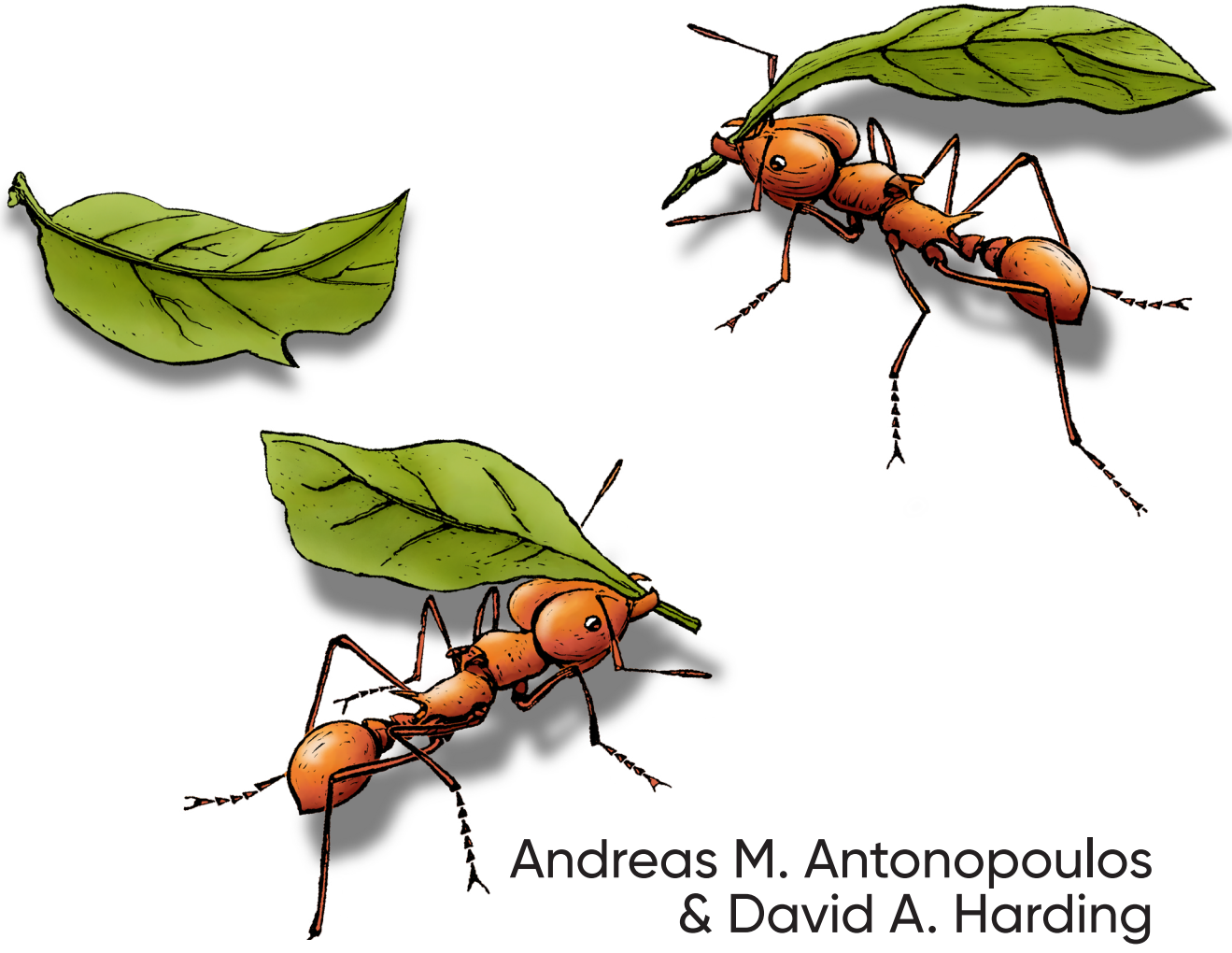
O'REILLY®

Third Edition

Mastering Bitcoin

Programming the Open Blockchain



Andreas M. Antonopoulos
& David A. Harding

O'REILLY®

Mastering Bitcoin

Join the technological revolution that's taking the financial world by storm. *Mastering Bitcoin* is your guide through the seemingly complex world of Bitcoin, providing the knowledge you need to participate in the internet of money. Whether you're building the next killer app, investing in a startup, or simply curious about the technology, this revised and expanded third edition provides essential detail to get you started.

Bitcoin, the first successful decentralized digital currency, has already spawned a multibillion-dollar global economy open to anyone with the knowledge and passion to participate. *Mastering Bitcoin* provides the knowledge. You supply the passion.

The third edition includes:

- **A broad introduction to Bitcoin and its underlying blockchain—ideal for nontechnical users, investors, and business executives**
- **An explanation of Bitcoin's technical foundation and cryptographic currency for developers, engineers, and software and systems architects**
- **Details of the Bitcoin decentralized network, peer-to-peer architecture, transaction lifecycle, and security principles**
- **New developments such as Taproot, Tapscript, Schnorr signatures, and the Lightning Network**
- **A deep dive into Bitcoin applications, including how to combine the building blocks offered by this platform into powerful new tools**
- **User stories, analogies, examples, and code snippets illustrating key technical concepts**

"Nearly a decade after the initial publishing, the third edition of *Mastering Bitcoin* cements the book's role as the go-to source of technical Bitcoin educational content. No other book is as comprehensive or up-to-date."

—**Olaoluwa Osuntokun**
CTO at Lightning Labs

"A comprehensive overview of what goes on under Bitcoin's hood and how things fit together."

—**Mark "Murch" Erhardt**
Bitcoin engineer at Chaincode Labs

Andreas M. Antonopoulos is an expert in Bitcoin and open blockchain technologies.

David A. Harding is coauthor of the Bitcoin Optech weekly newsletter.

DATA

US $69.99 CAN $87.99
ISBN: 978-1-098-15009-9

Twitter: @oreillymedia
linkedin.com/company/oreilly-media
youtube.com/oreillymedia

Praise for *Mastering Bitcoin*, Third Edition

Mastering Bitcoin is very useful for anyone who wants or needs to understand the technology of Bitcoin and the concepts of the protocol.

—René Pickhardt, Bitcoin Lightning Network developer

Delve into the fascinating world of Bitcoin with *Mastering Bitcoin*, the definitive guide that navigates through the intricacies of this digital currency. Whether you're a developer, an investor, or simply curious about the future of money, this comprehensive book serves as your roadmap, providing essential knowledge and empowering you to participate confidently in the Internet of money era.

—Jorge Lesmes, Senior Director at NTT DATA

Nearly a decade after the initial publishing, the third edition of *Mastering Bitcoin* cements the book's role as the go-to source of technical Bitcoin educational content. No other book is as comprehensive or up-to-date.

—Olaoluwa Osuntokun, CTO at Lightning Labs

A comprehensive overview of what goes on under Bitcoin's hood and how things fit together.

