# 本书介绍

**Introduction to the book**

There’s a lot of excitement about Bitcoin and cryptocurrencies. Optimists claim that Bitcoin will fundamentally alter payments, economics, and even politics around the world. Pessimists claim Bitcoin is inherently broken and will suffer an inevitable and spectacular collapse.

“比特币”和“加密货币”有很多令人兴奋的地方。

乐观主义者认为：比特币将从根本上改变世界各地的支付、经济，甚至政治。

悲观主义者认为：比特币本质上是破碎的，将不可避免地遭到严重崩溃。

Underlying these differing views is significant confusion about what Bitcoin is and how it works. We wrote this book to help cut through the hype and get to the core of what makes Bitcoin unique.

这些不同观点的背后是对如下的严重混乱：比特币是什么？比特币是如何运作的？

我们写这本书是为了帮助你跳出炒作，理解本质：比特币有什么独特之处？

To really understand what is special about Bitcoin, we need to understand how it works at a technical level. Bitcoin truly is a new technology and we can only get so far by explaining it through simple analogies to past technologies.

要真正理解比特币的特殊之处，就需要理解它在技术层面上是如何运作的。

比特币确实是一个新技术，我们只能通过简单的类比（对比过去的技术）来解释它。

We’ll assume that you have a basic understanding of computer science — how computers work, data structures and algorithms, and some programming experience. If you’re an undergraduate or graduate student of computer science, a software developer, an entrepreneur, or a technology hobbyist, this textbook is for you.

我们假设你对计算机科学有基本的了解：计算机如何工作、数据结构和算法、一些编程经验。

如果你是计算机科学的本科生或研究生、软件开发者、企业家、技术爱好者，那本书就是为你写的。

In this book we’ll address the important questions about Bitcoin.

在本书中，我们将讨论“比特币”的一些重要问题：

|  |  |  |
| --- | --- | --- |
| 1 | How does Bitcoin work? | 比特币是如何运作的？ |
| 2 | What makes it different? | 比特币有什么独特之处？ |
| 3 | How secure are your bitcoins? | 你的比特币的安全性如何？ |
| 4 | How anonymous are Bitcoin users? | 比特币用户是如何匿名的？ |
| 5 | What applications can we build using Bitcoin as a platform? | 在比特币平台上能够构建什么应用？ |
| 6 | Can cryptocurrencies be regulated? | 加密货币能被监管吗？ |
| 7 | If we were designing a new cryptocurrency today, what would we change? | 如何我们今天设计一个新的加密货币，我们会做些什么修改？ |
| 8 | What might the future hold? | 未来会怎么样？ |

Each chapter has a series of homework questions to help you understand these questions at a deeper level. In addition, there is a series of programming assignments in which you’ll implement various components of Bitcoin in simplified models. If you’re an auditory learner, most of the material of this book is also available as a series of video lectures. You can find all these on our ​ Coursera course​ . You should also supplement your learning with information you can find online including the Bitcoin wiki, forums, and research papers, and by interacting with your peers and the Bitcoin community.

每一章都有一些问题作业，帮助你在更深层次上理解这些问题。

此外，还有一些编程作业，你要实现比特币简化模型的各种组件。

如果你习惯于听课学习，本书的大部分材料在视频讲座中也有。可以在Coursera课程中找到这些内容。

你还应该使用网上信息补充学习：比特币维基、论坛和研究论文，以及与伙伴和比特币社区进行交流。

<https://press.princeton.edu/titles/10908.html>

After reading this book, you’ll know everything you need to be able to separate fact from fiction when reading claims about Bitcoin and other cryptocurrencies. You’ll have the conceptual foundations you need to engineer secure software that interacts with the Bitcoin network. And you’ll be able to integrate ideas from Bitcoin into your own projects.

读完本书后，当你阅读有关“比特币”和“其它加密货币”的内容时，就能够区分“事实”和“虚构”。

如果你要编写与比特币网络进行交互的软件，你就有了编写安全软件的概念基础。

你还能够把比特币中的思想整合到自己的项目中。

**A note of thanks**

We’re immensely grateful to the students who helped develop programming assignments and to everyone who provided feedback on the drafts of this book.

我们非常感谢帮助开发编程作业的学生，以及对本书草稿提供建议的人。

**Princeton students** Shivam Agarwal, Miles Carlsten, Paul Ellenbogen, Pranav Gokhale, Alex Iriza, Harry Kalodner, and Dillon Reisman, and **Stanford students** Allison Berke, Benedikt Bünz, and Alex Leishman deserve special praise.

We’re also thankful to Dan Boneh and Albert Szmigielski.

# 0序言:通向比特币的漫长道路

**Preface — The Long Road to Bitcoin**

**Jeremy Clark**

The path to Bitcoin is littered with the corpses of failed attempts. I’ve compiled a list of about a hundred cryptographic payment systems, both e-cash and credit card based technologies, that are notable in some way. Some are academic proposals that have been well cited while others are actual systems that were deployed and tested. Of all the names on this list, there’s probably only one that you recognize — PayPal. And PayPal survived only because it quickly pivoted away from its original idea of cryptographic payments on hand-held devices!

在通往比特币的道路上，到处都是失败的尝试。

我汇编了一个列表，列出了大约100个加密支付系统，它们是基于电子现金（e-cash）或信用卡的技术，

在某种程度上都有显著成就。

其中一些是被广泛引用的学术研究，其它则是实际做过部署和测试的系统。

在这个列表中，你可能只听说过一个：PayPal。

只有PayPal幸存了下来，是因为它很快就纠正了最初的想法：在移动设备上做加密支付。



**Table 1: Notable electronic payment systems and proposals**

**表1：著名的电子支付系统和建议**

There’s a lot to learn from this history.

Where do the ideas in Bitcoin come from?

Why do some technologies survive while many others die?

What does it take for complex technical innovations to be successfully commercialized?

从这段历史中，我们可以学到很多东西：

* 比特币的思想来自哪里？
* 为什么有些技术幸存下来，而很多技术都消亡了？
* 如何使复杂的技术创新成功地实现商业化？

If nothing else, this story will give you an appreciation of how remarkable it is that we finally have a real, working payment mechanism that’s native to the Internet.

别的不说，这个故事会让你理解这是多么了不起：

我们终于有了一个切实可行的、基于互联网的支付机制。

## 0.1传统金融体系

Traditional financial arrangements

Back in time before there were governments, before there was currency, one system that worked for acquiring goods was barter. Let’s say Alice wants a tool and Bob wants medicine. If each of them happen to have what the other person needs, then they can swap and both satisfy their needs.

