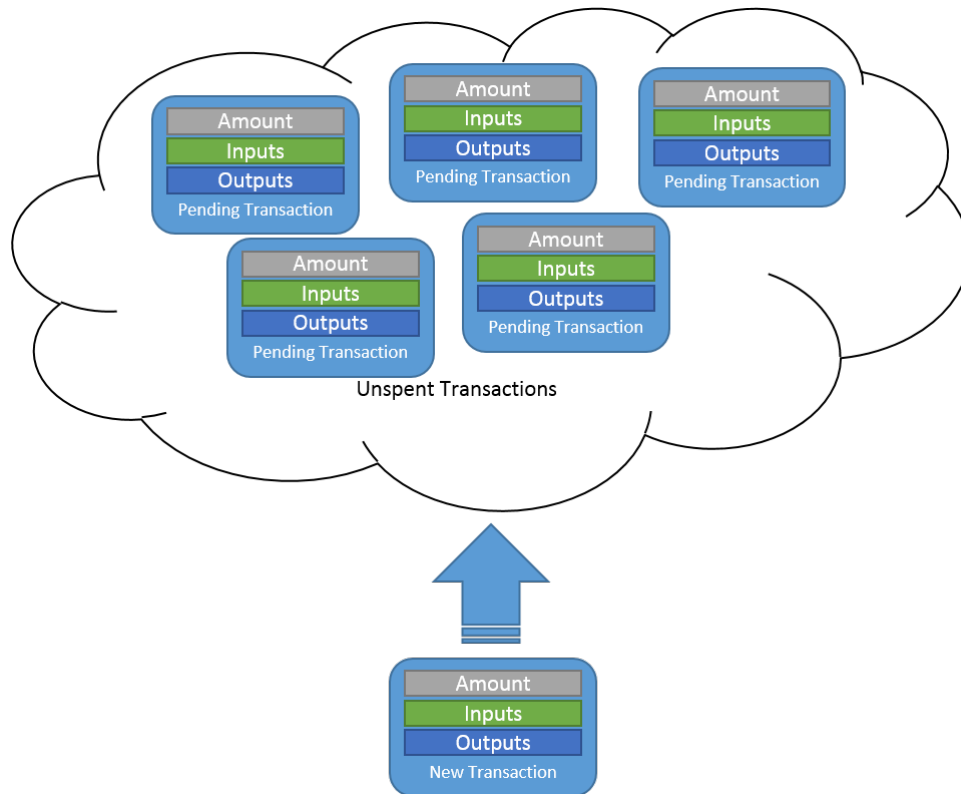
567 In the previous section, we provided a static view of the components of a generic blockchain. In
 568 this section, we discuss how a blockchain is expanded through the addition of new blocks
 569 representing sets of transactions. We discuss here a permissionless blockchain that utilizes the
 570 proof of work consensus method (the most popular method to date and the one used by Bitcoin
 571 and its derivatives). Information on other consensus methods is discussed in Section 4 below.

572 Blockchains are maintained through the consensus of a set of computers running blockchain
 573 software, known as mining nodes. There is no central authority determining which node
 574 publishes the next block on the blockchain. Each node maintains a copy of the blockchain and
 575 may propose a new block to the other mining nodes. Invalid blocks will be detected and rejected
 576 because it is difficult to compute a valid block, but computationally easy to verify one. Mining is
 577 an intentionally resource-intensive task, taking large amounts of processing power, memory, or
 578 both, depending on the specific blockchain application. The consensus protocol that determines
 579 which new block gets added to the blockchain is discussed in Section 4.

580 As mentioned earlier, any computer running blockchain software is considered a *node* of that
 581 blockchain. There are generally two types of nodes: full nodes and lightweight nodes. The job of
 582 a *full node* is to store the blockchain data, pass along the data to other nodes, and ensure newly
 583 added blocks are valid. Validation entails ensuring that the format of the block is correct, all
 584 hashes in the new block were computed correctly, the new block contains the hash of the
 585 previous block, and each transaction in the block is valid and signed by the appropriate parties.
 586 Full nodes may also act as mining nodes (i.e., generating new blocks). *Lightweight nodes* do not
 587 need to store full copies of the blockchain and often pass their data on to full nodes to be

588 processed. Lightweight nodes are generally found on smartphones and Internet of Things (IoT)
 589 devices—devices with limited computational and/or storage capability. Any node may propose
 590 new transactions, and these proposed transactions are propagated between nodes until they are
 591 eventually added to a block.

592 Proposed transactions within a blockchain system are stored on mining nodes within an *unspent*
 593 *transaction pool*—waiting to be included within a block as depicted in Figure 8.



594

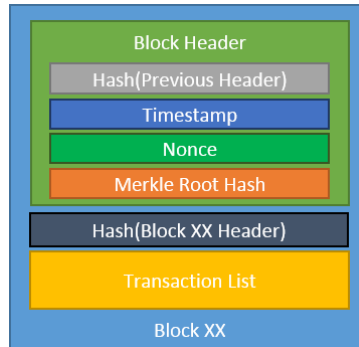
595

Figure 8: Transaction Being Added to Unspent Transaction Pool

596 When mining nodes put together a new candidate block, they include a set of unspent
 597 transactions. They may take a combination of older transactions that have been waiting for some
 598 time and newer transactions that offer a higher payment (in the form of a transaction fee paid by
 599 the user who submitted the transaction). The mining node checks that each transaction is itself
 600 valid since the other nodes would reject the block if it included invalid transactions. At this point,
 601 the mining node fills out all information required by the block structure discussed in Section 2.6,
 602 except the nonce.

603 Some blockchain systems require a form of sacrifice to create the next block – such as expending
 604 time and effort, or staking for the privilege. For systems, which require time and effort, the
 605 mining node calculates many random nonce values to attempt to solve a computationally
 606 difficult puzzle. The winning mining node gets the right to publish the next block (see Section
 607 4.1). Usually, mining nodes try many nonce values before solving a puzzle. Once a puzzle is
 608 solved with a particular nonce, the node creates a hash of the block’s data and stores it within the

609 block itself. Figure 9 depicts the high-level structure of the constructed block. The block is then
610 sent out to other nodes for verification; if everything is verified, the nodes accept it as the latest
611 block and continue to pass it along. Section 4.4 discusses what happens if multiple mining nodes
612 solve the challenge in the same timeframe, creating multiple competing 'next' blocks.



613

614

615

Figure 9: Finalized Block (Generalized)

616 4 Consensus

617 In our generic presentation of a blockchain from the previous section, many mining nodes are
 618 competing at the same time to solve a puzzle to gain the right of publishing the next block (and if
 619 applicable, a financial award). They are generally mutually distrusting users that may only know
 620 each other by their public addresses. Each user may be motivated by a desire for financial gain,
 621 not the well-being of the other mining nodes or even the network as a whole. In such a situation,
 622 why would a user propagate a block solved by another user? Also, who resolves conflicts when
 623 multiple mining nodes solve a block at approximately the same time? To make this work,
 624 blockchains use a variety of *consensus models* that enable a group of mutually distrusting users
 625 to work together.

626 Note that when a user joins a blockchain system, the user agrees to the initial state of the system.
 627 This is recorded in the only pre-configured block, the *genesis block*. Every blockchain has a
 628 published genesis block and every block must be added to the blockchain after it, based on an
 629 agreed-upon consensus method. Regardless of the method, however, each block must be valid
 630 and thus can be validated independently by each user in the blockchain network. By combining
 631 the initial state and the ability to verify every block since then, users can agree on the current
 632 state of the blockchain. Note that if there were ever two valid chains presented to a user, the
 633 default mechanism, in most blockchain systems, is that the longer chain is ‘more’ valid and
 634 should be adopted (this happens occasionally and will be discussed later).

635 The following properties are then in place:

- 636 • The initial state of the system is agreed upon.
- 637 • Users agree to the consensus method by which blocks are added to the system.
- 638 • Every block is linked to the previous block with a hash (except for the first ‘genesis’
 639 block, which has no previous block, and usually has a hash value of all 0’s for the
 640 previous block).
- 641 • Users can verify every block.

642 In practice, node software handles all the details. Key to the blockchain approach is that there is
 643 no need to have a trusted third party to give the state of the system—every user within the system
 644 can verify the system’s integrity. To add a new block to the blockchain, all participating nodes
 645 must come to a common agreement over time, however, so some temporary disagreement is
 646 permitted. The method of agreement (or consensus) must work even in the presence of possibly
 647 malicious users attempting to disrupt or take over the blockchain. This section discusses several
 648 major consensus models, as well as conflict resolution.

649 4.1 Proof of Work Consensus Model

650 In the *proof of work*² *model*, a user gets the right to publish the next block by solving a
 651 computationally intensive puzzle. The solution to this puzzle is the “proof” they have performed

² Proof of work is often abbreviated PoW.

652 work. The puzzle is designed such that solving the puzzle is difficult, but checking that a solution
 653 is valid is easy. This enables all other mining nodes to easily validate any proposed next blocks,
 654 and any proposed block that did not satisfy the puzzle would be rejected. A common puzzle
 655 method is to require that the hash of the block be less than a certain value. Mining nodes then
 656 make many small changes to the block (the nonce) trying to find a block hash that meets the
 657 requirement. For each attempt, the mining node must compute the hash for the entire block
 658 header, which is a computationally intensive process. The required value may be modified over
 659 time to adjust the difficulty to influence how often blocks are being published. For example,
 660 Bitcoin, which uses the proof of work model, adjusts the puzzle difficulty every two weeks to
 661 influence the block publication rate to be around once every ten minutes.