—Mark "Murch" Erhardt, Bitcoin engineer, Chaincode Labs

THIRD EDITION

Mastering Bitcoin

Programming the Open Blockchain

Andreas M. Antonopoulos and David A. Harding

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Mastering Bitcoin
by Andreas M. Antonopoulos and David A. Harding

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*https://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Michelle Smith
Development Editor: Angela Rufino
Production Editor: Clare Laylock
Copyeditor: Kim Cofer
Proofreader: Heather Walley

Indexer: nSight, Inc.
Interior Designer: David Futato
Cover Designer: Randy Comer
Illustrator: Kate Dullea

December 2014: First Edition
June 2017: Second Edition
November 2023: Third Edition

Revision History for the Third Edition
2023-11-03: First Release

See *https://oreilly.com/catalog/errata.csp?isbn=9781098150099* for release details.

978-1-098-15009-9

[LSI]

Dedicated to my mum, Theresa (1946–2017)
She taught me to love books and question authority
Thank you, mum
—Andreas

For Amanda
It wasn't until I met you that I
actually began living in paradise
—Dave

Table of Contents

Preface

Writing the Bitcoin Book

I (Andreas) first stumbled upon Bitcoin in mid-2011. My immediate reaction was more or less "Pfft! Nerd money!" and I ignored it for another six months, failing to grasp its importance. This is a reaction that I have seen repeated among many of the smartest people I know, which gives me some consolation. The second time I came across Bitcoin, in a mailing list discussion, I decided to read the whitepaper written by Satoshi Nakamoto and see what it was all about. I still remember the moment I finished reading those nine pages, when I realized that Bitcoin was not simply a digital currency, but a network of trust that could also provide the basis for so much more than just currencies. The realization that "this isn't money, it's a decentralized trust network," started me on a four-month journey to devour every scrap of information about Bitcoin I could find. I became obsessed and enthralled, spending 12 or more hours each day glued to a screen, reading, writing, coding, and learning as much as I could. I emerged from this state of fugue, more than 20 pounds lighter from lack of consistent meals, determined to dedicate myself to working on Bitcoin.

Two years later, after creating a number of small startups to explore various Bitcoin-related services and products, I decided that it was time to write my first book. Bitcoin was the topic that had driven me into a frenzy of creativity and consumed my thoughts; it was the most exciting technology I had encountered since the internet. It was now time to share my passion about this amazing technology with a broader audience.

Intended Audience

This book is mostly intended for coders. If you can use a programming language, this book will teach you how cryptographic currencies work, how to use them, and how to develop software that works with them. The first few chapters are also suitable as

an in-depth introduction to Bitcoin for noncoders—those trying to understand the inner workings of Bitcoin and cryptocurrencies.

Why Are There Bugs on the Cover?

The leafcutter ant is a species that exhibits highly complex behavior in a colony super-organism, but each individual ant operates on a set of simple rules driven by social interaction and the exchange of chemical scents (pheromones). Per Wikipedia: "Next to humans, leafcutter ants form the largest and most complex animal societies on Earth." Leafcutter ants don't actually eat leaves, but rather use them to farm a fungus, which is the central food source for the colony. Get that? These ants are farming!

Although ants form a caste-based society and have a queen for producing offspring, there is no central authority or leader in an ant colony. The highly intelligent and sophisticated behavior exhibited by a multimillion-member colony is an emergent property from the interaction of the individuals in a social network.

Nature demonstrates that decentralized systems can be resilient and can produce emergent complexity and incredible sophistication without the need for a central authority, hierarchy, or complex parts.

Bitcoin is a highly sophisticated decentralized trust network that can support myriad financial processes. Yet, each node in the Bitcoin network follows a few simple rules. The interaction between many nodes is what leads to the emergence of the sophisticated behavior, not any inherent complexity or trust in any single node. Like an ant colony, the Bitcoin network is a resilient network of simple nodes following simple rules that together can do amazing things without any central coordination.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

This element signifies a tip or suggestion.

This element signifies a general note.

This element indicates a warning or caution.

Code Examples

All the code snippets can be replicated on most operating systems with a minimal installation of compilers and interpreters for the corresponding languages. Where necessary, we provide basic installation instructions and step-by-step examples of the output of those instructions.

Some of the code snippets and code output have been reformatted for print. In all such cases, the lines have been split by a backslash (\) character, followed by a newline character. When transcribing the examples, remove those two characters and join the lines again and you should see identical results as shown in the example.

All the code snippets use real values and calculations where possible, so that you can build from example to example and see the same results in any code you write to calculate the same values.

Using Code Examples

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book

and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Mastering Bitcoin*, 3rd ed., by Andreas M. Antonopoulos and David A. Harding (O'Reilly). Copyright 2024 David Harding, ISBN 978-1-098-15009-9."

Some editions of this book are offered under an open source license, such as CC-BY-NC, in which case the terms of that license apply.

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Changes Since the Previous Edition

A particular focus in the third edition has been modernizing the 2017 second edition text and the remaining 2014 first edition text. In addition, many concepts that are relevant to contempory Bitcoin development in 2023 have been added:

Chapter 4
: We rearranged the address info so that we work through everything in historical order, adding a new section with P2PK (where "address" was "IP address"), refreshed the previous P2PKH and P2SH sections, and then added new sections for segwit/bech32 and taproot/bech32m.

Old Chapters 6 and 7
: Text from previous versions of Chapter 6, "Transactions," and Chapter 7, "Advanced Transactions," has been rearranged and expanded across four new chapters: Chapter 6, "Transactions" (the structure of transactions), Chapter 7, "Authorization and Authentication", Chapter 8, "Digital Signatures", and Chapter 9, "Transaction Fees".

Chapter 6
: We added almost entirely new text describing the structure of a transaction.

Chapter 7
: We added new text about MAST, P2C, scriptless multisignatures, taproot, and tapscript.

Chapter 8
: We revised the ECDSA text and added new text about schnorr signatures, multisignatures, and threshold signatures.

Chapter 9
: We added almost entirely new text about fees, RBF and CPFP fee bumping, transaction pinning, package relay, and CPFP carve-out.

Chapter 10
: We added text about compact block relay, added a significant update to bloom filters that better describes their privacy problems, and new text about compact block filters.

Chapter 11
: We added text about signet.

Chapter 12
: We added text about BIP8 and speedy trial.

Appendixes
: We removed library-specific appendixes. After the appendix containing the original whitepaper, we added a new appendix describing how the implementation and properties of Bitcoin differ from those proposed in the whitepaper.

Bitcoin Addresses and Transactions in This Book

The Bitcoin addresses, transactions, keys, QR codes, and blockchain data used in this book are, for the most part, real. That means you can browse the blockchain, look at the transactions offered as examples, retrieve them with your own scripts or programs, etc.

However, note that the private keys used to construct addresses are either printed in this book or have been "burned." That means if you send money to any of these addresses, the money will either be lost forever, or in some cases everyone who can read the book can take it using the private keys printed in here.

DO NOT SEND MONEY TO ANY OF THE ADDRESSES IN THIS BOOK. Your money will be taken by another reader or lost forever.

O'Reilly Online Learning

For more than 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit *https://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-889-8969 (in the United States or Canada)
707-829-7019 (international or local)
707-829-0104 (fax)
support@oreilly.com
https://www.oreilly.com/about/contact.html

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://oreil.ly/MasteringBitcoin3e*.