在还没有政府的时代，在货币出现之前，一种可以获取物品的系统是“物物交换”。

例如，Alice想要一个工具，Bob想要一些药物。

如果他俩碰巧都有对方需要的东西，那么就可以交换，这同时满足他俩的需求。

On the other hand, let’s say Alice has food that she’s willing to trade for a tool, while Bob, who has a tool, doesn’t have any need for food. He wants medicine instead. Alice and Bob can’t trade with each other, but if there’s a third person, Carol, who has medicine that she’s willing to trade for food, then it becomes possible to arrange a three-way swap where everyone gets what they need.

进一步，假设Alice有食物，她想要一个工具；Bob有一个工具，但他不需要食物，他想要一些药物。

Alice和Bob就没法做交易，但如果Carol有药物，并且她想要食物，那就可能实现三方交换，这样，每个人都能得到需要的东西。

|  |  |  |
| --- | --- | --- |
|  | **有** | **想要** |
| Alice | 食物 | 工具 |
| Bob | 工具 | 药物 |
| Carol | 药物 | 食物 |

The drawback, of course, is coordination — arranging a group of people, whose needs and wants align, in the same place at the same time. Two systems emerged to solve coordination: credit and cash. Historians, anthropologists, and economists debate which of the two developed first, but that’s immaterial for our purposes.

当然，这个系统的缺点是需要协调：在同一时间、同一地点，安排一组人，他们的供需能够一致。

为了解决协调问题，这就出现了两种系统：信用（credit）、现金（cash）。

历史学家、人类学家、经济学家会争论哪个系统先出现，但这对我们的来说并不重要。

In a credit-based system, in the example above, Alice and Bob would be able to trade with each other. Bob would give Alice the tool and Bob gets a favor that’s owed to him. In other words, Alice has a debt that she needs to settle with Bob some time in the future. Alice’s material needs are now satisfied, but she has a debt that she’d like to cancel, so that’s her new “want”. If Alice encounters Carol in the future, Alice can trade her food for Carol’s medicine, then go back to Bob with the medicine and cancel the debt.

继续用前面的例子来说明。

在“信用系统”中，Alice和Bob能进行交易。

Bob把工具给Alice，Alice就欠了Bob一个债务（favor）。

现在，Alice的需求得到了满足，但她想偿还所欠债务，这是她的“新需求”。

如果Alice遇到了Carol，Alice可以用食物换Carol的药物，然后把药物给Bob，就还清了债务。

ddk说明：

* Alice得到了Bob的工具，欠Bob一个债务，要用药物偿还
* Alice用食物换取了Carol的药物
* Alice把药物给了Bob，就还清了债务

On the other hand, in a cash-based system, Alice would buy the tool from Bob. Later, she might sell her food to Carol, and Carol can sell her medicine to Bob, completing the cycle. These trades can happen in any order, provided that the buyer in each transaction has cash on hand. In the end, of course, it’s as if no money ever changed hands.

在“现金系统”中，Alice买Bob的工具，Bob买Carol的药物，Carol买Alice的食物，这就完成了循环。

这些交易可以按任何顺序进行，前提是，每个交易的买方手里都要有现金。

当然，最后好像“钱”并没有转手。（ddk：各自手中的钱并没有增多或减少）

ddk说明：

* 以“现金”为媒介进行交换
* 每种物品都可以换成钱，钱可以买任何物品

Neither system is clearly superior. A cash-based system needs to be “bootstrapped” with some initial allocation of cash, without which no trades can occur. A credit-based system doesn’t need bootstrapping, but the drawback is that anyone who’s owed a debt is taking on some risk. There’s a chance that the other person never comes back to settle the debt.

这两个系统不分伯仲，没有哪个明显优于另一个。

* 现金系统：需要“初始分配一些现金”来启动这个系统的运转，没有“现金”就没法做交易。
* 信用系统：不需要启动运转，但缺点是债权人要冒一些风险，因为债务人可能不偿还债务。

Cash also allows us to be precise about how much something is worth. If you’re bartering, it’s hard to say if a tool is worth more than medicine or medicine is worth more than food. Cash lets us use numbers to talk about value. That’s why we use a blended system today — even when we’re using credit, we measure debt in the amount of cash it would take to settle it.

“现金”还能让我们准确衡量物品的价值。

如果你在做物物交换，很难说工具比药物更有价值，或药品比食物更有价值。

现金让我们可以使用数字来谈论价值。

因此，我们今天使用的是一种混合系统：即使我们在使用信用时，也用现金来计算要偿还的债务。

These ideas come up in many contexts, especially online systems where users trade virtual goods of some kind. For example, peer-to-peer file-sharing networks must deal with the problem of “freeloaders,” that is, users who download files without sharing in turn. While swapping files might work, there is also the issue of coordination: finding the perfect person who has exactly the file you want and wants exactly the file you have. In projects like MojoNation and academic proposals like Karma, users get some initial allocation of virtual cash that they must spend to receive a file and earn when they send a copy of a file to another user. In both cases, one or more central servers help keep track of users’ balances and may offer exchange services between their internal currency and traditional currency. While MojoNation did not survive long enough to implement such an exchange, it became the intellectual ancestor of some protocols used today: BitTorrent and Tahoe-LAFS.

这些想法（物物交换、现金、信用）出现在许多情况下，尤其是网上系统，用户交易的是某种虚拟商品。（ddk：最后都落到了使用“现金”来实现交换的想法上）

例如，“P2P文件共享网络”必须解决这个问题：有的用户只下载文件，而不上传文件。

虽然可以交换文件，但也有协调的问题：找到合适的人，他有你想要的文件，你也有他想要的文件。

在“MojoNation项目”和“Karma学术建议”中，用户获得一些初始分配的“虚拟现金”，用户下载文件时必须花费虚拟现金，上传文件时可以获得虚拟现金。

在这两个例子中，都有一个或多个中央服务器来跟踪用户余额，并可提供“内部货币”和“传统货币”的交易服务。

虽然MojoNation存在的时间不长，没能实现这样的交易，但它是当今使用的一些协议的先祖：BitTorrent、Tahoe-LAFS。

**【总结】**

|  |  |  |
| --- | --- | --- |
| **交易系统** | **优点** | **缺点** |
| **物物交换** |  | 多方交换时，需要相互协调：在同一时间、同一地点，安排一组人，他们的供需能够一致 |
| **信用** | 不需要启动运转 | 债权人要冒一些风险，因为债务人可能不偿还债务。 |
| **现金** | 现金能让我们准确衡量物品的价值。  避免了买家拖欠债务的可能性。 | 需要“初始分配一些现金”来启动这个系统的运转，没有“现金”就没法做交易。 |
| **混合系统** | 我们今天使用的是一种混合系统：  即使我们在使用信用时，也用现金来计算要偿还的债务。 | |