662 An important aspect of this model is that the past work put into a puzzle does not influence one's
 663 likelihood of solving future puzzles. Hashing a candidate block one thousand or one million
 664 times (with different nonce values) only increases the likelihood of solving the current puzzle (as
 665 the nonce input space is being reduced with each hash calculation), it does not increase the user's
 666 likelihood of solving any future puzzles, and therefore each puzzle to solve for a block is
 667 independent and requires the same amount of work. This means that when a user receives a
 668 completed block from another user, they are incentivized to include the new block because they
 669 know the other mining nodes will include it and start building off it. If they refuse to accept the
 670 new block, they will be building off a shorter chain of blocks and (as mentioned previously) by
 671 default, the longest valid chain is adopted.

672 As an example, consider a puzzle where, using the SHA-256 algorithm, a computer must find a
 673 hash value meeting the following target criteria:

674 `SHA256("blockchain" + Nonce) = Hash Value starting with "000000"`

675 In this example, the text string "blockchain" is appended with a nonce value and then the
 676 hash value is calculated. The nonce values used will be numeric values only. This is a relatively
 677 easy puzzle to solve and some sample output follows:

678 `SHA256("blockchain0") =`
 679 `0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab938`
 680 `(not solved)`

681 `SHA256("blockchain1") =`
 682 `0xdb0b9c1cb5e9c680dfff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10`
 683 `(not solved)`

684 ...

685 `SHA256("blockchain10730895") =`
 686 `0x000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d67587`
 687 `(solved)`

688 To solve this puzzle, it took 10,730,896 guesses (completed in 54 seconds on relatively old
 689 hardware, starting at 0 and testing one value at a time). However, each additional "leading zero"
 690 value increases the difficulty. By increasing the target by one additional leading zero
 691 ("000000"), the same hardware took 934,224,175 guesses to solve the puzzle (completed in 1
 692 hour, 18 minutes, 12 seconds):

```
693     SHA256("blockchain934224174") =
694     0x0000000e2ae7e4240df80692b7e586ea7a977eacbd031819d0e603257edb3a81
```

695 There is no shortcut to this process; mining nodes must expend computation effort, time, and
696 resources to find the correct nonce value for the target.

697 Once a user has performed this work, they send their block with a valid nonce to the other nodes
698 in the network. The recipient nodes verify that this work was done properly, add the block to
699 their copy of the blockchain, and resend the block to their peer nodes. In this manner, the new
700 block gets quickly distributed throughout the network of participating nodes. Verification of the
701 nonce is easy since only a single hash needs to be done to check to see if it solves the puzzle.

702 The proof of work consensus model is designed for the case where there is little to no trust
703 amongst users of the system. It ensures mining nodes cannot game the system³ by always being
704 able to solve the puzzles and thereby control the blockchain and the transactions added to it.
705 However, a major pitfall of the proof of work consensus model is its excessive use of energy in
706 solving the puzzles. This is not trivial; for example, currently the Bitcoin blockchain uses more
707 electricity than the entire country of Ireland, and it has been speculated that it will consume as
708 much electricity as the entire country of Denmark by 2020 [7][8][9]. Software and hardware
709 continually improve, with the result that puzzles can be solved more efficiently, but blockchain
710 networks are growing, and the puzzle targets get harder as more mining nodes participate.

711 Due to the increasing difficulty of the proof of work puzzles, it is becoming harder for any one
712 computer to solve a puzzle. Therefore, mining nodes have organized themselves into “pools” or
713 “collectives” whereby they collectively solve puzzles. This is because it is possible to distribute
714 the work between two or more nodes across a collective to share the workload and rewards.
715 Splitting the example program into quarters, each node can take an equal amount of the nonce
716 value range to test:

- 717 • Node 1: check nonce 0000000000 to 0536870911
- 718 • Node 2: check nonce 0536870912 to 1073741823
- 719 • Node 3: check nonce 1073741824 to 1610612735
- 720 • Node 4: check nonce 1610612736 to 2147483647

721 The following result was the first to be found to solve the puzzle:

```
722     SHA256("blockchain1700876653") =
723     0x00000003ba55d20c9cbd1b6fb34dd81c3553360ed918d07acf16dc9e75d7c7f1
```

724 This is a completely new nonce, but one that solved the puzzle. It took 90,263,918 guesses
725 (completed in 10 minutes, 14 seconds). Dividing up the work amongst many more machines
726 yields much better results, as well as more consistent rewards in a proof of work model.

³ Use the rules and procedures meant to protect the system to actually manipulate the system for a desired result.

727 4.2 Proof of Stake Consensus Model

728 The proof of stake model is based on the idea that the more stake⁴ a user has in the system, the
729 more likely it will want the system to succeed, and the less likely it will want to subvert it. Proof
730 of stake blockchain systems use the amount of stake a user has as a determining factor for new
731 block creation. The methods for how the blockchain system uses the stakes can vary – from
732 random selection of staked users, to multi-round voting, to a coin aging system. Regardless of
733 the exact approach, users with more stake are more likely to produce new blocks.

734 With this consensus model, there is no need to perform resource intensive computations (time,
735 electricity, processing power) as found in proof of work. Since this consensus method utilizes
736 less resources, some blockchains have decided to forego a reward for new block creation; these
737 systems are designed so that all the cryptocurrency is already distributed among users rather than
738 new coins being generated at a constant pace.

739 Within a proof of stake blockchain system, where the choice of block creator is a random choice
740 (sometimes referred to as *Chain-based proof of stake*), the blockchain system will look at all
741 users with stake and choose amongst them based on their stake to overall system stake ratio. So,
742 if a user had 42% of the stake they would be chosen 42% of the time; those with 1% would be
743 chosen 1% of the time.

744 When the choice of block creator is a multi-round voting system (sometime referred to as
745 *Byzantine Fault Tolerance proof of stake* [10]) there is added complexity. The blockchain system
746 will select several staked users to create proposed blocks. The system will then ask all staked
747 users to vote for the next block. After several rounds of this voting, a new block is decided upon.
748 This method allows all staked users to have a voice in the block selection process for every new
749 block.

750 Finally, there is a method of proof of stake which allows users to create blocks by “spending”
751 aged cryptocurrency (sometimes referred to as “Coin age” proof of stake). The user’s staked
752 cryptocurrency has an additional “age” property, and after a certain amount of time (such as 30
753 days) the staked cryptocurrency can be “spent” and allow the user to create a new block on the
754 blockchain. The “spent” cryptocurrency then has its “age” reset to 0, and it cannot be used again
755 until after the requisite time has passed. This method allows for users with more stake to create
756 more blocks, but to not dominate the system – since they have a cooldown timer attached to
757 every cryptocurrency spent creating blocks.

758 Under proof of stake systems, the “rich” can more easily stake more of the digital assets, earning
759 themselves more assets; however, to obtain the majority of assets within a system in order to
760 “control” it is generally cost prohibitive.

⁴ Stake is an amount of cryptocurrency that the user has invested into the system, either by locking it via a special transaction type, or by sending it to a specific address; the amount of staked cryptocurrency is generally no longer able to be spent. The likelihood of a user creating a new block is tied to the ratio of their stake to the overall blockchain system amount of staked cryptocurrency.

761 4.3 Round Robin Consensus Model

762 In some blockchain systems there does exist some level of trust between mining nodes. In this
763 case, there is no need for a complicated consensus mechanisms to determine which participant
764 adds the next block to the chain. This consensus model is often used for private blockchains and
765 is called *round robin*, where nodes take turns in creating blocks. To handle situations where a
766 mining node is not available when it is their turn, these systems may include an element of
767 randomness to enable available nodes to publish blocks so that unavailable nodes will not cause
768 a halt in block production. This model ensures no one node creates the majority of the blocks, it
769 benefits from a straightforward approach, it lacks cryptographic puzzles, and has low power
770 requirements.

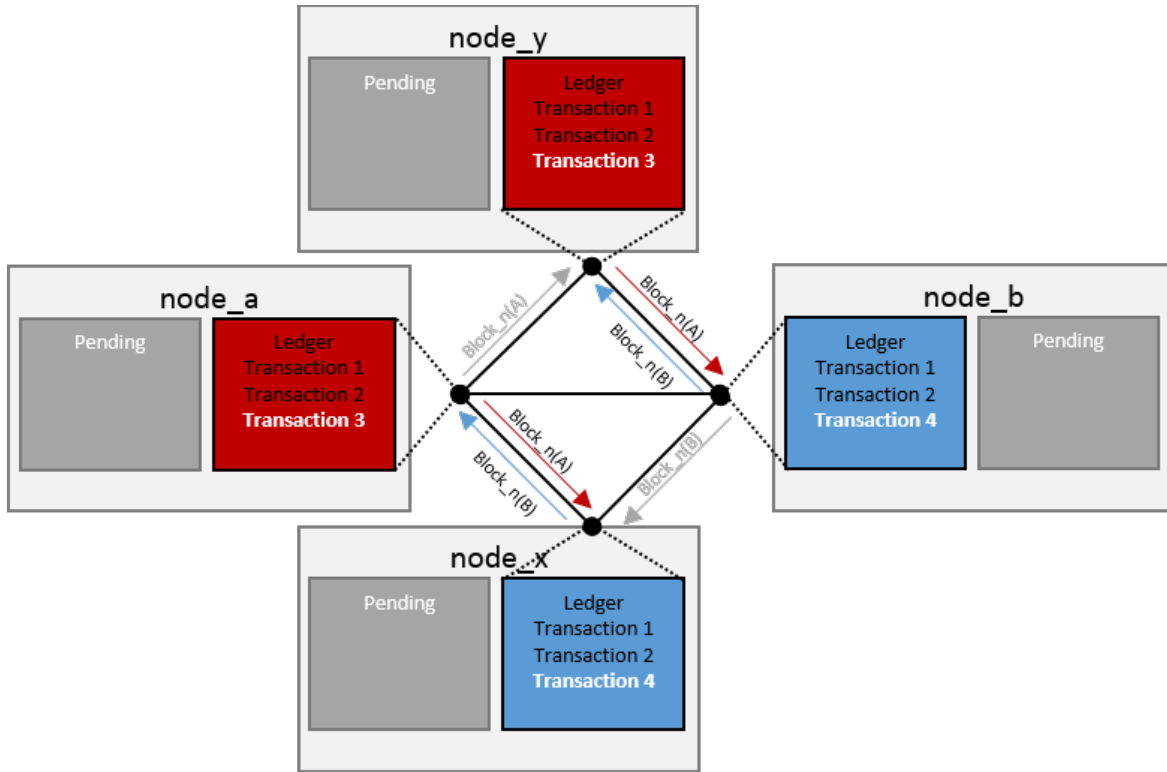
771 Unfortunately, due to the need for some level of trust amongst nodes, round robin does not work
772 well in the permissionless open networks used by most blockchain based cryptocurrencies
773 because malicious nodes can continuously add additional nodes to increase the odds of
774 subverting the network.