For news and information about our books and courses, visit *https://oreilly.com*.

Find us on LinkedIn: *https://linkedin.com/company/oreilly-media*.

Follow us on Twitter: *https://twitter.com/oreillymedia*.

Watch us on YouTube: *https://youtube.com/oreillymedia*.

Contacting the Authors

You can contact Andreas M. Antonopoulos on his personal site: *https://antonopoulos.com*.

Follow Andreas on Facebook: *https://facebook.com/AndreasMAntonopoulos*.

Follow Andreas on Twitter: *https://twitter.com/aantonop*.

Follow Andreas on LinkedIn: *https://linkedin.com/company/aantonop*.

Many thanks to all of Andreas's patrons who support his work through monthly donations. You can follow his Patreon page here: *https://patreon.com/aantonop*.

Information about *Mastering Bitcoin*, as well as Andreas's Open Edition and translations, is available on *https://bitcoinbook.info*.

You can contact David A. Harding on his personal site: *https://dtrt.org*.

Acknowledgments for the First and Second Editions

By Andreas M. Antonopoulos

This book represents the efforts and contributions of many people. I am grateful for all the help I received from friends, colleagues, and even complete strangers, who joined me in this effort to write the definitive technical book on cryptocurrencies and Bitcoin.

It is impossible to make a distinction between the Bitcoin technology and the Bitcoin community, and this book is as much a product of that community as it is a book on the technology. My work on this book was encouraged, cheered on, supported, and rewarded by the entire Bitcoin community from the very beginning until the very end. More than anything, this book has allowed me to be part of a wonderful community for two years and I can't thank you enough for accepting me into this community. There are far too many people to mention by name—people I've met at conferences, events, seminars, meetups, pizza gatherings, and small private gatherings, as well as many who communicated with me by Twitter, on reddit, on bitcointalk.org, and on GitHub who have had an impact on this book. Every idea, analogy, question, answer, and explanation you find in this book was at some point inspired, tested, or improved through my interactions with the community. Thank you all for your support; without you this book would not have happened. I am forever grateful.

The journey to becoming an author starts long before the first book, of course. My first language (and schooling) was Greek, so I had to take a remedial English writing course in my first year of university. I owe thanks to Diana Kordas, my English writing teacher, who helped me build confidence and skills that year. Later, as a professional, I developed my technical writing skills on the topic of data centers, writing for *Network World* magazine. I owe thanks to John Dix and John Gallant, who gave me my first writing job as a columnist at *Network World* and to my editor Michael Cooney and my colleague Johna Till Johnson who edited my columns and made them fit for publication. Writing 500 words a week for four years gave me enough experience to eventually consider becoming an author.

Thanks also to those who supported me when I submitted my book proposal to O'Reilly by providing references and reviewing the proposal. Specifically, thanks to John Gallant, Gregory Ness, Richard Stiennon, Joel Snyder, Adam B. Levine, Sandra Gittlen, John Dix, Johna Till Johnson, Roger Ver, and Jon Matonis. Special thanks to Richard Kagan and Tymon Mattoszko, who reviewed early versions of the proposal and Matthew Taylor, who copyedited the proposal.

Thanks to Cricket Liu, author of the O'Reilly title *DNS and BIND*, who introduced me to O'Reilly. Thanks also to Michael Loukides and Allyson MacDonald at O'Reilly, who worked for months to help make this book happen. Allyson was especially

patient when deadlines were missed and deliverables delayed as life intervened in our planned schedule. For the second edition, I thank Timothy McGovern for guiding the process, Kim Cofer for patiently editing, and Rebecca Panzer for illustrating many new diagrams.

The first few drafts of the first few chapters were the hardest, because Bitcoin is a difficult subject to unravel. Every time I pulled on one thread of the Bitcoin technology, I had to pull on the whole thing. I repeatedly got stuck and a bit despondent as I struggled to make the topic easy to understand and create a narrative around such a dense technical subject. Eventually, I decided to tell the story of Bitcoin through the stories of the people using Bitcoin and the whole book became a lot easier to write. I owe thanks to my friend and mentor, Richard Kagan, who helped me unravel the story and get past the moments of writer's block. I thank Pamela Morgan, who reviewed early drafts of each chapter in the first and second edition of the book and asked the hard questions to make them better. Also, thanks to the developers of the San Francisco Bitcoin Developers Meetup group as well as Taariq Lewis and Denise Terry for helping test the early material. Thanks also to Andrew Naugler for infographic design.

During the development of the book, I made early drafts available on GitHub and invited public comments. More than a hundred comments, suggestions, corrections, and contributions were submitted in response. Those contributions are explicitly acknowledged, with my thanks, in "Early Release Draft (GitHub Contributions)" on page xxiii. Most of all, my sincere thanks to my volunteer GitHub editors Ming T. Nguyen (1st edition) and Will Binns (2nd edition), who worked tirelessly to curate, manage, and resolve pull requests, issue reports, and perform bug fixes on GitHub.

Once the book was drafted, it went through several rounds of technical review. Thanks to Cricket Liu and Lorne Lantz for their thorough review, comments, and support.

Several Bitcoin developers contributed code samples, reviews, comments, and encouragement. Thanks to Amir Taaki and Eric Voskuil for example code snippets and many great comments; Chris Kleeschulte for contributing information about Bitcore; Vitalik Buterin and Richard Kiss for help with elliptic curve math and code contributions; Gavin Andresen for corrections, comments, and encouragement; Michalis Kargakis for comments, contributions, and btcd writeup; and Robin Inge for errata submissions improving the second print. In the second edition, I again received a lot of help from many Bitcoin Core developers, including Eric Lombrozo who demystified segregated witness, Luke Dashjr who helped improve the chapter on transactions, Johnson Lau who reviewed segregated witness and other chapters, and many others. I owe thanks to Joseph Poon, Tadge Dryja, and Olaoluwa Osuntokun who explained Lightning Network, reviewed my writing, and answered questions when I got stuck.

I owe my love of words and books to my mother, Theresa, who raised me in a house with books lining every wall. My mother also bought me my first computer in 1982, despite being a self-described technophobe. My father, Menelaos, a civil engineer who just published his first book at 80 years old, was the one who taught me logical and analytical thinking and a love of science and engineering.

Thank you all for supporting me throughout this journey.

Acknowledgments for the Third Edition

By David A. Harding

The introduction to the noninteractive schnorr signature protocol that starts with first describing the interactive schnorr identity protocol in "Schnorr Signatures" on page 187 was heavily influenced by the introduction to the subject in "Borrommean Ring Signatures" (2015) by Gregory Maxwell and Andrew Poelstra. I am deeply indebted to each of them for all of their freely provided assistance over the past decade.

Invaluable technical reviews on drafts of this manuscript were provided by Jorge Lesmes, Olaoluwa Osuntokun, René Pickhardt, and Mark "Murch" Erhardt. In particular, Murch's incredibly in-depth and insightful review, and his willingness to evaluate multiple iterations of the same text, have elevated the quality of this book beyond my highest expectations.