这些想法（物物交换、现金、信用）出现在许多情况下，尤其是网上系统，用户交易的是某种虚拟商品。

例如：“P2P文件共享网络”使用“虚拟现金”来实现文件交换。

## 0.2网络信用卡的弊端

**The trouble with credit cards online**

Credit and cash are fundamental ideas, to the point that we can sort the multitude of electronic payment methods into two piles. Bitcoin is obviously in the “cash” pile, but let’s look at the other one first.

“信用”和“现金”是基本的思想，据此可以把大量的“电子支付方法”分为两类。

比特币显然属于“现金”类，但让我们先看看“信用”类。

Credit card transactions are the dominant payment method that is used on the web today. If you’ve ever bought something from an online seller such as Amazon, you know how the arrangement goes. You type in your credit card details, you send it to Amazon, and then Amazon turns around with these credit card details and they talk to the “system”—a financial system involving processors, banks, credit card companies, and other intermediaries.

信用卡交易是当今网上使用的主流支付方式。

如果你在Amazon买过东西，你就知道这个体系是怎么运转的。

你输入信用卡信息，发送给Amazon，Amazon用这些信息与“系统”进行对话，它是一个金融系统，涉及processor、银行、信用卡公司、其它中介机构。

On the other hand, if you use something like PayPal, what you see is an intermediary architecture. There’s a company that sits between you and the seller, so you send your credit card details to this intermediary, which approves the transaction and notifies the seller. The intermediary will settle its balance with the seller at the end of each day.

另一方面，如果你用的是PayPal，你看到的是一种中介体系结构。

有一个中介在你和卖家之间，你把信用卡信息发给这个中介，它批准交易并通知卖家。

这个中介每天会与卖家进行结算。

What you gain from this architecture is that you don’t have to give the seller your credit card details, which can be a security risk. You might not even have to give the seller your identity, which would improve your privacy as well. The downside is that you lose the simplicity of interacting directly with the seller. Both you and the seller might have to have an account with the same intermediary.

在中介体系结构中，你不必把信用卡信息提供给卖家，提供给卖家可能是一个安全风险。

你可能甚至不必向卖家提供你的身份，这也会提高你的隐私。

缺点是，你失去了与卖家直接互动的简单性。

你和卖家都必须在中介那里有一个账户。

Today most of us are comfortable with giving out our credit card information when shopping online, or at least we’ve grudgingly accepted it. We’re also used to companies collecting data about our online shopping and browsing activity. But in the 1990s, the web was new, standards for protocol-level encryption were just emerging, and these concerns made consumers deeply uncertain and hesitant. In particular, it was considered crazy to hand over your credit card details to online vendors of unknown repute over an insecure channel. In such an environment, there was a lot of interest in the intermediary architecture.

今天，在网上购物时，多数人习惯了提供信用卡信息，或至少勉强接受了。

我们也习惯了公司收集我们的数据：网上购物和浏览活动。

但在1990年代，网络还是新生事物，协议级加密的标准刚刚出现，这些担忧使消费者深陷不确定性和犹豫。

特别是，这被认为是很疯狂：在一个不安全的通道上，把自己的信用卡信息提供给不知名的网上卖家。

在这样的环境中，人们对中介体系结构更感兴趣。

A company called FirstVirtual was an early payment intermediary, founded in 1994. Incidentally, they were one of the first companies to set up a purely virtual office with employees spread across the country and communicating over the Internet — hence the name.

FirstVirtual公司成立于1994年，是一家早期的支付中介公司。

他们也是最早设立纯虚拟办公室的公司之一，员工遍布全国，通过互联网进行沟通。

FirstVirtual’s proposed system was a little like PayPal’s current system but preceded it by many years. As a user you’d enroll with them and provide your credit card details. When you want to buy something from a seller, the seller contacts FirstVirtual with the details of the requested payment, FirstVirtual confirms these details with you, and if you approve your credit card gets billed.

FirstVirtual提出的系统有点像PayPal，但比它早很多年。

用户向FirstVirtual注册，并提供信用卡信息。

当你要买东西时，卖家会联系FirstVirtual，告知支付信息，FirstVirtual向你确认这些信息，以及你是否同意使用信用卡进行支付。

But two details are interesting. First, all of this communication happened over email; web browsers back in the day were just beginning to universally support encryption protocols like HTTPS, and the multi-party nature of payment protocol added other complexities. (Other intermediaries took the approach of encoding information into URLs or using a custom encryption protocol on top of HTTP.) Second, the customer would have ninety days to dispute the charge, and the merchant would receive the money only after three months! Today the merchant does get paid immediately, but, there still is the risk that the customer will file a chargeback or dispute the credit card statement. If that happens, the merchant will have to return the payment to the credit card company.

但有两个细节很有趣。

* 所有沟通都是通过email进行的。那时的Web浏览器刚开始普遍支持诸如HTTPS之类的加密协议，而支付协议的多方性质增加了其它复杂性。

（其它中介采用的方法是，将信息编码到URLs中，或在HTTP上层使用一个自定义加密协议。）

* 客户有90天时间来质疑收费，商家只有在90天后才收到这笔钱！

今天，商家可以立即得到付款，但仍然有风险：客户可能索回付款，或对信用卡账单提出申诉。

如果发生这种情况，商家就必须将款项退还给信用卡公司。

In the mid ‘90s there was a competing approach to the intermediary architecture which we’ll call the SET architecture. SET also avoids the need for customers to send credit card information to merchants, but it additionally avoids the user having to enroll with the intermediary. In SET, when you are ready to make a purchase, your browser passes your view of the transaction details to a shopping application on your computer which, together with your credit card details, encrypts it in such a way that only the intermediary can decrypt it, and no one else can (including the seller). Having encrypted your data it this way, you can send it to the seller knowing that it’s secure. The seller blindly forwards the encrypted data to the intermediary — along with their own view of the transaction details. The intermediary decrypts your data and approves the transaction only if your view matches the seller’s view.