775 4.4 Ledger Conflicts and Resolutions

776 As discussed previously, it is possible that multiple blocks will be published at approximately the
777 same time. This can cause differing versions of a blockchain to exist at any given moment; these
778 must be resolved quickly in order to have consistency in the blockchain. In this section, we
779 discuss how these situations are handled.

780 With any distributed network, some systems within the network will be behind on information or
781 have alternative information. This depends on network latency between nodes and the proximity
782 of groups of nodes. Blockchain systems that allow any node to generate blocks are more prone to
783 have conflicts due to this openness. A major part of agreeing on the state of the blockchain
784 system (coming to consensus) is resolving conflicting data.

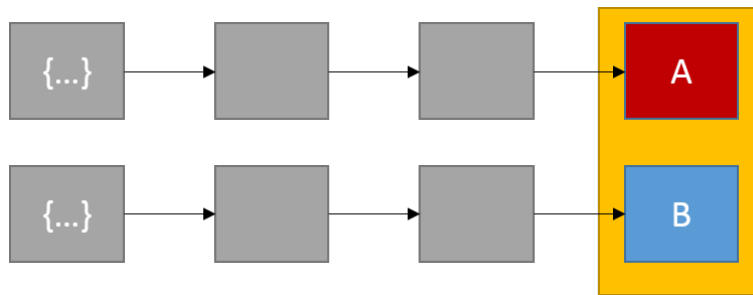
785 For example, if `node_A` creates `block_n(A)` and distributes it to some peers, and `node_B`
786 creates `block_n(B)` and distributes it to some peers, there will be a conflict. `block_n` will
787 not be the same across the network. This conflict is shown in Figure 11, `node_a`'s ledger is in
788 red, and `node_b`'s ledger is in blue; they each made `block_n`, but have different transactions
789 within them (`block_n(A)` contains Transaction 3, but not Transaction 4, while `block_n(B)`
790 contains Transaction 4 but not Transaction 3).



791
792

Figure 10: Distributed Network in Conflict

793 Conflicts temporarily generate different versions of the blockchain, which is depicted in Figure
 794 11. These differing versions are not “wrong”; rather, they were created with the information the
 795 node had available. The competing blocks will likely have differing transactions within the
 796 transaction list, so those with `block_n(A)` may see transfers of digital assets not present in
 797 `block_n(B)`. If the blockchain deals with digital currency, money may both be spent and
 798 unspent, depending on which version of the blockchain is being viewed.

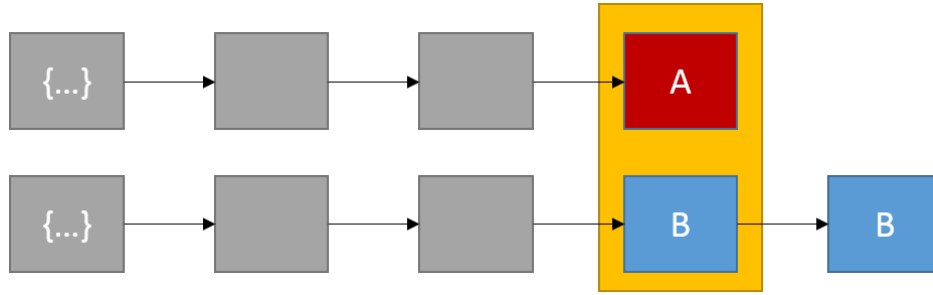


799
800

Figure 11: Blockchains in Conflict

801 Conflicts are usually quickly resolved. Most blockchain systems will wait until the next block is
 802 generated and use that chain as the “official” blockchain, thus adopting the “longer blockchain”.
 803 As in Figure 12, the blue blockchain becomes the “official” chain, as it got the next valid block.
 804 Any transaction that was present in the chain that was not selected, but not present in the new
 805 “official” chain, is returned to the unspent transaction pool. Note that this set of waiting

806 transactions is maintained locally at each node (there is no central server in the architecture).



807

808

Figure 12: Chain B Adds the Next Block

809 **5 Forking**

810 Updating technology can be difficult at the best of times, such as when systems are centralized.
811 When a system is comprised of many users, distributed around the world, and governed by the
812 consensus of the users, it becomes extremely difficult. Changes to the blockchain software and
813 implementation are called *forks*.

814 **5.1 Soft Forks**

815 A *soft fork* is a change to the technology that **will not** completely prevent users who do not adopt
816 the change (e.g., an update to the latest version) from using the changed blockchain system.
817 Since non-updated nodes will recognize the new blocks as valid, a soft fork can be backwards
818 compatible, only requiring that a *majority* of nodes upgrade to enforce the new soft fork rules.

819 An example of a soft fork occurred on Bitcoin when a new consensus rule was added to support
820 escrow⁵ and time-locked refunds. In 2014, a proposal was made to repurpose an operation code
821 that performed no operation (OP_NOP2) to CHECKLOCKTIMEVERIFY, which allows a
822 transaction output to be made unspendable at a point in the future [11]. For future clients that
823 implement this change, the blockchain interpreter will perform this new operation, but for clients
824 that do not support the change, the script is still valid, and execution will continue as if a NOP⁶
825 had been executed.

826 **5.2 Hard Forks**

827 A *hard fork* is a change to the technology that **will** completely prevent users who do not adopt it
828 from using the changed blockchain system. Under a hard fork, the blockchain protocol will
829 change in a manner that requires users to either upgrade to stay with the developer's "main fork"
830 or to continue on the original path without the upgrades. Users on different hard forks cannot
831 interact with one another. Any change to the block structure, such as the hashing algorithm
832 choice, will require a hard fork.

833 A well-known example of a hard fork is from Ethereum. In 2016, a smart contract was
834 constructed on Ethereum called the Decentralized Autonomous Organization (DAO). Due to
835 flaws in how the smart contract was constructed, an attacker extracted Ether, the cryptocurrency
836 used by Ethereum, essentially allowing theft of \$50 million [12]. A hard fork proposal was voted
837 on by Ether holders, and 89 percent agreed to hard fork and create a new version of the
838 blockchain that returned the stolen funds.

839 With cryptocurrencies, if there is a hard fork and the blockchain splits, the coins each person has
840 at the time of the split will be mirrored on each fork. If all the activity moves to the new chain,
841 the old one will eventually not be used. In the case of the Ethereum hard fork, the vast majority

⁵ Funds placed into a third party to be disseminated based on conditions (via multi-signature transactions)

⁶ NOP meaning No Operation

842 of support moved to the new fork, and the old fork was renamed to Ethereum Classic, which has
843 only a fraction of the original user base.

844 5.3 Cryptographic Changes and Forks

845 If flaws are found in the cryptographic technologies for a blockchain application, the only
846 solution may be to create a hard fork, depending on the significance of the flaw. For example, if
847 a flaw was found in the underlying algorithms, there could be a fork requiring all future clients to
848 use a stronger algorithm. Until more than 50 percent of the network is on the new software
849 version, the vulnerability could still exist. Switching to a new hashing algorithm could pose a
850 significant practical problem because it could invalidate all existing specialized mining
851 hardware.

852 Hypothetically, if SHA-256 were discovered to have a flaw, there would need to be a hard fork
853 to migrate to a new hash algorithm. The block that switches over to the new hash algorithm
854 would “lock” all previous blocks into SHA-256 (for verification), and all new blocks would need
855 to utilize the new hashing algorithm. For example, Bitcoin uses SHA-256 hashes, which is easy
856 and fast to implement in hardware ASICs. Other cryptocurrencies such as Ethereum use Keccak-
857 256 (based on SHA-3) [13], while Litecoin uses the scrypt hashing algorithm.

858 One possibility for the need to change cryptographic features present in a blockchain system
859 would be the development of a practical quantum computer system, which would be capable of
860 greatly weakening (and in some cases, rendering useless) existing cryptographic algorithms.
861 NIST Internal Report (NISTIR) 8105, Report on Post-Quantum Cryptography [14] provides a
862 table describing the impact of quantum computing on common cryptographic algorithms. Table
863 3 replicates this table.