I also owe a debt of gratitude to Jimmy Song for suggesting me for this project, to my coauthor Andreas for allowing me to update his bestselling text, to Angela Rufino for guiding me through the O'Reilly authorship process, and to all of the other staff at O'Reilly for making the writing of the third edition a pleasant and productive experience.

Finally, I don't know how I can thank all of the Bitcoin contributors who have helped me on my journey—from creating the software I use, to teaching me how it works, to helping me pass on what little knowledge I've gained. There are too many of you to list your names, but I think of you often and know that my contributions to this book would not have been possible without all that you've done for me.

Early Release Draft (GitHub Contributions)

Many contributors offered comments, corrections, and additions to the early-release draft on GitHub. Thank you all for your contributions to this book.

Following is a list of notable GitHub contributors, including their GitHub ID in parentheses:

- Abdussamad Abdurrazzaq (AbdussamadA)
- Adán SDPC (aesedepece)
- Akira Chiku (achiku)
- Alex Waters (alexwaters)
- Andrew Donald Kennedy (grkvlt)
- Andrey Esaulov (andremaha)
- andronoob
- AnejaBK
- Appaji (CITIZENDOT)
- ariesunny
- Arthur O'Dwyer (Quuxplusone)
- bargitta
- Basem Alasi (Bamskki)
- bisqfan
- bitcoinctf
- blip151
- Bryan Gmyrek (physicsdude)
- Carlos Sims (simsbluebox)
- Casey Flynn (cflynn07)
- cclauss
- Chapman Shoop (belovachap)
- chrisd95
- Christie D'Anna (avocadobreath)
- Cihat Imamoglu (cihati)
- Cody Scott (Siecje)
- coinradar
- Cragin Godley (cgodley)
- Craig Dodd (cdodd)
- dallyshalla
- Dan Nolan (Dan-Nolan)
- Dan Raviv (danra)
- Darius Kramer (dkrmr)
- Darko Janković (trulex)
- David Huie (DavidHuie)
- didongke
- Diego Viola (diegoviola)
- Dimitris Tsapakidis (dimitris-t)
- Dirk Jäckel (biafra23)
- Dmitry Marakasov (AMDmi3)
- drakos (Jolly-Pirate)
- drstrangeM
- Ed Eykholt (edeykholt)
- Ed Leafe (EdLeafe)
- Edward Posnak (edposnak)
- Elias Rodrigues (elias19r)
- Eric Voskuil (evoskuil)
- Eric Winchell (winchell)
- Erik Wahlström (erikwam)
- effectsToCause (vericoin)
- Esteban Ordano (eordano)
- ethers
- Evlix
- fabienhinault
- Fan (whiteath)
- Felix Filozov (ffilozov)
- Francis Ballares (fballares)
- François Wirion (wirion)
- Frank Höger (francyi)
- Gabriel Montes (gabmontes)
- Gaurav Rana (bitcoinsSG)
- genjix
- Geremia
- Gerry Smith (Hermetic)

- gmr81
- Greg (in3rsha)
- Gregory Trubetskoy (grisha)
- Gus (netpoe)
- halseth
- harelw
- Harry Moreno (morenoh149)
- Hennadii Stepanov (hebasto)
- Holger Schinzel (schinzelh)
- Ioannis Cherouvim (cherouvim)
- Ish Ot Jr. (ishotjr)
- ivangreene
- James Addison (jayaddison)
- Jameson Lopp (jlopp)
- Jason Bisterfeldt (jbisterfeldt)
- Javier Rojas (fjrojasgarcia)
- Jordan Baczuk (JBaczuk)
- Jeremy Bokobza (bokobza)
- JerJohn15
- jerzybrzoska
- Jimmy DeSilva (jimmydesilva)
- Jo Wo (jowo-io)
- Joe Bauers (joebauers)
- joflynn
- Johnson Lau (jl2012)
- Jonathan Cross (jonathancross)
- Jorgeminator
- jwbats
- Kai Bakker (kaibakker)
- kollokollo
- krupawan5618
- kynnjo
- Liangzx
- lightningnetworkstores
- lilianrambu
- Liu Yue (lyhistory)
- Lobbelt
- Lucas Betschart (lclc)
- Matt Wesley (MatthewWesley)
- Magomed Aliev (30mb1)
- Mai-Hsuan Chia (mhchia)
- Marco Falke (MarcoFalke)
- María Martín (mmartinbar)
- Marcus Kiisa (mkiisa)
- Mark Erhardt (Xekyo)
- Mark Pors (pors)
- Martin Harrigan (harrigan)
- Martin Vseticka (MartyIX)
- Marzig (marzig76)
- Matt McGivney (mattmcgiv)
- Matthijs Roelink (Matthiti)
- Maximilian Reichel (phramz)
- MG-ng (MG-ng)
- Michalis Kargakis (kargakis)
- Michael C. Ippolito (michaelcippolito)
- Michael Galero (mikong)
- Michael Newman (michaelbnewman)
- Mihail Russu (MihailRussu)
- mikew (mikew)
- milansismanovic
- Minh T. Nguyen (enderminh)
- montvid
- Morfies (morfies)

- Nagaraj Hubli (nagarajhubli)
- Nekomata (nekomata-3)
- nekonenene
- Nhan Vu (jobnomade)
- Nicholas Chen (nickycutesc)
- Ning Shang (syncom)
- Oge Nnadi (ogennadi)
- Oliver Maerz (OliverMaerz)
- Omar Boukli-Hacene (oboukli)
- Óscar Nájera (Titan-C)
- Parzival (Parz-val)
- Paul Desmond Parker (sunwukonga)
- Philipp Gille (philippgille)
- ratijas
- rating89us
- Raul Siles (raulsiles)
- Reproducibility Matters (TheCharlatan)
- Reuben Thomas (rrthomas)
- Robert Furse (Rfurse)
- Roberto Mannai (robermann)
- Richard Kiss (richardkiss)
- rszheng
- Ruben Alexander (hizzvizz)
- Sam Ritchie (sritchie)
- Samir Sadek (netsamir)
- Sandro Conforto (sandroconforto)
- Sanjay Sanathanan (sanjays95)
- Sebastian Falbesoner (theStack)
- Sergei Tikhomirov (s-tikhomirov)
- Sergej Kotliar (ziggamon)
- Seiichi Uchida (topecongiro)
- shaysw
- Simon de la Rouviere (simondlr)
- simone-cominato
- sindhoor7
- Stacie (staciewaleyko)
- Stephan Oeste (Emzy)
- Stéphane Roche (Janaka-Steph)
- takaya-imai
- Thiago Arrais (thiagoarrais)
- Thomas Kerin (afk11)
- Tochi Obudulu (tochicool)
- Tosin (tkuye)
- Vasil Dimov (vasild)
- venzen
- Vlad Stan (motorina0)
- Vijay Chavda (VijayChavda)
- Vincent Déniel (vincentdnl)
- weinim
- wenxiaolong (QingShiLuoGu)
- wenzhenxiang
- Will Binns (wbnns)
- wintercooled
- wjx
- wll2007
- Wojciech Langiewicz (wlk)
- Yancy Ribbens (yancyribbens)
- yjjnls
- Yoshimasa Tanabe (emag)
- yuntai
- yurigeorgiev4
- Zheng Jia (zhengjia)
- Zhou Liang (zhouguoguo)

CHAPTER 1
Introduction

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem. Units of currency called bitcoin are used to store and transmit value among participants in the Bitcoin network. Bitcoin users communicate with each other using the Bitcoin protocol primarily via the internet, although other transport networks can also be used. The Bitcoin protocol stack, available as open source software, can be run on a wide range of computing devices, including laptops and smartphones, making the technology easily accessible.