在90年代中期，有一种与“中介体系结构”竞争的方法，称为“SET体系结构”。

SET也避免了客户向商家提供信用卡信息，但它还不需要：用户向中介进行注册。

SET的购物流程：

* 用户操作：
  + 浏览器把你的“交易细节视图”传给本机的“购物应用程序”
  + 购物应用程序再加上你的信用卡信息，对这些信息进行加密（只有中介可以解密）。
  + 你把加密数据发送给商家。
* 商家操作：商家把数据转给中介，包括商家自己的交易细节视图。
* 中介操作：中介解密你的数据，只有当你和商家的视图匹配时，才批准这个交易。

SET was a standard developed by VISA and MasterCard, together with many technology heavyweights of the day: Netscape, IBM, Microsoft, Verisign, and RSA. It was an umbrella specification that unified several existing proposals.

SET是由下列公司开发的一个标准：VISA、MasterCard、Netscape、IBM、微软、Verisign、RSA。

这是一个大的规范，它统一了几个已有的提案。

One company that implemented SET was called CyberCash. It was an interesting company in many ways. In addition to credit card payment processing, they had a digital cash product called CyberCoin. This was a micropayment system — intended for small payments such as paying a few cents to read an online newspaper article. That meant that you’d probably never have more than $10 in your CyberCoin account at any time. Yet, amusingly, they were able to get U.S. government (FDIC) insurance for each account for up to $100,000.

CyberCash公司实现了SET，这是一个在许多方面都很有趣的公司。

除了信用卡支付处理之外，他们还有一个数字现金产品，称为CyberCoin。

这是一个小额支付系统，例如支付几美分来阅读网上报纸的一篇文章。

这意味着，在你的CyberCoin账户中一般不会超过10美元。

但有趣的是，他们为每个账户获得了最高可达10万美元的美国政府（FDIC）保险。

There’s more. Back when CyberCash operated, there was a misguided — and now abandoned — U.S. government restriction on the export of cryptography, which was considered a weapon. That meant software that incorporated meaningful encryption couldn’t be offered for download to users in other countries. However, CyberCash was able to get a special exemption for their software from the Department of State. The government’s argument was that extracting the encryption technology out of CyberCash’s software would be harder than writing the crypto from scratch.

还有更多。

在CyberCash运营的时候，对密码系统的出口有一个误导的限制（现已取消），当时认为密码系统是一种武器。这意味着，包含重要加密的软件不能提供给其它国家的用户下载。

但是，CyberCash让国务院特别豁免了他们的软件。

政府的论点是：从CyberCash的软件中提取加密技术，比从零开始编写还要困难。

Finally, CyberCash has the dubious distinction of being one of the few companies affected by the Y2K bug — it caused their payment processing software to double-bill some customers. They later went bankrupt in 2001. Their intellectual property was acquired by Verisign who then turned around and sold it to PayPal where it lives today.

最后，CyberCash受到了千年虫的影响，导致他们的支付处理软件对有些客户收了双倍的钱。

CyberCash在2001年破产，知识产权被Verisign收购，Verisign又转手卖给了PayPal，PayPal今天还活着。

Why didn’t SET work? The fundamental problem has to do with certificates. A certificate is a way to securely associate a cryptographic identity, that is, a public key, with a real-life identity. It’s what a website needs to obtain, from companies like Verisign that are called certification authorities, in order to show up as secure in your browser (typically indicated by a lock icon). Putting security before usability, CyberCash and SET decided that not only would processors and merchants in their system have to get certificates, all users would have to get one as well. Getting a certificate is about as pleasant as doing your taxes, so the system was a disaster.

为什么SET不行？根本的问题是与“证书”有关。

证书是一种方法，它将“加密身份（即公钥）”安全地与“实际身份”关联起来。

网站需要从Verisign这类公司（称为“认证机构”）获取证书，目的是在你的浏览器判断它是安全的（通常由一个锁图标指示）。

CyberCash和SET认为安全性重于可使用性，所以他们要求，不仅processors和商家要获得证书，所有用户也要获得证书。

获得证书就像纳税一样繁琐，所以这个系统是一场灾难。

Over the decades, mainstream users have said a firm and collective ‘no’ to any system that requires end-user certificates, and such proposals have now been relegated to academic papers. Bitcoin deftly sidesteps this hairy problem by avoiding real-life identities altogether. In Bitcoin, public keys themselves are the identities by which users are known, as we’ll see in Chapter 1.

在过去几十年里，主流用户都排斥任何需要最终用户证书的系统，这些提案现在已被沦为了学术论文。

比特币巧妙地回避了这个问题，它完全避免了真实身份。

在比特币中，“公钥”就是“用户身份”。

In the mid 90s, when SET was being standardized, the World Wide Web Consortium was also looking at standardizing financial payments. They wanted to do it by extending the HTTP protocol instead so that users wouldn’t need extra software for transactions—they could just use their browser. In fact, they had a very general proposal for how you might extend the protocol, and one of the use cases that they had was doing payments. This never happened -- the whole extension framework was never deployed in any browsers. In 2015, almost two decades later, the W3C has announced that it wants to take another crack at it, and that Bitcoin will be part of that standardization this time around. Given all the past failures, however, I won’t be holding my breath.