864 **Table 3: Impact of Quantum Computing on Common Cryptographic Algorithms**

Cryptographic Algorithm	Type	Purpose	Impact from Large-Scale Quantum Computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	N/A	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

865 The cryptographic algorithms utilized within most blockchain technologies for public/private key
866 pairs will need to be replaced if a powerful quantum computer become a reality. This is because
867 algorithms that rely on the computational complexity of integer factorization (such as RSA) or
868 work on solving discrete logarithms (such as DSA and Diffie-Hellman) are very susceptible to
869 quantum computing. The hashing algorithms and Merkle trees that are the other basis for
870 blockchains are much less susceptible to quantum computing attacks, but are still weakened
871 when quantum computers become a reality.
872

873 6 Smart Contracts

874 A *smart contract* is a collection of code and data (sometimes referred to as functions and state)
875 that is deployed to a blockchain (e.g., Ethereum). Future transactions sent to the blockchain can
876 then send data to public methods offered by the smart contract. The contract executes the
877 appropriate method with the user provided data to perform a service. The code, being on the
878 blockchain, is immutable and therefore can be used (among other purposes) as a trusted third
879 party for financial transactions that are more complex than simply sending funds between
880 accounts. A smart contract can perform calculations, store information, and automatically send
881 funds to other accounts. It doesn't necessarily even have to perform a financial function. For
882 example, the authors of this document have created smart contracts that publicly generate
883 trustworthy random numbers [15].

884 In practice, all mining nodes execute the smart contract code simultaneously when mining new
885 blocks. Thus, smart contract execution may be more expensive than the simple fund transfers in
886 other blockchain based cryptocurrencies. Often, the user issuing a transaction to a smart contract
887 will have to pay for the cost of the code execution in addition to the normal transaction fees.
888 There is a limit on how much execution time can be consumed by a call to a smart contract. If
889 this limit is exceeded, execution stops and the transaction is discarded. This mechanism not only
890 rewards the miners for executing the smart contract code, but also prevents malicious users from
891 deploying and then accessing smart contracts that will perform a denial of service on the mining
892 nodes by consuming all resources (e.g., using infinite loops).

893 **7 Blockchain Categorization**

894 Blockchains are generally categorized based on the permission model, which determines who
895 can access them. If anyone can read and write to a blockchain, it is *permissionless*. If only
896 particular users can read and write to it, it is *permissioned*. In simple terms, a permissioned
897 blockchain is similar to a corporate intranet that is controlled, while a permissionless blockchain
898 is like the public internet, where anyone can participate.

899 **7.1 Permissioned**

900 Permissioned blockchains defy the original conception of the Bitcoin blockchain where everyone
901 can read and write to the blockchain, and the ledger is transparent/public. Organizations that
902 wish to work together, but do not fully trust one another, can establish a permissioned blockchain
903 and invite business partners to record their transactions on a shared distributed ledger. This
904 permissioned blockchain can have the same traceability of assets as they pass through the
905 blockchain, as well as the same distributed, resilient, and redundant data storage system as a
906 permissionless blockchain. These organizations can determine the consensus mechanism to be
907 used, based on how much they trust one another.

908 Permissioned blockchains can be set up so anyone can read them, but only selected members can
909 record transactions on them. This type of blockchain would provide full insight into the internal
910 interactions of the organization by anyone who has an interest, but the public at large would not
911 be able to interfere with the data. Permissioned blockchains can also be set up so anyone can
912 record transactions on the blockchain, but only selected members can read the data.

913 **7.1.1 Application Considerations for Permissioned Blockchains**

914 While permissioned blockchains are often considered an improvement over current systems,
915 certain design characteristics must be considered carefully to ensure security. For example, when
916 using a database, it is possible to have detailed permission granularity, such as allowing certain
917 users to perform specific queries or only allowing certain users to write to specific tables.
918 Applications that use a blockchain may need to consider whether the permissions supported by
919 the blockchain are granular enough to permit enough roles to be created within the system
920 (permissions allow for more traditional roles, such as administrator, user, validator, auditor, etc.).
921 This also applies to how permissions are administered. Once a user is given write access to the
922 blockchain, is it possible to revoke that permission? Most blockchain implementations are
923 immutable, which can make permissions more complicated.

924 Trust is another critical consideration when deciding to build an application on a blockchain.
925 Within a permissioned blockchain system the method of consensus is generally less
926 computationally intensive – therefore it could be possible for users to act maliciously. However,
927 the trust does not need to extend to all users. It is possible for the maintainer of the blockchain to
928 designate a limited set of mining nodes. If these are trustworthy, it is then unnecessary for the
929 user population at large to be trustworthy as the miners will enforce the blockchain rules.

930 Another important consideration is having a tamper-evident design. If a malicious mining node
931 tried to change a block, they might for example forge a transaction to give themselves money.
932 Would this kind of change be detected? Are there systems in place to determine what happened?

933 Immutability is important and is one of the founding principles of the blockchain. In general,
934 malicious transactions that enter the blockchain cannot be undone, even if they are identified. To
935 do so requires rewriting published blocks which essentially forks the blockchain and requires the
936 approval of the majority of mining nodes. In a permissioned system this can be easier since the
937 mining nodes are generally a trusted set that have a special relationship. It is much more difficult,
938 but technically possible, for a permissionless systems such as Bitcoin.

939 **7.1.2 Use Case Examples**

940 The following sections depict examples of use cases (not an exhaustive list). Inclusion or
941 exclusion from this section does not validate or invalidate any potential use case.

942 **Banking**

943 Suppose a number of banks want to keep a private, distributed ledger available to only the
944 participating banks. This would provide the ability to record transactions from each bank in a
945 way that is visible to the participants, but not the public. However, to do this as a private
946 blockchain (to avoid having to use an expensive proof of work algorithm), each bank takes turns
947 signing the blocks under a distributed consensus algorithm such as Byzantine Paxos [16].

948 There are a few interesting considerations when using a private blockchain with few participants,
949 such as the ability to overcome its immutability. If there was some major disaster or exception
950 situation, the banks could coordinate to roll back the blockchain and write a different transaction.
951 Additionally, the transactions would not be anonymous because a banking ID would be needed
952 to join.

953 **Supply Chain**

954 Recording the transfer of physical goods from a producer, to a shipping terminal, to a ship, to a
955 cargo train, to a delivery truck and to a store is an appealing application of blockchain
956 technology. A blockchain could play a crucial role in trust and transparency with end customers.
957 The blockchain could also be used to monitor supplier actions. Suppliers can record the product
958 produced (such as X number of widgets on a certain date) in a way that other viewers of the
959 blockchain can verify. With a blockchain, it is possible for warehouses to manage logistics
960 efficiently by avoiding overstocking.

961 **Insurance and Healthcare**

962 Whenever someone visits a care provider, a myriad of transactions take place behind the scenes.
963 Administrative transactions from nurses, doctors, staff, medical providers, insurance companies,
964 and pharmacies could all be written to a blockchain. Transactions (such as checking benefits,
965 eligibility, coverage, and the available medicine supply) could be read from the blockchain.

966 Currently, records of these transactions reside in disparate systems, sharing results at the end of
967 an (often manual) process.

968 **7.2 Permissionless**

969 Permissionless blockchains are decentralized platforms with no central authority and are open to
970 participation without users requesting access. Permissionless blockchains often utilize a
971 consensus method that requires more than a trivial effort in order to prevent bad users from
972 easily subverting the system. Such consensus methods include proof of work and proof of stake
973 methods. The reason a permissionless blockchain can work is because there are rewards for
974 participating in the process.

975 **7.2.1 Application Considerations for Permissionless Blockchains**

976 When deciding whether to utilize a permissionless blockchain, one must consider whether the
977 application needs the following qualities:

- 978 • **Public facing data** – Since permissionless ledgers tend to allow anyone to inspect and
979 contribute to the blockchain, the data is generally public. Does the data for the application
980 need to be available to everyone? Is there any harm to having public data?
- 981 • **Full transactional history** – Due to the open nature of data for these systems, anyone
982 can track the transfer of assets between accounts, from the creation of assets, to each
983 transaction in progress.
- 984 • **False data attempts** – Since anyone could contribute to the blockchain, some could
985 submit false data to the blockchain, mimicking data from valid sources. Is there a way for
986 the application to ensure it only gathers data from reputable sources?
- 987 • **Data immutability** – Many applications follow the “CRUD” (create, read, update,
988 delete) functions for data. With a blockchain, there is only “CR” (create, read). There are
989 methods that can be employed to “deprecate” older data if a newer version is found, but
990 there is no removal process for the original data. Can the application handle (possibly
991 outdated) immutable data? Does the data lend itself to being immutable?
- 992 • **Transactional throughput capacity** – Currently transactions on blockchains are not
993 conducted at the same pace as other solutions (e.g., blocks are not added quickly enough),
994 so some slowdown while waiting for data to be posted may be incurred. Can the
995 application handle that?

996 **7.2.2 Use Case Examples**

997 The following sections depict examples of use cases (not an exhaustive list). Inclusion or
998 exclusion from this section does not validate or invalidate any potential use case.

999 **Trusted Timestamping**

1000 Trusted timestamping is a way to prove that certain information existed at a given point [17].
1001 The use of a blockchain allows a party to prove they had access to a piece of data in a way that
1002 cannot be repudiated. For example, if a person wanted to prove they had possession of a file,

1003 they could hash the file and record the hash value as an annotation to a transaction. Then, if he or
1004 she ever needs to prove possession of the file, it is recorded publicly.

1005 Other use cases of leveraging timestamping on a blockchain include proving a task was
1006 completed on a certain date, proving possession of a photo, proving a contract was signed, or
1007 proving events occurred.

1008 **Energy Industry**

1009 Another blockchain application is the recording of autonomous, machine-to-machine
1010 transactions regarding electricity use [18]. This would take advantage of digital platform
1011 opportunities and changing business models for tracing transactions on the smart grid. One
1012 notable use case in the energy industry for the blockchain is in recording certificates. There are
1013 different power plants generating energy and creating certificates that attest to the amount of
1014 energy produced for subsequent exchange. Currently, there are problems such as emission
1015 certificates being spent twice, as well as the need to address regulatory challenges and provide
1016 more uniform access for everybody in the market. A blockchain can effectively track the
1017 issuance and spending of these energy certificates.