In this book, the unit of currency is called "bitcoin" with a small *b*, and the system is called "Bitcoin," with a capital *B*.

Users can transfer bitcoin over the network to do just about anything that can be done with conventional currencies, including buying and selling goods, sending money to people or organizations, or extending credit. Bitcoin can be purchased, sold, and exchanged for other currencies at specialized currency exchanges. Bitcoin is arguably the perfect form of money for the internet because it is fast, secure, and borderless.

Unlike traditional currencies, the bitcoin currency is entirely virtual. There are no physical coins or even individual digital coins. The coins are implied in transactions that transfer value from spender to receiver. Users of Bitcoin control keys that allow them to prove ownership of bitcoin in the Bitcoin network. With these keys, they can sign transactions to unlock the value and spend it by transferring it to a new owner. Keys are often stored in a digital wallet on each user's computer or smartphone.

Possession of the key that can sign a transaction is the only prerequisite to spending bitcoin, putting the control entirely in the hands of each user.

Bitcoin is a distributed, peer-to-peer system. As such, there is no central server or point of control. Units of bitcoin are created through a process called "mining," which involves repeatedly performing a computational task that references a list of recent Bitcoin transactions. Any participant in the Bitcoin network may operate as a miner, using their computing devices to help secure transactions. Every 10 minutes, on average, one Bitcoin miner can add security to past transactions and is rewarded with both brand new bitcoins and the fees paid by recent transactions. Essentially, Bitcoin mining decentralizes the currency-issuance and clearing functions of a central bank and replaces the need for any central bank.

The Bitcoin protocol includes built-in algorithms that regulate the mining function across the network. The difficulty of the computational task that miners must perform is adjusted dynamically so that, on average, someone succeeds every 10 minutes regardless of how many miners (and how much processing) are competing at any moment. The protocol also periodically decreases the number of new bitcoins that are created, limiting the total number of bitcoins that will ever be created to a fixed total just below 21 million coins. The result is that the number of bitcoins in circulation closely follows an easily predictable curve where half of the remaining coins are added to circulation every four years. At approximately block 1,411,200, which is expected to be produced around the year 2035, 99% of all bitcoins that will ever exist will have been issued. Due to Bitcoin's diminishing rate of issuance, over the long term, the Bitcoin currency is deflationary. Furthermore, nobody can force you to accept any bitcoins that were created beyond the expected issuance rate.

Behind the scenes, Bitcoin is also the name of the protocol, a peer-to-peer network, and a distributed computing innovation. Bitcoin builds on decades of research in cryptography and distributed systems and includes at least four key innovations brought together in a unique and powerful combination. Bitcoin consists of:

- A decentralized peer-to-peer network (the Bitcoin protocol)
- A public transaction journal (the blockchain)
- A set of rules for independent transaction validation and currency issuance (consensus rules)
- A mechanism for reaching global decentralized consensus on the valid blockchain (proof-of-work algorithm)

As a developer, I see Bitcoin as akin to the internet of money, a network for propagating value and securing the ownership of digital assets via distributed computation. There's a lot more to Bitcoin than first meets the eye.

In this chapter we'll get started by explaining some of the main concepts and terms, getting the necessary software, and using Bitcoin for simple transactions. In the following chapters, we'll start unwrapping the layers of technology that make Bitcoin possible and examine the inner workings of the Bitcoin network and protocol.

Digital Currencies Before Bitcoin

The emergence of viable digital money is closely linked to developments in cryptography. This is not surprising when one considers the fundamental challenges involved with using bits to represent value that can be exchanged for goods and services. Three basic questions for anyone accepting digital money are:

- Can I trust that the money is authentic and not counterfeit?
- Can I trust that the digital money can only be spent once (known as the "double-spend" problem)?
- Can I be sure that no one else can claim this money belongs to them and not me?

Issuers of paper money are constantly battling the counterfeiting problem by using increasingly sophisticated papers and printing technology. Physical money addresses the double-spend issue easily because the same paper note cannot be in two places at once. Of course, conventional money is also often stored and transmitted digitally. In these cases, the counterfeiting and double-spend issues are handled by clearing all electronic transactions through central authorities that have a global view of the currency in circulation. For digital money, which cannot take advantage of esoteric inks or holographic strips, cryptography provides the basis for trusting the legitimacy of a user's claim to value. Specifically, cryptographic digital signatures enable a user to sign a digital asset or transaction proving the ownership of that asset. With the appropriate architecture, digital signatures also can be used to address the double-spend issue.

When cryptography started becoming more broadly available and understood in the late 1980s, many researchers began trying to use cryptography to build digital currencies. These early digital currency projects issued digital money, usually backed by a national currency or precious metal such as gold.

Although these earlier digital currencies worked, they were centralized and, as a result, were easy to attack by governments and hackers. Early digital currencies used a central clearinghouse to settle all transactions at regular intervals, just like a traditional banking system. Unfortunately, in most cases these nascent digital currencies were targeted by worried governments and eventually litigated out of existence. Some failed in spectacular crashes when the parent company liquidated abruptly. To be robust against intervention by antagonists, whether legitimate governments or criminal elements, a *decentralized* digital currency was needed to avoid a single point of attack. Bitcoin is such a system, decentralized by design, and free of any central authority or point of control that can be attacked or corrupted.

History of Bitcoin

Bitcoin was first described in 2008 with the publication of a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System,"[1] written under the alias of Satoshi Nakamoto (see Appendix A). Nakamoto combined several prior inventions such as digital signatures and Hashcash to create a completely decentralized electronic cash system that does not rely on a central authority for currency issuance or settlement and validation of transactions. A key innovation was to use a distributed computation system (called a "proof-of-work" algorithm) to conduct a global lottery every 10 minutes on average, allowing the decentralized network to arrive at *consensus* about the state of transactions. This elegantly solves the issue of double-spend where a single currency unit can be spent twice. Previously, the double-spend problem was a weakness of digital currency and was addressed by clearing all transactions through a central clearinghouse.

The Bitcoin network started in 2009, based on a reference implementation published by Nakamoto and since revised by many other programmers. The number and power of machines running the proof-of-work algorithm (mining) that provides security and resilience for Bitcoin have increased exponentially, and their combined computational power now exceeds the combined number of computing operations of the world's top supercomputers.

Satoshi Nakamoto withdrew from the public in April 2011, leaving the responsibility of developing the code and network to a thriving group of volunteers. The identity of the person or people behind Bitcoin is still unknown. However, neither Satoshi Nakamoto nor anyone else exerts individual control over the Bitcoin system, which operates based on fully transparent mathematical principles, open source code, and consensus among participants. The invention itself is groundbreaking and has already spawned new science in the fields of distributed computing, economics, and econometrics.