在90年代中期，当SET被标准化时，Web联盟也想对金融支付进行标准化。

他们想通过扩展HTTP协议来实现，这样用户在做交易时就不需要额外的软件，只需要用浏览器。

实际上，他们有一个更通用的提案“如何扩展这个协议”，其中一个用例就是支付。

但这从未实现过，整个扩展框架从未部署在任何浏览器中。

2015年，Web联盟宣布它想采用另一种方法，这次，比特币将成为标准化的一部分。

然而，鉴于以往的失败，我对此并不乐观。

**【总结】**

|  |  |  |
| --- | --- | --- |
| **信用类电子支付方法** | | |
| **1非中介体系结构**  **例如：**  **Amazon** | **处理** | 1. 你输入信用卡信息，发送给Amazon 2. Amazon用这些信息与“系统”进行对话，它是一个金融系统，涉及processor、银行、信用卡公司、其它中介机构。 |
| **优点** | 直接与卖家互动 |
| **缺点** | 信用卡信息提供给商家可能是一个风险 |
| **2中介体系结构**  **例如：FirstVirtual**  **PayPal** | **处理** | 有一个中介在你和卖家之间，你把信用卡信息发给这个中介，它批准交易并通知卖家。这个中介每天会与卖家进行结算。  FirstVirtual公司成立于1994年，是一家早期的支付中介公司。FirstVirtual提出的系统有点像PayPal，但比它早很多年。  用户向FirstVirtual注册，并提供信用卡信息。当你要买东西时，卖家会联系FirstVirtual，告知支付信息，FirstVirtual向你确认这些信息，以及你是否同意使用信用卡进行支付。  两个细节：所有沟通都是通过email进行的。 客户有90天时间来质疑收费，商家只有在90天后才收到这笔钱！ |
| **优点** | 不必向卖家提供你的身份，这也会提高你的隐私。 |
| **缺点** | 你和卖家都必须在中介那里有一个账户。  你失去了与卖家直接互动的简单性。 |
| **3 SET体系结构**  **例如：CyberCash** | **处理** | 90年代中期，有一种与“中介体系结构”竞争的方法，称为“SET体系结构”。  CyberCash公司实现了SET，CyberCash在2001年破产，知识产权被Verisign收购，Verisign又转手卖给了PayPal，PayPal今天还活着。  SET的购物流程：   * 用户操作：   + 浏览器把你的“交易细节视图”传给本机的“购物应用程序”   + 购物应用程序再加上你的信用卡信息，对这些信息进行加密（只有中介可以解密）。   + 你把加密数据发送给商家。 * 商家操作：商家把数据转给中介，包括商家自己的交易细节视图。 * 中介操作：中介解密你的数据，只有当你和商家的视图匹配时，才批准这个交易。 |
|  | **优点** | SET避免了客户向商家提供信用卡信息。  也不需要用户向中介进行注册。 |
|  | **缺点** | 根本的问题是与“证书”有关。证书是一种方法，它将“加密身份（即公钥）”安全地与“实际身份”关联起来。  网站需要从Verisign这类公司（称为“认证机构”）获取证书，目的是在你的浏览器判断它是安全的（通常由一个锁图标指示）。  CyberCash和SET认为安全性重于可使用性，所以他们要求，不仅processors和商家要获得证书，所有用户也要获得证书。  获得证书就像纳税一样繁琐，所以这个系统是一场灾难。在过去几十年里，主流用户都排斥任何需要最终用户证书的系统，这些提案现在已被沦为了学术论文。 |
| **4 Web联盟** | **处理** | 90年代中期，当SET被标准化时，Web联盟也想对金融支付进行标准化。  他们想通过扩展HTTP协议来实现，这样用户在做交易时就不需要额外的软件，只需要用浏览器。实际上，他们有一个更通用的提案“如何扩展这个协议”，其中一个用例就是支付。  2015年，Web联盟宣布它想采用另一种方法，这次，比特币将成为标准化的一部分。 |
|  | **优点** | 用户在做交易时就不需要额外的软件，只需要用浏览器。 |
|  | **缺点** | 从未实现过，整个扩展框架从未部署在任何浏览器中。 |

## 0.3从信用到（加密）现金

**From Credit to (Crypto) Cash**

Now let’s turn to cash. We compared cash and credit earlier, and noted that a cash system needs to be “bootstrapped,” but the benefit is that it avoids the possibility of a buyer defaulting on her debt. Cash offers two additional advantages. The first is better anonymity. Since your credit card is issued in your name, the bank can track all your spending. But when you pay in cash, the bank doesn’t come into the picture, and the other party doesn’t need to know who you are. Second, cash can enable offline transactions where there’s no need to phone home to a third party in order to get the transaction approved. Maybe later, they go to a third party like a bank to deposit the cash, but that’s much less of a hassle.

我们现在来看看“现金”。

前面比较了“现金”和“信用”，现金系统需要“启动”，但好处是避免了买家拖欠债务的可能性。

现金还有另外两个优势。

* 更好的“匿名性”。

由于信用卡是以你的名义发行的，银行可以跟踪你的所有支出。

但当你用现金支付时，银行就不会出现，收款人也不需要知道你是谁。

* 现金可以进行“离线交易”，不需要打电话给第三方以获准交易。

也许以后，他们会去第三方（例如银行）存钱，但这并不是什么麻烦事。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **信用** | **现金** | **比特币** |
| **衡量价值** | 在现代系统中，信用也用现金来衡量 | 现金能让我们准确衡量物品的价值。 |  |
| **启动运转** | 不需要启动运转 | 需要“初始分配一些现金”来启动这个系统的运转，没有“现金”就没法做交易。 |  |
| **债务风险** | 债权人要冒一些风险，因为债务人可能不偿还债务。 | 避免了买家拖欠债务的可能性 |  |
| **匿名性** | 由于信用卡是以你的名义发行的，银行可以跟踪你的所有支出。 | 更好的“匿名性”。当你用现金支付时，银行就不会出现，收款人也不需要知道你是谁。 | 达不到“现金”的匿名程度  （第6章） |
| **离线交易** |  | 现金可以进行“离线交易”，不需要打电话给第三方以获准交易。 | 也不是完全“离线工作”  （第3章） |

Bitcoin doesn’t quite offer these two properties, but comes close enough to be useful.

比特币并不完全提供这两个属性（匿名性和离线交易），但很接近，所以也很有用。

Bitcoin is not anonymous to the same level as cash is. You don’t need to use your real identity to pay in Bitcoin, but it’s possible that your transactions can be tied together based on the public ledger of transactions with clever algorithms, and then further linked to your identity if you’re not careful. We’ll get into the messy but fascinating details behind Bitcoin anonymity in Chapter 6.

“比特币”达不到“现金”的匿名程度。

支付比特币时不需要用你的真实身份，但是，根据公共交易账本，有可能用巧妙的算法把你的交易联系起来，如果你不小心的话，就可能进一步关联到你的身份。

我们在第6章讨论比特币的“匿名性”。

Bitcoin doesn’t work in a fully offline way either. The good news is it doesn’t require a central server, instead relying on a peer-to-peer network which is resilient in the way that the Internet itself is. In Chapter 3 we’ll look at tricks like “green addresses” and micropayments which allow us to do offline payments in certain situations or under certain assumptions.