1018 Another example of how blockchains are applicable in the energy industry is in the trading of
1019 excess renewable energy. Buildings can be wired with devices measuring energy usage and
1020 recording it to a blockchain, enabling excess energy to be sold and bought on a market.

1021 **8 Blockchain Platforms**

1022 Many blockchains are in use today, primarily for digital cash solutions. This section discusses a
1023 selection of blockchain platforms to highlight the technical differences and approaches being
1024 used. This is not an endorsement of any of these platforms, nor should it be construed as a list of
1025 the most popular or important platforms.⁷

1026 **8.1 Cryptocurrencies**

1027 Numerous applications of blockchain technologies are primarily oriented around moving
1028 currency from one account to another. This section profiles several examples of such blockchain
1029 applications.

1030 **8.1.1 Bitcoin (BTC)**

1031 Bitcoin is a digital cash system that has been previously discussed as the pioneer in using a
1032 blockchain. New blocks are created approximately once every 10 minutes using SHA-256
1033 hashing to link them together. It is a proof of work system where mining nodes must find a
1034 nonce to include in their block such that the hash of the block is less than some predetermined
1035 difficulty value. The difficulty is adjusted up or down to attempt to achieve the 10-minute target
1036 for block creation. Early in Bitcoin's history, individual computers could mine and publish
1037 blocks; currently Bitcoin requires specialized hardware, large datacenters, or many individuals
1038 working together in a mining pool to win the competition to publish blocks.

1039 With Bitcoin, the paying of transaction fees is technically optional since the mining nodes get
1040 most of their funds through the publication of blocks. This fee is designed to be a small fee for
1041 each transaction, but it can and has become large due to a substantial backlog of pending
1042 transactions. Paying a higher transaction fee can give a transaction a greater priority for getting
1043 added to the blockchain. Initially, mining nodes got 50 Bitcoin for each block, and only half of
1044 that after a certain number of blocks. For example, the reward for mining a block was 12.5
1045 Bitcoins in July 2016. Per the Bitcoin protocol, this reward will halve every 210,000 blocks
1046 (around four years) and will decrease to zero once 21 million Bitcoins have been produced [19].
1047 Bitcoin mining will continue at that point, but the reward will be completely derived from
1048 transaction fees.

1049 One last technical note of interest is that each Bitcoin transaction contains code written in a
1050 language called Script. This code represents a simple program that specifies the transaction. It
1051 contains no loops and is highly restricted with regards to functionality (i.e., it is not Turing
1052 complete⁸). Bitcoin transactions today use only a small portion of the available features of Script.
1053 In practice, most Bitcoin transactions use one of just a few templates of code for the movement
1054 of funds between parties.

⁷ The website *Map of Coins* (<http://mapofcoins.com/>) provides a good example of a number of blockchain systems, but is still far from being a complete listing

⁸ A Turing complete system (computer system, programming language, etc.) can be used for any algorithm, regardless of complexity, to find a solution.

1055 **8.1.2 Bitcoin Cash (BCC)**

1056 In July 2017, approximately 80 to 90 percent of the Bitcoin computing power voted to
1057 incorporate Segregated Witness (SegWit, where transactions are split into two segments:
1058 transactional data, and signature data), which made it possible to reduce the amount of data being
1059 verified in each block. Signature data can account for up to 65 percent of a transaction block, so
1060 a change in how signatures are implemented could be useful. When SegWit was activated, it
1061 caused a hard fork, and all the mining nodes and users who did not want to change started calling
1062 the original Bitcoin blockchain Bitcoin Cash (BCC). Technically, Bitcoin is a fork and Bitcoin
1063 Cash is the original blockchain. When the hard fork occurred, people had access to the same
1064 amount of coins on Bitcoin and Bitcoin Cash.

1065 **8.1.3 Litecoin (LTC)**

1066 Litecoin is inspired by and is very similar to Bitcoin, but aims to provide faster confirmation
1067 times. Litecoin has implemented SegWit, splitting transactions into two segments and hiding an
1068 increased block size [20]. The “witness” signature is separated from the Merkle tree. Another
1069 difference between Bitcoin and Litecoin is Litecoin uses the Scrypt algorithm for hashing instead
1070 of SHA-256. The Scrypt algorithm is more difficult to solve than SHA-256 because it uses more
1071 memory, which makes development of custom application-specific integrated circuits (ASICs)
1072 more difficult. There is a larger maximum number of coins which can be mined (84 million
1073 Litecoins). Litecoin is a complement to Bitcoin, with higher transaction volumes, and not
1074 designed to replace it [21].

1075 **8.1.4 Ethereum (ETH)**

1076 Ethereum is a blockchain platform focused on providing smart contracts. Smart contracts are
1077 programs that exist on the blockchain that can be accessed by Ethereum users. They can both
1078 receive and send funds while performing arbitrary computation. A properly designed contract
1079 can act as a trusted third party in financial transactions since its code is both public and
1080 immutable. Ethereum’s transaction programming language is Turing complete. Mining nodes
1081 receive funds through mining and transaction fees.

1082 Ethereum also has a concept called “gas” used to power the transactional computations (and is
1083 generally around 1/100,000 of an Ether). Every transaction consumes gas as it executes, and the
1084 originator of a particular transaction must pay sufficient gas, or the execution of the transaction
1085 aborts. There is a maximum gas limit per smart contract (currently three million gas) to prevent
1086 computationally expensive programs from being submitted to the Ethereum mining nodes. This
1087 is because all mining nodes must execute the transactions in parallel [22].

1088 The submission of a transaction to an Ethereum contract causes a program to be run in parallel
1089 on the mining nodes’ computers. The resulting state of the contract is stored on the blockchain
1090 by the user that publishes the next block.

1091 **8.1.5 Ethereum Classic (ETC)**

1092 Ethereum Classic was created when Ethereum hard forked after the DAO hack [12]. An attacker
1093 had drained approximately \$50 million, and the Ethereum Foundation created a hard fork to

1094 move the stolen funds back to a state before the attack took place. Users who owned Ethereum
1095 before the DAO hard fork had the same amount of Ethereum Classic (ETC) after the fork. The
1096 reason it exists is because a number of users of the Ethereum blockchain rejected the fork for
1097 philosophical reasons [23], including the principle that the blockchain cannot be changed, and
1098 decided to keep using the unforked Ethereum blockchain. The mining and software is largely the
1099 same between Ethereum and Ethereum Classic, with the difference being that Ethereum is a fork
1100 and the more popular chain.

1101 **8.1.6 Dash (DASH)**

1102 Dash is a cryptocurrency built with the objective of providing faster transactions. It uses a
1103 “masternode” network and can make transactions within four seconds [24]. Dash uses
1104 deterministic ordering for the masternodes by using the hash and proof of work for each block.
1105 Interestingly, becoming a masternode requires 1000 Dash collateral, which makes it very
1106 expensive (nearly impossible) to control more than 50 percent of the network [25]. The collateral
1107 requirement for masternodes seeks to alleviate the problems of untrusted nodes in a peer-to-peer
1108 network.

1109 Dash uses a different hashing algorithm than most, x11. This consists of using all 11 SHA-3
1110 contestant algorithms (including BLAKE, JH, Keccak, and Skein), with each hash being
1111 submitted to the next algorithm in the chain [25]. The reasoning is that multiple algorithm use
1112 makes it harder for an ASIC to be created that targets solving these hashes in hardware.

1113 **8.1.7 Ripple (XRP)**

1114 Ripple is the name of both a cryptocurrency and the payment network on which it is transferred.
1115 The goal of Ripple is to build on the approach of Bitcoin and to connect different payment
1116 systems together. It has a fixed supply of 100 billion XRP, with half of them designated for
1117 circulation [26]. Ripple clients do not need to download the entire blockchain, making it easy for
1118 clients to join in seconds. Additionally, there is no mining reward for running a server because
1119 each transaction costs a small amount of Ripple, similar to Ethereum gas. Therefore, there are no
1120 mining nodes or mining pools; instead, about one-thousandth of a cent from each transaction is
1121 destroyed [27]. Ripple is not designed with explicit goals for anonymity, but it does have features
1122 providing privacy, such as using proxied gateway payments.

1123 **8.2 Hyperledger**

1124 Hyperledger is a group of projects aiming to create enterprise-grade, open-source distributed
1125 ledgers [28]. The Hyperledger Project is supported and hosted by the Linux Foundation.
1126 Although hosted by the Linux Foundation, each project was developed and contributed by
1127 different sources. There are several projects within the Hyperledger Project, each one providing a
1128 blockchain platform to solve specific problems.

1129 **8.2.1 Hyperledger Fabric**

1130 This is a modular, permissioned blockchain that can run smart contracts (called chaincode) [29].
1131 The Fabric blockchain was initially contributed to the Hyperledger Project by Digital Asset and
1132 IBM.

1133 **8.2.2 Hyperledger Sawtooth**

1134 This is a modular distributed ledger using proof of elapsed time as the consensus protocol. In a
1135 *proof of elapsed time system*, every participant requests a “wait time” from a hardware enclave (a
1136 trusted and secure feature available on some hardware), which distributes wait times randomly.
1137 Whichever participant was awarded the shortest time creates the next block in the chain. The use
1138 of Hyperledger Sawtooth is tightly coupled to hardware that supports the hardware enclave
1139 feature. Hyperledger Sawtooth was initially contributed by Intel.