A Solution to a Distributed Computing Problem

Satoshi Nakamoto's invention is also a practical and novel solution to a problem in distributed computing, known as the "Byzantine Generals' Problem." Briefly, the problem consists of trying to get multiple participants without a leader to agree on a course of action by exchanging information over an unreliable and potentially compromised network. Satoshi Nakamoto's solution, which uses the concept of proof of work to achieve consensus *without a central trusted authority*, represents a breakthrough in distributed computing.

1 "Bitcoin: A Peer-to-Peer Electronic Cash System", Satoshi Nakamoto.

Getting Started

Bitcoin is a protocol that can be accessed using an application that speaks the protocol. A "Bitcoin wallet" is the most common user interface to the Bitcoin system, just like a web browser is the most common user interface for the HTTP protocol. There are many implementations and brands of Bitcoin wallets, just like there are many brands of web browsers (e.g., Chrome, Safari, and Firefox). And just like we all have our favorite browsers, Bitcoin wallets vary in quality, performance, security, privacy, and reliability. There is also a reference implementation of the Bitcoin protocol that includes a wallet, known as "Bitcoin Core," which is derived from the original implementation written by Satoshi Nakamoto.

Choosing a Bitcoin Wallet

Bitcoin wallets are one of the most actively developed applications in the Bitcoin ecosystem. There is intense competition, and while a new wallet is probably being developed right now, several wallets from last year are no longer actively maintained. Many wallets focus on specific platforms or specific uses and some are more suitable for beginners while others are filled with features for advanced users. Choosing a wallet is highly subjective and depends on the use and user expertise. Therefore, it would be pointless to recommend a specific brand or wallet. However, we can categorize Bitcoin wallets according to their platform and function and provide some clarity about all the different types of wallets that exist. It is worth trying out several different wallets until you find one that fits your needs.

Types of Bitcoin wallets

Bitcoin wallets can be categorized as follows, according to the platform:

Desktop wallet
: A desktop wallet was the first type of Bitcoin wallet created as a reference implementation. Many users run desktop wallets for the features, autonomy, and control they offer. Running on general-use operating systems such as Windows and macOS has certain security disadvantages, however, as these platforms are often insecure and poorly configured.

Mobile wallet
: A mobile wallet is the most common type of Bitcoin wallet. Running on smartphone operating systems such as Apple iOS and Android, these wallets are often a great choice for new users. Many are designed for simplicity and ease-of-use, but there are also fully featured mobile wallets for power users. To avoid downloading and storing large amounts of data, most mobile wallets retrieve information from remote servers, reducing your privacy by disclosing to third parties information about your Bitcoin addresses and balances.

Web wallet
: Web wallets are accessed through a web browser and store the user's wallet on a server owned by a third party. This is similar to webmail in that it relies entirely on a third-party server. Some of these services operate using client-side code running in the user's browser, which keeps control of the Bitcoin keys in the hands of the user, although the user's dependence on the server still compromises their privacy. Most, however, take control of the Bitcoin keys from users in exchange for ease-of-use. It is inadvisable to store large amounts of bitcoin on third-party systems.

Hardware signing devices
: Hardware signing devices are devices that can store keys and sign transactions using special-purpose hardware and firmware. They usually connect to a desktop, mobile, or web wallet via USB cable, near-field-communication (NFC), or a camera with QR codes. By handling all Bitcoin-related operations on the specialized hardware, these wallets are less vulnerable to many types of attacks. Hardware signing devices are sometimes called "hardware wallets", but they need to be paired with a full-featured wallet to send and receive transactions, and the security and privacy offered by that paired wallet plays a critical role in how much security and privacy the user obtains when using the hardware signing device.

Full node versus Lightweight

Another way to categorize Bitcoin wallets is by their degree of autonomy and how they interact with the Bitcoin network:

Full node
: A full node is a program that validates the entire history of Bitcoin transactions (every transaction by every user, ever). Optionally, full nodes can also store previously validated transactions and serve data to other Bitcoin programs, either on the same computer or over the internet. A full node uses substantial computer resources—about the same as watching an hour-long streaming video for each day of Bitcoin transactions—but the full node offers complete autonomy to its users.

Lightweight client
: A lightweight client, also known as a simplified-payment-verification (SPV) client, connects to a full node or other remote server for receiving and sending Bitcoin transaction information, but stores the user wallet locally, partially validates the transactions it receives, and independently creates outgoing transactions.

Third-party API client
: A third-party API client is one that interacts with Bitcoin through a third-party system of APIs rather than by connecting to the Bitcoin network directly. The

wallet may be stored by the user or by third-party servers, but the client trusts the remote server to provide it with accurate information and protect its privacy.

Bitcoin is a peer-to-peer (P2P) network. Full nodes are the *peers:* each peer individually validates every confirmed transaction and can provide data to its user with complete authority. Lightweight wallets and other software are *clients:* each client depends on one or more peers to provide it with valid data. Bitcoin clients can perform secondary validation on some of the data they receive and make connections to multiple peers to reduce their dependence on the integrity of a single peer, but the security of a client ultimately relies on the integrity of its peers.

Who controls the keys

A very important additional consideration is *who controls the keys*. As we will see in subsequent chapters, access to bitcoins is controlled by "private keys," which are like very long PINs. If you are the only one to have control over these private keys, you are in control of your bitcoins. Conversely, if you do not have control, then your bitcoins are managed by a third-party who ultimately controls your funds on your behalf. Key management software falls into two important categories based on control: *wallets*, where you control the keys, and the funds and accounts with custodians where some third-party controls the keys. To emphasize this point, I (Andreas) coined the phrase: *Your keys, your coins. Not your keys, not your coins.*

Combining these categorizations, many Bitcoin wallets fall into a few groups, with the three most common being desktop full node (you control the keys), mobile lightweight wallet (you control the keys), and web-based accounts with third parties (you don't control the keys). The lines between different categories are sometimes blurry, as software runs on multiple platforms and can interact with the network in different ways.

Quick Start

Alice is not a technical user and only recently heard about Bitcoin from her friend Joe. While at a party, Joe is enthusiastically explaining Bitcoin to everyone around him and is offering a demonstration. Intrigued, Alice asks how she can get started with Bitcoin. Joe says that a mobile wallet is best for new users and he recommends a few of his favorite wallets. Alice downloads one of Joe's recommendations and installs it on her phone.

When Alice runs her wallet application for the first time, she chooses the option to create a new Bitcoin wallet. Because the wallet she has chosen is a noncustodial wallet, Alice (and only Alice) will be in control of her keys. Therefore, she bears responsibility for backing them up, since losing the keys means she loses access to

her bitcoins. To facilitate this, her wallet produces a *recovery code* that can be used to restore her wallet.

Recovery Codes

Most modern noncustodial Bitcoin wallets will provide a recovery code for their user to back up. The recovery code usually consists of numbers, letters, or words selected randomly by the software, and is used as the basis for the keys that are generated by the wallet. See Table 1-1 for examples.