比特币也不是完全“离线工作”。

好消息是它不需要中央服务器，但依赖一个P2P网络，这个P2P网络像互联网一样具有弹性。

在第3章中，我们将看到一些技巧，例如“绿色地址”和“小额支付”，它们允许在某些情况或假设下进行“离线支付”。

The earliest ideas of applying cryptography to cash came from David Chaum in 1983. Let’s understand this through a physical analogy. Let’s say I start giving out pieces of paper that say: “The bearer of this note may redeem it for one dollar by presenting it to me” with my signature attached. If people trust that I’ll keep my promise and consider my signature unforgeable, they can pass around these pieces of paper just like banknotes. In fact, banknotes themselves got their start as promissory notes issued by commercial banks. It’s only in fairly recent history that governments stepped in to centralize the money supply and legally require banks to redeem notes.

1983年，David Chaum最早提出了把“密码学”应用于“现金”的想法。

我们通过类比来理解这个想法。

假设，我分发了一些纸条，上面写着：持有这张纸的人可以到我这里兑换一美元，并且有我的签名。

如果人们相信我会遵守诺言（我写的那句话），并且认为我的签名不可伪造（我写的签名），他们就可以像钞票一样使用这些纸条。

实际上，钞票（banknotes）本身是源于商业银行发行的期票（promissory notes）。

只是在近代，政府开始集中供应货币，并在法律上要求银行兑换钞票。

I can do the same thing electronically with digital signatures, but that runs into the annoying “double spending” problem — if you receive a piece of data representing a unit of virtual cash, you can make two (or more) copies of it and pass it on to different people. To stick with our analogy, let’s stretch it a little bit and assume that people can make perfect copies and we have no way to tell copies from the original. Can we solve double spending in this world?

我可以用“数字签名”做同样的事情，但会遇到“双重支付”问题：如果你收到了一个数据，它表示一个“虚拟现金”，你可以拷贝两份，并转给不同的人。

为了进行类比，我们假设有人可以制作完美的副本，我们没有办法区分原本和副本。

我们怎么解决“双重支付”问题呢？

Here’s a possible solution: I put unique serial numbers into each note I give out. When you receive such a note from someone, you check my signature, but you also call me on the phone to ask if a note with that serial number has already been spent. Hopefully I’ll say no, in which case you accept the note. I’ll record the serial number as spent in my ledger, and if you try to spend that note, it won’t work because the recipient will call me and I’ll tell them the note has already been spent. What you’ll need to do instead is to periodically bring me all the notes you’ve received, and I’ll issue you the same number of ​*new*​ notes with fresh serial numbers.

有一个可能的办法是：我在每个票据上写上一个唯一序列号。

当你从某人那里收到一张票据（序列号A）时，你先确认是我的签名，还要给我打电话，问我票据A是否已被花费。

如果我说还没被花费，你就可以接受票据A。我会在账本中把票据A标记为已被花费。

如果你要花费票据A，就不行了，因为收款人会给我打电话，我会告诉他：票据A已被花费。

你需要做的是：定期把收到的所有票据给我，我会给你等值的新票据，它们有新的序列号。

ddk问题：

* 如果多个人拿着相同序列号的票据来兑换怎么办？是否要记下票据的所有人是谁？
* 例如：X将票据复制了一份，现在他有A1和A2，他把A1支付给Y，然后立刻拿着A2去发行人那里兑换，怎么办？难道要跟踪谁持有A吗？这样不就失去了匿名性。
* 根据下一段的描述，这个方案没有匿名性。发行人要记录票据的序列号和持有者的身份。

This works. It’s cumbersome in real life, but straightforward digitally provided I’ve set up a server to do the signing and record-keeping of serial numbers. The only problem is that this isn’t really cash any more, because it’s not anonymous — when I issue a note to you I can record the serial number along with your identity, and I can do the same when someone else later redeems it. That means I can keep track of all the places where you’re spending your money.

这是可行的。它在现实生活中很繁琐，但在网络上却很简单，我设置一个服务器来进行签名和跟踪记录序列号。

唯一的问题是，这不是真正的现金，因为它不是匿名的。

当我给你一张票据时，我可以记录序列号和你的身份，以后当其他人要兑换时，我也可以这样记录序列号和身份。这意味着，我可以跟踪你的钱花到哪儿了。

This is where Chaum’s innovation comes in. He figured out to both keep the system anonymous and prevent double-spending by inventing the digital equivalent of the following procedure: when I issue a new note to you, ​ *you*​pick the serial number. You write it down on the piece of paper, but cover it so that I can’t see it. Then I’ll sign it, still unable to see the serial number. This is called a “blind signature” in cryptography. It’ll be in your interest to pick a long, random serial number to ensure that it will most likely be unique. I don’t have to worry that you’ll pick a serial number that’s already been picked — you can only shoot yourself in the foot by doing so and end up with a note that can’t be spent.

这就是David Chaum做出创新的地方。

他想出了一个办法能够“使系统匿名”，并“防止双花”。

方法是使用下列程序发行“数字现金”：

* 当我给你一个新票据时，你选择一个序列号，把它写在票据上，不要让我看见。

你要选一个长的随机序列号，以保证它尽可能是唯一的。

* 然后，我进行签名，仍然不要让我看到序列号。这在密码学中称为“盲签名”。

我不担心你会选已在使用的序列号，因为如果你那样做，最终你会无法花费这个票据。

This was the first serious digital cash proposal. It works, but it still requires a server run by a central authority, such as a bank, and for everyone to trust that entity. Moreover, every transaction needs the participation of this server to go through. If the server goes down temporarily, payments grind to a halt. A few years later, in 1988, Chaum in collaboration with two other cryptographers Fiat and Naor proposed ​*offline* ​electronic cash. At first sight this might seem to be impossible: if you try to spend the same digital note or coin at two different shops, how can they possibly stop this unless they’re both connected to the same payment network or central entity?

这是第一个重要的“数字现金”提议。

它可以工作，但仍然需要由中央机构（例如银行）运营一个服务器，并让每个人都信任这个结构。

此外，服务器要参与每个交易。如果服务器出问题，支付就会停止。

1988年，Chaum与另外两位密码学家（Fiat和Naor）一起提出了“离线电子现金”。

乍一看，这似乎是不可能的：如果你试图在两个不同的商店花费相同的数字现金，他们怎么阻止双花呢？除非它们都连接到同一支付网络或中央结实体。

The clever idea is to stop worrying about preventing double-spending and focus on detecting it, after the fact, when the merchant re-connects to the bank server. After all, this is why you’re able to use your credit card on an airplane even if there is no network connection up in the skies. The transaction processing happens later when the airline is able to re-connect to the network. If your card is denied, you’ll owe the airline (or your bank) money. If you think about it, quite a bit of traditional finance is based on the idea of detecting an error or loss, followed by attempting to recover the money or punish the perpetrator. If you write someone a personal check, they have no guarantee that the money is actually in your account, but they can come after you if the check bounces. Conceivably, if an offline electronic cash system were widely adopted, the legal system would come to recognize double spending as a crime.