1140 **8.2.3 Hyperledger Iroha**

1141 This acts as an Identity/Know Your Customer (KYC) service using blockchain technologies,
1142 which allows institutions to share data and manage identity. Hyperledger Iroha was initially
1143 contributed by Soramitsu, Hitachi, NTT Data, and Colu.

1144 **8.2.4 Hyperledger Burrow**

1145 Hyperledger Burrow is a permissioned smart contract-enabled blockchain platform. It accepts
1146 Ethereum-based smart contract code. Hyperledger Burrow was originally contributed by Monax
1147 and co-sponsored by Intel.

1148 **8.2.5 Hyperledger Indy**

1149 This is an independent identity platform providing provenance for trust transactions and
1150 accountability. It supports user-controlled exchanges of verifiable claims about identifying
1151 information, as well as revocation models. It supports three important privacy features:
1152 Decentralized Identifiers (DIDs), pointers to off-ledger sources so that no personal data is written
1153 to the ledger, and zero-knowledge-proofs. The Indy code is being contributed to the Hyperledger
1154 Project by the Sovrin Foundation.

1155 **8.3 MultiChain**

1156 MultiChain is an open source blockchain platform that enables anyone to setup, configure, and
1157 deploy a private, semi-private, or public blockchain. MultiChain is a fork of Bitcoin, but with
1158 many modifications. Users can determine whether there is to be an associated cryptocurrency, as
1159 well as the consensus method (round robin or proof of work). In the default configuration,
1160 MultiChain is a private, permissioned-based blockchain using round-robin consensus. This
1161 means that the first person to set up the blockchain acts as an administrator and initial node; all
1162 additional users must direct their MultiChain blockchain clients to this first node, and the
1163 administrator must grant them permissions.

1164 MultiChain Streams [30] are a unique feature; they are described as “shared immutable key-
1165 value time series databases” which are stored on a blockchain.

1166 **9 Blockchain Limitations and Misconceptions**

1167 There is a tendency to overhype and overuse most nascent technology. Many projects will
1168 attempt to incorporate the technology, even if it is unnecessary. This stems from the technology
1169 being relatively new and not well understood, or the technology being surrounded by
1170 misconceptions. Blockchain technology has not been immune. This section highlights some of
1171 the limitations and misconceptions of blockchain technology.

1172 **9.1 Blockchain Control**

1173 A common misconception is that permissionless blockchains are systems without control and
1174 ownership. The phrase “no one controls a blockchain!” is often exclaimed; however, while no
1175 user, government, or country controls a blockchain, there is still a group of core developers who
1176 are responsible for the system’s development. These developers may act in the interest of the
1177 community at large, but they still maintain some level of control. For example, in 2013 Bitcoin
1178 developers released a new version of the most popular Bitcoin client which introduced a flaw
1179 and started two competing chains-of-blocks. The developers had to decide to either keep the new
1180 version (which had not yet been adopted by everyone) or revert to the old version [31]. Either
1181 choice would result in one chain being discarded—and some people’s monetary transactions
1182 becoming invalid.

1183 The developers made a choice, reverted to the old version, and successfully controlled the
1184 progress of the Bitcoin blockchain. This example was an unintentional fork; however, developers
1185 can purposely build new clients, and with enough adoption from the user base, a successful fork
1186 can be created. These forks are often discussed at length and given a long adoption period before
1187 being made mandatory to continue recording transactions on the new “main” fork.

1188 The phrase “no one controls a blockchain!” would be better stated as, “no one controls with
1189 whom and when you can perform transactions, within the rules of the blockchain system.”

1190 **9.2 Malicious Users**

1191 While the blockchain system can enforce transaction rules and specifications, it cannot enforce a
1192 code of conduct. This is problematic in permissionless blockchain systems, since users are
1193 pseudonymous and there is not a one-to-one mapping between blockchain nodes and users of the
1194 system. Permissionless blockchains provide incentive (e.g., a cryptocurrency) to motivate users
1195 to act fairly; however, some may choose to act maliciously if that provides greater incentives.
1196 The largest problem for malicious users is getting enough power (be it a stake in the system,
1197 processing power, etc.) to cause damage. Once a large enough malicious collusion is created,
1198 malicious mining actions can include:

- 1199 • Ignoring transactions from specific users, nodes, or even entire countries.
- 1200 • Creating an altered, alternative chain in secret, then submitting it once the alternative
1201 chain is longer than the real chain. The honest nodes will switch to the chain that has the
1202 most “work” done (per the blockchain protocol). This could attack the concept of
1203 “immutability” within a blockchain system [32].

- 1204 • Refusing to transmit blocks to other nodes, essentially disrupting the distribution of
1205 information.

1206 While malicious users can be annoyances and create short-term harm, blockchains can perform
1207 hard forks to combat them. Whether damages done (money lost) would be reversed would be up
1208 to the developers and users of the blockchain system.

1209 **9.3 No Trust**

1210 Another common misinterpretation comes from people hearing that there is no “trusted third
1211 party” in a blockchain and assuming blockchain systems are “trustless” environments. While
1212 there is no trusted third party certifying transactions in permissionless blockchain systems (in
1213 permissioned systems it is less clear, as administrators of those systems act as an administrator of
1214 trust by granting users admission and permissions), there is still a great deal of trust needed to
1215 work within a blockchain system:

- 1216 • There is trust in the cryptographic technologies utilized. For example, cryptographic
1217 algorithms or implementations can have flaws, and smart contracts can have unintended
1218 loopholes and flaws.
- 1219 • There is trust in the developers of the software to produce software that is as bug-free as
1220 possible.
- 1221 • There is trust that most users of the blockchain are not colluding in secret. If a single
1222 group or individual can control more than 50 percent of all block creation power, it is
1223 possible to subvert a permissionless blockchain system. However, generally obtaining the
1224 necessary computational power is prohibitively expensive.
- 1225 • There is trust that nodes are accepting and processing transactions fairly.

1226 **9.4 Resource Usage**

1227 Blockchain technology has enabled a worldwide network of value where every transaction is
1228 verified and the blockchain is kept in sync amongst a multitude of users. For blockchain systems
1229 utilizing proof of work, this means there is a large number of users churning away processing
1230 time and consuming a lot of electricity. A proof of work method is a great solution for “hard to
1231 create, easy to verify” proofs, but as discussed in Section 4.1, it requires significant resource
1232 usage.

1233 An additional strain on resources occurs whenever a new full node is created; the node must
1234 obtain (usually through downloading) most of or all the blockchain data (Bitcoin’s blockchain
1235 data is over 100 gigabytes in size as of this writing) [33]. This process uses a lot of network
1236 bandwidth.

1237 Blockchains are often compared to databases, and while they both store information, blockchains
1238 have limits on the amount of data that can be stored and are not meant to be a general storage
1239 medium. In order to quickly calculate hashes on transactions and distribute transactions amongst
1240 the network, transactions need to be relatively small. Large amounts of data are usually stored
1241 “off chain,” with “pointers/references” or hashes of the data stored within the blockchain itself.
1242 Blockchains also benefit from data being immutable, which is not a trait general purpose data
1243 usually needs.

1244 **9.5 Transfer of Burden of Credential Storage to Users**

1245 Since blockchains are not centralized, there is no intrinsic central place for user key
1246 management. Users must manage their own private keys, meaning if one is lost, anything related
1247 to that private key is lost (digital assets, etc.). There is no “forgot my password” or “recover my
1248 account” feature for blockchain systems. While centralized management solutions can be put
1249 into place, they create the same problems current systems have: central points of failure.

1250 **9.6 Private/Public Key Infrastructure and Identity**

1251 Some people, when hearing that blockchain technology incorporates a public/private key
1252 infrastructure, immediately believe it intrinsically supports identity. This is not the case, as there
1253 is not a one-to-one relationship of private key pairs to users (a user can have multiple private
1254 keys), nor is there a one-to-one relationship between blockchain addresses and public keys
1255 (multiple addresses can be derived from a single public key). Nodes on the Bitcoin blockchain
1256 validate transactions before they are added to a block and subsequently incorporated into the
1257 blockchain. One stage of this validation requires the user that initiated the transaction to sign the
1258 transaction with a private key. Blockchain nodes verify the signature to prove the user does in
1259 fact own the Bitcoin value being transferred.

1260 Digital signatures are often used to prove identity in the cybersecurity world, and this can lead to
1261 confusion about the potential application of a blockchain to identity management. A
1262 blockchain’s transaction signature verification process links transactions to the owners of private
1263 keys, but provides no facility for associating real-world identities with these owners. In some
1264 cases, it is possible to connect real-world identities with private keys, but these connections are
1265 made through processes outside, and not explicitly supported by, the blockchain. For example, a
1266 law enforcement agency could request records, from an exchange, that would connect
1267 transactions to specific individuals. Another example is an individual posting an address online
1268 for donations.

1269 While it is possible to use blockchains in identity management frameworks that require a
1270 distributed ledger component, it is important to understand that typical blockchain
1271 implementations are not designed to serve as standalone identity management systems. There is
1272 more to having secure digital identities than simply implementing a blockchain.

1273

10 Conclusions

1274
1275 Blockchains are a significant new avenue for technological advancements, enabling secure
1276 transactions without the need for a central authority. Starting in 2009, with Bitcoin leveraging
1277 blockchain technology, there has been an increasing number of blockchain based
1278 cryptocurrencies. Possibly more importantly, new applications beyond the realm of currencies
1279 are building upon the fundamentals of blockchain technology.

1280 The first applications were digital currencies with the distribution of a global ledger containing
1281 all transactions. These transactions are secured with cryptographic hashes, and transactions are
1282 signed and verified using public/private key pairs. The transaction history is summarized with
1283 Merkle trees, to efficiently and securely record a chain of events in a way that any attempt to edit
1284 or change a past transaction will also require a recalculation of all subsequent blocks of
1285 transactions.