Table 1-1. Sample recovery codes

Wallet	Recovery code
BlueWallet	(1) media (2) suspect (3) effort (4) dish (5) album (6) shaft (7) price (8) junk (9) pizza (10) situate (11) oyster (12) rib
Electrum	nephew dog crane clever quantum crazy purse traffic repeat fruit old clutch
Muun	LAFV TZUN V27E NU4D WPF4 BRJ4 ELLP BNFL

A recovery code is sometimes called a "mnemonic" or "mnemonic phrase," which implies you should memorize the phrase, but writing the phrase down on paper takes less work and tends to be more reliable than most people's memories. Another alternative name is "seed phrase" because it provides the input ("seed") to the function that generates all of a wallet's keys.

If something happens to Alice's wallet, she can download a new copy of her wallet software and enter this recovery code to rebuild the wallet database of all the onchain transactions she's ever sent or received. However, recovering from the recovery code will not by itself restore any additional data Alice entered into her wallet, such as the labels she associated with particular addresses or transactions. Although losing access to that metadata isn't as important as losing access to money, it can still be important in its own way. Imagine you need to review an old bank or credit card statement and the name of every entity you paid (or who paid you) has been blanked out. To prevent losing metadata, many wallets provide an additional backup feature beyond recovery codes.

For some wallets, that additional backup feature is even more important today than it used to be. Many Bitcoin payments are now made using *offchain* technology, where not every payment is stored in the public blockchain. This reduces user's costs and improves privacy, among other benefits, but it means that a mechanism like recovery codes that depends on onchain data can't guarantee recovery of all of a user's bitcoins. For applications with offchain support, it's important to make frequent backups of the wallet database.

Of note, when receiving funds to a new mobile wallet for the first time, many wallets will often re-verify that you have securely backed-up your recovery code. This can range from a simple prompt to requiring the user to manually re-enter the code.

Although many legitimate wallets will prompt you to re-enter your recovery code, there are also many malware applications that mimic the design of a wallet, insist you enter your recovery code, and then relay any entered code to the malware developer so they can steal your funds. This is the equivalent of phishing websites that try to trick you into giving them your bank passphrase. For most wallet applications, the only times they will ask for your recovery code are during the initial set up (before you have received any bitcoins) and during recovery (after you lost access to your original wallet). If the application asks for your recovery code any other time, consult with an expert to ensure you aren't being phished.

Bitcoin Addresses

Alice is now ready to start using her new Bitcoin wallet. Her wallet application randomly generated a private key (described in more detail in "Private Keys" on page 55) that will be used to derive Bitcoin addresses that direct to her wallet. At this point, her Bitcoin addresses are not known to the Bitcoin network or "registered" with any part of the Bitcoin system. Her Bitcoin addresses are simply numbers that correspond to her private key that she can use to control access to the funds. The addresses are generated independently by her wallet without reference or registration with any service.

There are a variety of Bitcoin addresses and invoice formats. Addresses and invoices can be shared with other Bitcoin users who can use them to send bitcoins directly to your wallet. You can share an address or invoice with other people without worrying about the security of your bitcoins. Unlike a bank account number, nobody who learns one of your Bitcoin addresses can withdraw money from your wallet—you must initiate all spends. However, if you give two people the same address, they will be able to see how many bitcoins the other person sent you. If you post your address publicly, everyone will be able to see how much bitcoin other people sent to that address. To protect your privacy, you should generate a new invoice with a new address each time you request a payment.

Receiving Bitcoin

Alice uses the *Receive* button, which displays a QR code, shown in Figure 1-1.

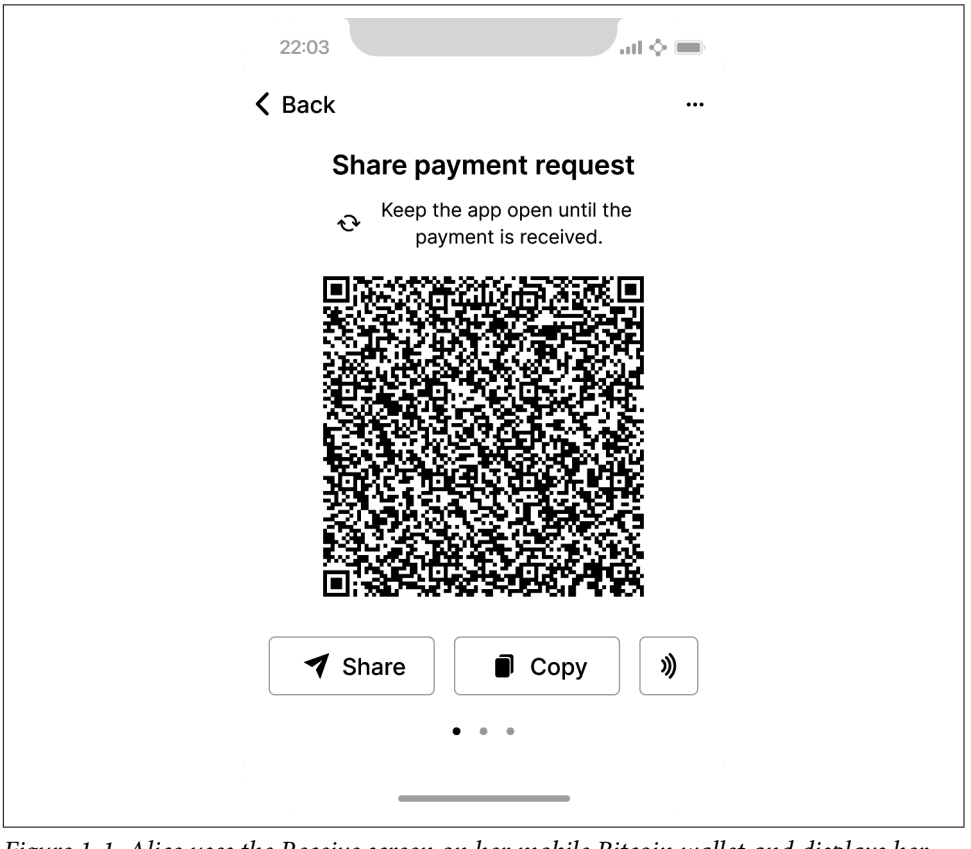


Figure 1-1. Alice uses the Receive screen on her mobile Bitcoin wallet and displays her address in a QR code format.

The QR code is the square with a pattern of black and white dots, serving as a form of barcode that contains the same information in a format that can be scanned by Joe's smartphone camera.

Any funds sent to the addresses in this book will be lost. If you want to test sending bitcoins, please consider donating it to a bitcoin-accepting charity.

Getting Your First Bitcoin

The first task for new users is to acquire some bitcoin.

Bitcoin transactions are irreversible. Most electronic payment networks such as credit cards, debit cards, PayPal, and bank account transfers are reversible. For someone selling bitcoin, this difference introduces a very high risk that the buyer will reverse the electronic payment after they have received bitcoin, in effect defrauding the seller. To mitigate this risk, companies accepting traditional electronic payments in return for bitcoin usually require buyers to undergo identity verification and credit-worthiness checks, which may take several days or weeks. As a new user, this means you cannot buy bitcoin instantly with a credit card. With a bit of patience and creative thinking, however, you won't need to.