这个聪明的想法是：停止“为防止双花而担忧”，而是专注于“在事后检测双花”，在商家重新连接银行服务器时检测双花。

这就是为什么：虽然天上没有网络连接，但你可以在飞机上使用信用卡。

当航空公司重新连接到网络时，就会处理交易。如果你的卡被拒绝，你就会欠航空公司（或银行）的钱。

想一想，传统金融的很大一部分是基于检测错误或损失的想法，然后是试图收回钱或惩罚作恶者。

如果你给某人开了个人支票，他们不能保证钱在你的账户里，但他们可以在支票出问题时找你。

可想而知，如果离线电子现金系统被广泛采用，法律系统会把双花视为一种犯罪。

ddk说明：双花只有本人才能进行，因为钱在本人手里。所以，如果发生了双花，那肯定是钱的所有者做了双花。

Chaum, Fiat, and Naor’s idea for detecting double spending was an intricate cryptographic dance. At a high level, what it achieved was this: every digital coin issued to you encodes your identity, but in such a way that no one except you, not even the bank, can decode it. Every time you spend your coin, the recipient will require you to decode a random subset of the encoding, and they’ll keep a record of this. This decoding isn’t enough to allow them to determine your identity. But if you ever double spend a coin, eventually both recipients will go to the bank to redeem their notes, and when they do this, the bank can put the two pieces of information together to decode your identity completely, with an overwhelmingly high probability.

Chaum-Fiat-Naor检测双花的方法是：使用复杂的密码学。

在高层次上，它实现的是：

* 发给你的每个数字货币都编码了你的身份，但只有你能解码它，即使银行也无法解码。
* 当你花钱时，收款人要求你解码那个编码的一部分，然后记下来。这个解码不足以确定你的身份。
* 如果你做了双花，当两个收款人到银行兑换票据时，银行会把这两个信息放在一起，从而确定你的身份。

You might wonder if someone can frame you as a double spender in this system. Say you spend a coin with me, and then I turned around and tried to double-spend it (without redeeming it with the bank and getting a new coin with my identity encoded). This won’t work — the new recipient will ask me to decode a random subset, and this will almost certainly not be the same as the subset you decoded for me, so I won’t be able to comply with their decoding request.

你可能会怀疑：是否会有人用这个系统故意陷害你。

假设你给我一个货币，然后我试图双花它（没有到银行进行兑换，所以没有有我的身份编码的新货币。）

这是行不通的，新的收款人会让我解码一个随机子集，这几乎肯定不会与你为我解码的子集一样，所以我没法解码。

Over the years, many cryptographers have looked at this construction and improved it in various ways. In the Chaum-Fiat-Naor scheme, if a coin is worth $100, and you wanted to buy something that cost only $75, say, there’s no way to split that coin into $75 and a $25. All you could do is go back to the bank, cash in the $100 coin, and ask for a $75 coin and a $25 coin. But a paper by Okamoto and Ohta uses “Merkle trees” to create a system that does allow you to subdivide your coins. Merkle trees would show up in Bitcoin as well, and we’ll meet them in Chapter 1. The Chaum-Fiat-Naor scheme also leaves a lot of room for improvements in efficiency. In particular, the application of something called zero-knowledge proofs to this scheme (most notably by Brands; and Camenisch, Hohenberger, and Lysyanskaya) was very fruitful—zero-knowledge proofs have also been applied to Bitcoin as we will see in Chapter 6.

多年来，许多密码学家已经研究了这种结构，并以用各种方法来改进它。

在Chaum-Fiat-Naor方案中，如果一个货币值100美元，而你想买的东西只有75美元，没有办法把这个货币分割成75美元和25美元。你必须到银行把这个货币换成现金，然后再要一个75美元的货币和一个25美元的货币。

但Okamoto和Ohta写了一篇论文，使用“Merkle树”来创建一个系统，这个系统可以让你分割货币。

比特币也用到了Merkle树，我们将在第1章介绍。

Chan-Fiat-Naor方案还在效率上留下了很大的改进空间。特别是，对这个机制应用“零知识证明”成果显著。比特币也用到了零知识证明，我们将在第6章介绍。

But back to Chaum: he took his ideas and commercialized them. He formed a company in 1989 called DigiCash, probably the earliest company that tried to solve the problem of online payments. They had about a five-year head start on other companies like FirstVirtual and CyberCash that we just discussed. The actual cash in Digicash’s system was called Ecash and they had another system called cyberbucks. There were banks that actually implemented it — a few in the US and at least one in Finland. This was in the 1990s, long before Bitcoin, which might come as surprise to some Bitcoin enthusiasts who view banks as tech-phobic, anti-innovative behemoths.

回到David Chaum：他把他的想法进行了商业化。

他在1989成立了DigiCash公司，这可能是最早尝试解决网上支付问题的公司。

他们比其它公司（例如FirstVirtual和CyberCash）早了大约5年。

DigiCash系统中的现金被称为Ecash，他们还有另外一个称为cyberbucks的系统。

实际上有一些银行实施了它，在美国有一些银行，在芬兰至少有一家银行。

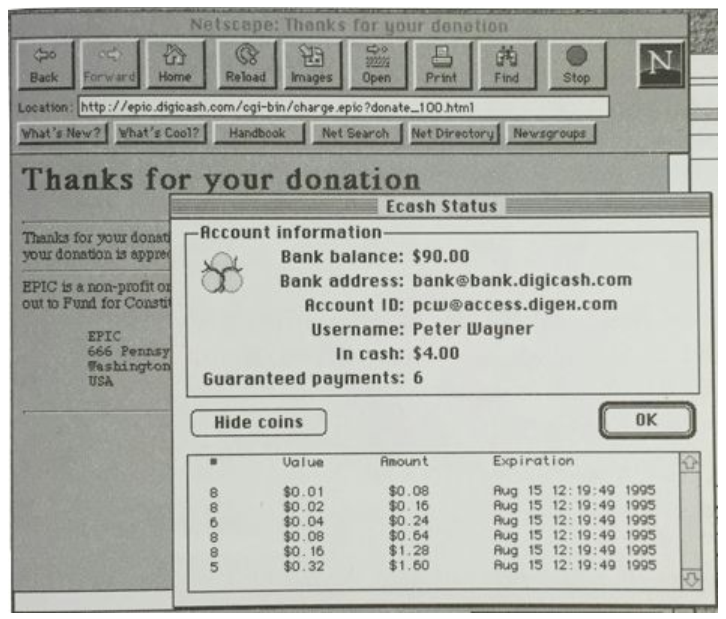
这是在1990年代，在比特币之前很久，这可能让一些比特币狂热者感到惊讶，这些比特币狂热者把银行视为惧怕技术、抵制创新的庞然大物。

Ecash is based on Chaum’s protocols. Clients are anonymous, so banks can’t trace how they’re spending their money. But merchants in ecash aren’t anonymous. They have to return coins as soon as they receive them, so the bank knows how much they’re making, at what times, and so on.