1286 The use of blockchains is still in its early stages, but it is built on widely understood and sound
1287 cryptographic principles. Moving forward, it is likely that blockchains will be another tool that
1288 can be used to solve newer sets of problems. Financial organizations are likely to be the
1289 businesses most impacted by blockchains. They may need to adapt or even completely change
1290 their practices to focus on being platforms for value exchange and not just places to store value.

1291 Blockchains are also digitizing assets other than money. Companies that need to maintain a
1292 public record, such as holding land title, marriage, or birth records, should consider how their
1293 problem sets might be addressed by blockchain technologies. Blockchains also have strong
1294 potential for storing and recording supply chain records. A blockchain can record each step in a
1295 product's life, from when it was created in a factory, to when it was shipped and subsequently
1296 delivered to a store, and finally to when a consumer purchased it. There may even be new
1297 industries, such as digital notaries who can prove a person had access to a specific piece of
1298 information by recording the hash of it into the blockchain. There are many potential uses and
1299 opportunities for blockchain technologies.

1300 As detailed throughout this publication, a blockchain relies on existing network, cryptographic,
1301 and recordkeeping technologies but uses them in a new manner. It will be important that
1302 organizations are able to look at the technologies and both the advantages and disadvantages of
1303 using them. Once a blockchain is implemented and widely adopted, it becomes very difficult to
1304 change it without forking. Once something is recorded in a blockchain, it is usually there forever,
1305 even when there is a mistake. For some organizations these are desirable features. For others,
1306 these may be deal breakers preventing the adoption of blockchain.

1307 Blockchain technologies have the power to disrupt many industries. To avoid missed
1308 opportunities and undesirable surprises, organizations should start investigating whether or not a
1309 blockchain can help them.

1310 **Appendix A—Acronyms**

1311 Selected acronyms and abbreviations used in this paper are defined below.

ASIC	Application-Specific Integrated Circuit
BCC	Bitcoin Cash
BFT	Byzantine Fault Tolerant
BTC	Bitcoin
CPU	Central Processing Unit
CR	Create, Read
CRUD	Create, Read, Update, Delete
DAO	Decentralized Autonomous Organization
DID	Decentralized Identifier
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ETC	Ethereum Classic
ETH	Ethereum
EVM	Ethereum Virtual Machine
FIPS	Federal Information Processing Standard
FOIA	Freedom of Information Act
GPU	Graphics Processing Unit
I2P	Invisible Internet Project
IoT	Internet of Things
IR	Internal Report
ITL	Information Technology Laboratory
KYC	Know Your Customer
NIST	National Institute of Standards and Technology

NISTIR	National Institute of Standards and Technology Internal Report
RSA	Rivest-Shamir-Adleman
SegWit	Segregated Witness
SHA	Secure Hash Algorithm
XMR	Monero
XRP	Ripple

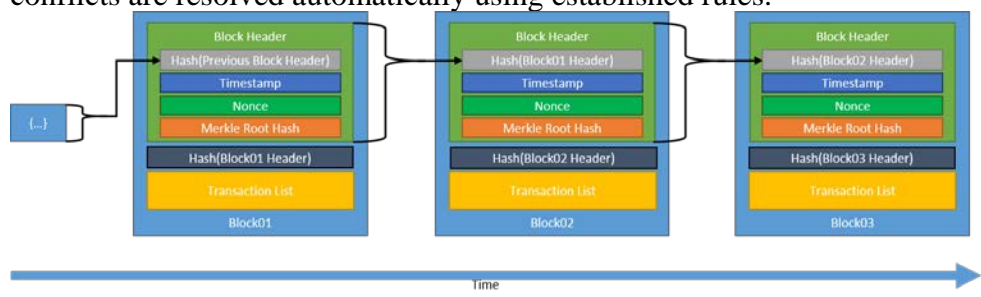
1312

1313

1314 **Appendix B—Glossary**

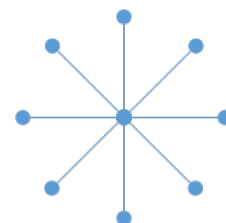
1315 Selected terms used in this paper are defined below.

- Address A short, alphanumeric string derived from a user’s public key using a hash function, with additional data to detect errors. Addresses are used to send and receive digital assets.
- Assets Anything that can be transferred.
- Block A set of validated transactions.
- Block header The portion of the block that contains information about the block itself (block metadata), usually including the timestamp for posting the block, the Merkle tree root hash, the previous block’s hash, and the cryptographic nonce (if needed).
- Blockchain A distributed digital ledger of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify. New blocks are replicated across all copies of the ledger within the network, and any conflicts are resolved automatically using established rules.



Byzantine Fault Tolerant proof of stake consensus model A proof of stake consensus model where the blockchain decides the next block by allowing all staked members to “vote” on which submitted block to include next.

Centralized network A network configuration where participants must communicate with a central authority to communicate with one another. Since all participants must go through a single centralized source, the loss of that source would prevent all participants from communicating.

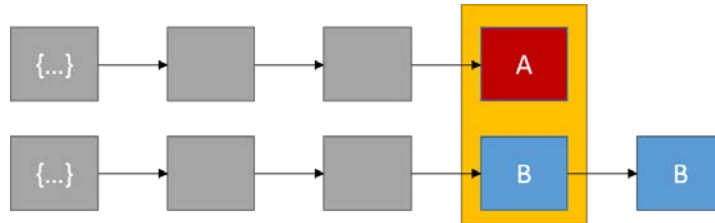


Chain-based proof of stake consensus model A proof of stake consensus model where the blockchain system decides the next block through pseudo-random selection, based on a personal stake to overall system asset ratio.

Conflict One or more participants disagree on the state of the system.

Conflict resolution

A predefined method for coming to a consensus on the state of the system (e.g., when portions of the system participants claim there is State_A and the rest of the participants claim there is State_B, there is a conflict – the system will automatically resolve this conflict by choosing the “Valid” state as being the one from whichever group adds the next block of data; any transactions “lost” by the state not chosen are added back into the unspent transaction pool).



Consensus algorithm

A predefined method to determine whether some data can be committed to a data store. Also known as a *consensus model*.

Cryptocurrency

A digital asset/credit/unit within the system, which is cryptographically sent from one user to another user. In the case of cryptocurrency creation (such as the reward for mining), the system itself generates and distributes the currency via the same cryptographic mechanisms. These assets are transferred from one wallet to another by using digital signatures with public/private key pairs.

Cryptographic hash function

A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties:

1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and
2. (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.

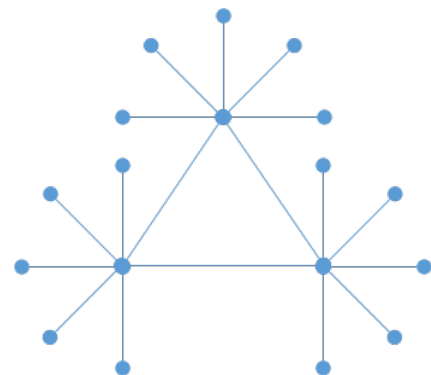
From NIST SP 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms, <http://dx.doi.org/10.6028/NIST.SP.800-175B>

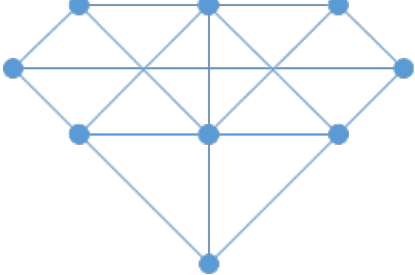
Cryptographic nonce

An arbitrary number (usually randomly selected) that is used once.

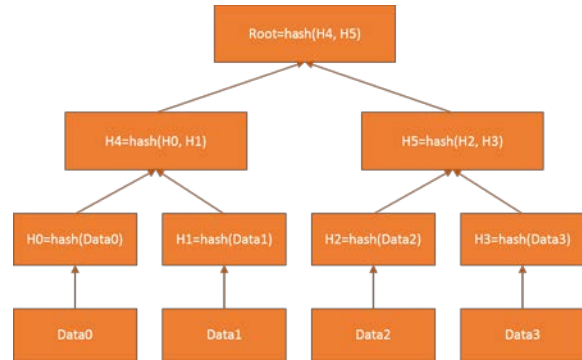
Decentralized network

A network configuration where there are multiple authorities that serve as a centralized hub for a subsection of participants. Since some participants are behind a centralized hub, the loss of that hub will prevent those participants from communicating.



Digital signature	A cryptographic technique that utilizes private/public keys to determine authenticity (i.e., users can verify that the message was signed with a private key corresponding to the specified public key), non-repudiation (a user cannot deny having sent a message) and integrity (that the message was not altered during transmission).
Distributed network	<div style="display: flex; align-items: flex-start;"> <div style="flex: 1;"> <p>A network configuration where every participant can communicate with one another without going through a centralized point. Since there are multiple pathways for communication, the loss of any participant will not prevent communication. Also known as <i>peer-to-peer</i>.</p> </div> <div style="flex: 1; text-align: center;">  </div> </div>
Fork	A change to blockchain software and implementation.
Full node	A blockchain node that stores the blockchain data, passes along the data to other nodes, and ensures that newly added blocks are valid.
Genesis block	The first block of a blockchain system; it records the initial state of the system.
Hard fork	A fork that will completely prevent users who do not adopt it from using the changed blockchain system. Users must either upgrade to stay with the developer's main fork or continue on the original path without upgrades. Users on different hard forks cannot interact with one another.
Hash chain	An append-only data structure where data is bundled into blocks that include a hash of the previous block's data within the newest block. This data structure provides evidence of tampering because any modification to a block's data will change the hash digest recorded by the following block.
Hash digest	The output of a hash function (e.g., $\text{hash}(\text{data}) = \text{digest}$). Also known as a <i>digest</i> .
Hashing	A method of calculating a relatively unique output (called a <i>hash digest</i>) for an input of nearly any size (a file, text, image, etc.) Hash algorithms are designed to be one-way; calculating the digest of an input is simple, but reconstructing the input from the digest is significantly difficult, and to be collision-resistant, so that it is computationally infeasible to find two inputs which result in the same digest. Additionally, the smallest change of input, even a single bit, will result in a completely different output digest.
Immutable	Data that can only be written, not modified.
Ledger	A collection of transactions recorded chronologically.
Lightweight node	A blockchain node that does not need to store a full copy of the blockchain and often passes its data to full nodes to be processed.