Here are some methods for acquiring bitcoin as a new user:

- Find a friend who has bitcoins and buy some from him or her directly. Many Bitcoin users start this way. This method is the least complicated. One way to meet people with bitcoins is to attend a local Bitcoin meetup listed at Meetup.com.
- Earn bitcoin by selling a product or service for bitcoin. If you are a programmer, sell your programming skills. If you're a hairdresser, cut hair for bitcoins.
- Use a Bitcoin ATM in your city. A Bitcoin ATM is a machine that accepts cash and sends bitcoins to your smartphone Bitcoin wallet.
- Use a Bitcoin currency exchange linked to your bank account. Many countries now have currency exchanges that offer a market for buyers and sellers to swap bitcoins with local currency. Exchange-rate listing services, such as BitcoinAverage, often show a list of Bitcoin exchanges for each currency.

One of the advantages of Bitcoin over other payment systems is that, when used correctly, it affords users much more privacy. Acquiring, holding, and spending bitcoin does not require you to divulge sensitive and personally identifiable information to third parties. However, where bitcoin touches traditional systems, such as currency exchanges, national and international regulations often apply. In order to exchange bitcoin for your national currency, you will often be required to provide proof of identity and banking information. Users should be aware that once a Bitcoin address is attached to an identity, other associated Bitcoin transactions may also become easy to identify and track—including transactions made earlier. This is one reason many users choose to maintain dedicated exchange accounts independent from their wallets.

Alice was introduced to Bitcoin by a friend, so she has an easy way to acquire her first bitcoins. Next, we will look at how she buys bitcoins from her friend Joe and how Joe sends the bitcoins to her wallet.

Finding the Current Price of Bitcoin

Before Alice can buy bitcoin from Joe, they have to agree on the *exchange rate* between bitcoin and US dollars. This brings up a common question for those new to Bitcoin: "Who sets the price of bitcoins?" The short answer is that the price is set by markets.

Bitcoin, like most other currencies, has a *floating exchange rate*. That means that the value of bitcoin fluctuates according to supply and demand in the various markets where it is traded. For example, the "price" of bitcoin in US dollars is calculated in each market based on the most recent trade of bitcoins and US dollars. As such, the price tends to fluctuate minutely several times per second. A pricing service will aggregate the prices from several markets and calculate a volume-weighted average representing the broad market exchange rate of a currency pair (e.g., BTC/USD).

There are hundreds of applications and websites that can provide the current market rate. Here are some of the most popular:

Bitcoin Average
: A site that provides a simple view of the volume-weighted average for each currency.

CoinCap
: A service listing the market capitalization and exchange rates of hundreds of cryptocurrencies, including bitcoins.

Chicago Mercantile Exchange Bitcoin Reference Rate
: A reference rate that can be used for institutional and contractual reference, provided as part of investment data feeds by the CME.

In addition to these various sites and applications, some Bitcoin wallets will automatically convert amounts between bitcoin and other currencies.

Sending and Receiving Bitcoin

Alice has decided to buy 0.001 bitcoins. After she and Joe check the exchange rate, she gives Joe an appropriate amount of cash, opens her mobile wallet application, and selects Receive. This displays a QR code with Alice's first Bitcoin address.

Joe then selects Send on his smartphone wallet and opens the QR code scanner. This allows Joe to scan the barcode with his smartphone camera so that he doesn't have to type in Alice's Bitcoin address, which is quite long.

Joe now has Alice's Bitcoin address set as the recipient. Joe enters the amount as 0.001 bitcoins (BTC); see Figure 1-2. Some wallets may show the amount in a different denomination: 0.001 BTC is 1 millibitcoin (mBTC) or 100,000 satoshis (sats).

Some wallets may also suggest Joe enter a label for this transaction; if so, Joe enters "Alice". Weeks or months from now, this will help Joe remember why he sent these 0.001 bitcoins. Some wallets may also prompt Joe about fees. Depending on the wallet and how the transaction is being sent, the wallet may ask Joe to either enter a transaction fee rate or prompt him with a suggested fee (or fee rate). The higher the transaction fee, the faster the transaction will be confirmed (see "Confirmations" on page 14).

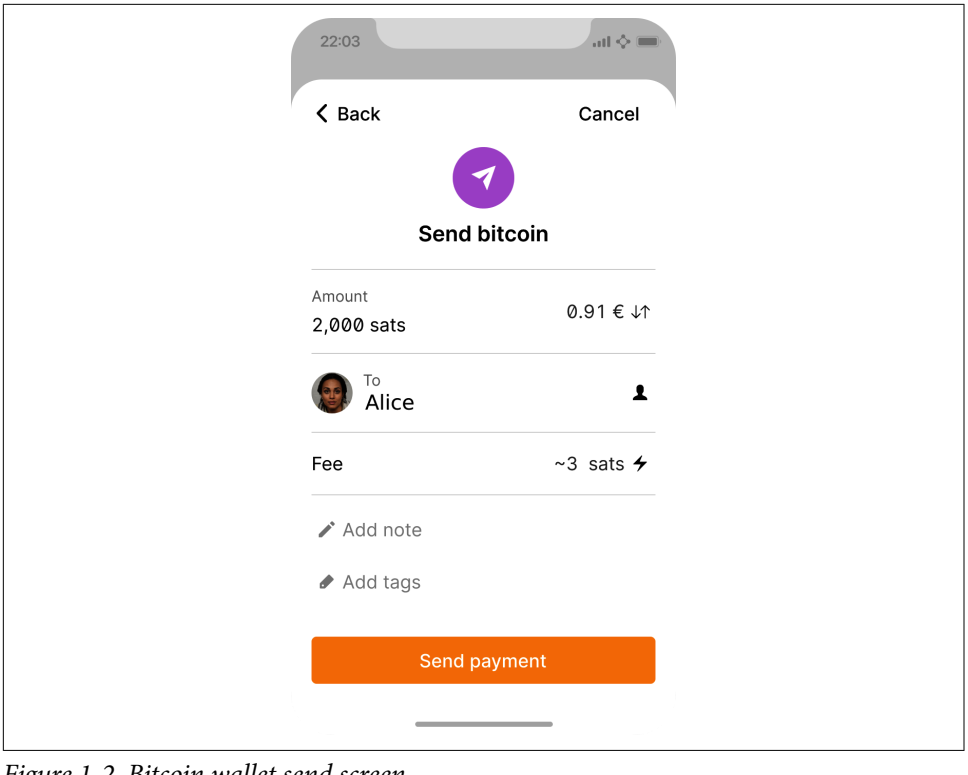


Figure 1-2. Bitcoin wallet send screen.

Joe then carefully checks to make sure he has entered the correct amount, because he is about to transmit money and mistakes will soon become irreversible. After double-checking the address and amount, he presses Send to transmit the transaction. Joe's mobile Bitcoin wallet constructs a transaction that assigns 0.001 BTC to the address provided by Alice, sourcing the funds from Joe's wallet, and signing the transaction with Joe's private keys. This tells the Bitcoin network that Joe has authorized a transfer of value to Alice's new address. As the transaction is transmitted via the