Merkle tree A data structure where the data is hashed and combined until there is a singular root hash that represents the entire structure.



Mining The act of performing the required work (as defined by the system’s consensus algorithm) to add the next block to the system and usually rewarded with a cryptocurrency. Also known as *minting*.

Mining Node One of a subset of nodes in charge of maintaining the blockchain. Also known as a *minter*.

Node An individual system within the blockchain.

Permissioned A system where every user must have their permissions assigned by an administrator.

Permissionless A system where all users’ permissions are equal and not set by any administrator.

Permissions Allowable user actions (e.g., read, write, execute).

Proof of stake consensus model A consensus model where the blockchain network is secured by users locking an amount of cryptocurrency into the blockchain system, a process called *staking*. Participants with more stake in the system are more likely to want it to succeed and to not be subverted, which gives them more weight during consensus.

Proof of work consensus model A consensus model where a mining node obtains the right to publish the next block by expending time, energy, and computational cycles to solve a hard-to-solve, but easy-to-verify problem (e.g., finding the nonce which, when combined with the data to be added to the block, will result in a specific output pattern).

Public/private key cryptography A cryptographic system where users have a private key that is kept secret and used to generate a public key (which is freely provided to others). Users can digitally sign data with their private key, and the resulting signature can be verified by anyone using the corresponding public key. Also known as asymmetric cryptography.

Round robin consensus model A consensus model for private blockchains where nodes are pseudo-randomly selected to create blocks, but a node must wait several block-creation cycles before being chosen again to add another new block. This model ensures that no one participant creates the majority of the blocks, and it benefits from a straightforward approach, lacking cryptographic puzzles, and having low power requirements.

Soft fork	A fork that will not completely prevent users who do not adopt it from using the changed blockchain system. A soft fork can be backwards compatible, only requiring that a majority of mining nodes upgrade to enforce the new soft fork rules.
Transaction	A recording of a transfer of assets (digital currency, units of inventory, etc.) between parties.
Transaction pool	A distributed queue where candidate transactions wait until they are added to the blockchain. Also known as <i>Unspent transaction pool</i> .
Turing complete	A system (computer system, programming language, etc.) that can be used for any algorithm, regardless of complexity, to find a solution.
User	Any single person, group, business, or organization which is using or operating a blockchain node
Wallet	Software used to manage public/private keys and addresses used for transactions.

1316

1317

Appendix C—References

- [1] Clarke, A.C., “Hazards of Prophecy: The Failure of Imagination,” from *Profiles of the Future: An Inquiry into the Limits of the Possible*, 1962.
- [2] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S., *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press, 2016.
- [3] Nakamoto, S., “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. <https://bitcoin.org/bitcoin.pdf>
- [4] National Institute of Standards and Technology (NIST), Federal Information Processing Standards (FIPS) Publication 180-4, *Secure Hash Standard (SHS)*, August 2015. <https://doi.org/10.6028/NIST.FIPS.180-4>
- [5] National Institute of Standards and Technology (NIST), Secure Hashing website, <https://csrc.nist.gov/projects/hash-functions>
- [6] National Institute of Standards and Technology (NIST), Federal Information Processing Standards (FIPS) Publication 186-4, *Digital Signature Standard*, July 2013. <https://doi.org/10.6028/NIST.FIPS.186-4>
- [7] Deetman, S., “Bitcoin Could Consume as Much Electricity as Denmark by 2020,” *Motherboard*, March 29, 2016. https://motherboard.vice.com/en_us/article/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020
- [8] Hern, A., “Bitcoin mining consumes more electricity a year than Ireland,” *The Guardian*, November 27, 2017. <https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland>
- [9] Power Compare, <https://powercompare.co.uk/bitcoin/>
- [10] Bahsoun, J.P., Guerraoui, R., and Shoker, A., “Making BFT Protocols Really Adaptive,” *2015 IEEE International Parallel and Distributed Processing Symposium*, Hyderabad, India, pp. 904-913, 2015. <https://doi.org/10.1109/IPDPS.2015.21>
- [11] Todd, P., Bitcoin Improvement Proposal (BIP) 65, “OP_CHECKLOCKTIMEVERIFY,” October 1, 2014. <https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki>
- [12] Wong, J. and Kar, I., “Everything you need to know about the Ethereum ‘hard fork,’” *Quartz Media*, July 18, 2016. <https://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/>
- [13] National Institute of Standards and Technology (NIST), Federal Information Processing Standards (FIPS) Publication 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, August 2015. <https://doi.org/10.6028/NIST.FIPS.202>

- [14] Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D., National Institute of Standards and Technology (NIST), NIST Internal Report (NISTIR) 8105, *Report on Post-Quantum Cryptography*, April 2016. <https://doi.org/10.6028/NIST.IR.8105>
- [15] Mell, P., Kelsey, J., and Shook, J., “Cryptocurrency Smart Contracts for Distributed Consensus of Public Randomness.” October 7, 2017. https://doi.org/10.1007/978-3-319-69084-1_31
- [16] Lamport, L., “Leaderless Byzantine Paxos,” *Distributed Computing: 25th International Symposium: DISC 2011*, p. 141-142, December 27, 2011. <https://www.microsoft.com/en-us/research/publication/leaderless-byzantine-paxos>
- [17] Gipp, B., Meuschke, N., and Gernandt, A., “Decentralized Trusted Timestamping using the Crypto Currency Bitcoin,” in *Proceedings of the iConference 2015*, Newport Beach, California, 2015.
- [18] Mattila, J., Seppälä, T., Naucler, C., Stahl, R., Tikkanen, M., Bådenlid, A., and Seppälä, J., The Research Institute of the Finnish Economy (ETLA) Working Papers No. 43, “Industrial Blockchain Platforms: An Exercise in Use Case Development in the Energy Industry,” The Research Institute of the Finnish Economy, October 11, 2016. <https://www.etla.fi/wp-content/uploads/ETLA-Working-Papers-43.pdf>
- [19] Donnelly, J., “What is the 'Halving'? A Primer to Bitcoin's Big Mining Change,” *CoinDesk*, June 12, 2016. <https://www.coindesk.com/making-sense-bitcoins-halving/>
- [20] Hertig, A., “Litecoin’s SegWit Activation: Why it Matters and What’s Next,” *CoinDesk*, April 26, 2017. <https://www.coindesk.com/litecoins-segwit-activation-why-it-matters-and-whats-next/>
- [21] Litecoin Project. <https://litecoin.org/>
- [22] Wood, G., “Ethereum: A Secure Decentralised Generalised Transaction Ledger.” <https://bravenewcoin.com/assets/Whitepapers/Ethereum-A-Secure-Decentralised-Generalised-Transaction-Ledger-Yellow-Paper.pdf>
- [23] Pearson, J., “The Ethereum Hard Fork Spawned a Shaky Rebellion,” *Motherboard*, July 27, 2016. https://motherboard.vice.com/en_us/article/the-ethereum-hard-fork-spawned-a-shaky-rebellion-ethereum-classic-etc-eth
- [24] “What Is Dash?”, WeUseCoins. <https://www.weusecoins.com/what-is-dash/>
- [25] Duffield, E. and Diaz, D., “Dash: A Privacy-Centric Crypto-Currency.” <https://github.com/dashpay/dash/wiki/Whitepaper>
- [26] “Introduction to Ripple for Bitcoiners,” last modified December 10, 2013. https://wiki.ripple.com/Introduction_to_Ripple_for_Bitcoiners
- [27] Brown, A., “10 things you need to know about Ripple,” *CoinDesk*, May 17, 2013. <https://www.coindesk.com/10-things-you-need-to-know-about-ripple/>

- [28] “Hyperledger Business Blockchain Technologies,” The Linux Foundation.
<https://www.hyperledger.org/projects>
- [29] Cachin, C., “Architecture of the Hyperledger blockchain fabric,” in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, July 2016.
- [30] Greenspan, G., “Introducing MultiChain Streams,” MultiChain, September 15, 2016. <http://www.multichain.com/blog/2016/09/introducing-multichain-streams/>
- [31] Narayanan, A., “Analyzing the 2013 Bitcoin fork: centralized decision-making saved the day,” MultiChain, July 28, 2015. <https://freedom-to-tinker.com/2015/07/28/analyzing-the-2013-bitcoin-fork-centralized-decision-making-saved-the-day>
- [32] Greenspan, G., “The Blockchain Immutability Myth,” MultiChain, May 4, 2017. <https://www.multichain.com/blog/2017/05/blockchain-immutability-myth/>
- [33] “Bitcoin blockchain size reaches 100 GB,” Coinfox, December 19, 2016. <http://www.coinfox.info/news/6700-bitcoin-blockchain-size-reaches-100-gb>

1318