Ecash是基于Chaum的协议。

客户是匿名的，所以银行无法追踪客户的花钱方式。

但ecash中的商家不是匿名的。他们在收到货币时要退回它，所以银行知道他们赚了多少钱、何时等。



**Figure 2: Screenshot of DigiCash**

Figure 2 shows a screenshot from the software. As you can see, it shows you your balance as well as all the coins that you have that have been issued to you from the bank. Since there’s no way to split your coins, the bank issues you a whole set of coins in denominations of a cent, two cents, four cents, and so on — powers of two. That way, you (or your software, on your behalf) can always select a set of coins to pay for the exact amount of a transaction.

图2显示了DigiCash软件的截图。

正如你看到的，它显示了你的余额，以及银行发给你的所有货币。

因为没有办法分割货币，银行给你一套货币，面额为1分、2分、4分等，它们都是2的幂。

这样，你可以选择一组货币来支付交易的金额。

When you want to make a transaction, say, as in this example, you want to make a donation to the non-profit privacy group EPIC, you’d click on a donation link that takes you to the Digicash website. That would then open a reverse web connection back to your computer. That means your computer had to have the ability to accept incoming connections and act as a server. You’d have to have your own IP address and your ISP would have to allow incoming connections. If the connection was successful, then the ecash software would launch on your computer and you’d be able to approve the transaction and send the money.

当你想做交易时，例如，你想给EPIC捐款，你点击一个捐赠链接，进入DigiCase网站。

然后打开一个反向Web连接，回到你的计算机。这意味着，你的计算机必须有能力作为服务器接受进入的连接。你必须有自己的IP地址，并且你的ISP还要允许进入连接。

如果连接成功，那么ecash软件将在你的计算机上启动，你就能够批准交易和转钱了。

Chaum had several patents on Digicash technology, in particular, the blind-signature scheme that it used. This was controversial, and it stopped other people from developing ecash systems that used the same protocol. But a bunch of cryptographers who hung out on what was called the cypherpunks mailing list wanted an alternative. Cyperpunks was the predecessor to the mailing list where Satoshi Nakamoto would later announce Bitcoin to the world, and this is no coincidence. We’ll talk about the cypherpunk movement and the roots of Bitcoin in Chapter 7.

Chaum在DigiCase技术上有几项专利，特别是它使用的“盲签名”机制。

这是有争议的，它阻止了其他人开发使用相同协议的电子现金系统。

但是一群密码学者（他们在Cypherpunks邮件列表中）想要另一种选择。

Cyperpunks是中本聪所在的邮件列表的前身，中本聪后来向世界公布了比特币，这不是巧合。

在第7章，我们将讨论Cypherpunks运动和比特币的根源。

The cypherpunk cryptographers implemented a version of of ecash called MagicMoney. It did violate the patents, but was billed as being only for experimental use. It was a fun piece of software to play with. The interface was all text-based. You could send transactions by email. You would just copy and paste the transactions into your email and send it to another user. Hopefully, you’d use end-to-end email encryption software such as PGP to protect the transaction in transit.

cypherpunk密码学者实现了一个电子现金版本，称为MaigicMoney。

它确实侵犯了专利，但他们宣称只用于实验目的。

这是一个有趣的软件。界面都是基于文本的。

你可以通过email发送交易，你只需复制和粘贴交易到你的email，然后发送给另一个用户。

希望你是使用端到端加密软件（例如PGP）来保护传输的交易。

Then there’s a proposal called Lucre by Ben Laurie with contributions from many other people. Lucre tries to replace the blind-signature scheme in ecash with a non-patent-encumbered alternative, with the rest of the system largely the same.

然后，Ben Laurie提出了一个提案Lucre。

Lucre试图用一个非专利方法替代电子现金中的“盲签名”机制，而其它部分都相同。

Yet another proposal, by Ian Goldberg, tries to fix the problem of not being able to split your coins to make change. His idea was that the merchant could send you coins back if they had some coins, so that you might overpay for the item if you didn’t have exact change, and then you’d get some coins back. But notice that this introduces an anonymity problem. As we saw earlier, in ecash, senders are anonymous but merchants aren’t. When the merchant sends cash back, technically, they’re the sender, so they’re anonymous. But you, as someone who has to return this cash to the bank, aren’t anonymous. There’s no way to design this system without breaking the anonymity of users trying to buy goods. So Goldberg came up with a proposal where there were different types of coins that would allow these transactions to occur, allow you to get change back, and still preserve your anonymity.

Ian Goldberg提出的另一个提议，试图解决不能分钱找零的问题。

他的想法是，如果商家有一些货币的话，他们可以把钱返回给你，相当于找零。

但注意，这引入了匿名性问题。正如我们之前看到的，在ecash中，付款人是匿名的，但商家不是。

当商家返回现金时，从技术上讲，他们是付款人，所以他们是匿名的。

但你必须把这些现金返回给银行，所以你就不是匿名的。

没有办法在不打破用户匿名性的情况下设计这个系统。

所以，Goldberg提出了一个建议：有不同类型的货币（允许这些交易发生），允许你得到找零，并且仍然保持你的匿名性。

Now, why did DigiCash fail? The main problem with DigiCash was that it was hard to persuade the banks and the merchants to adopt it. Since there weren’t many merchants that accepted ecash, users didn’t want it either. Worse, it didn’t support user-to-user transactions, or at least not very well. It was really centered on the user-to-merchant transaction. So if merchants weren’t on board, there was no other way to bootstrap interest in the system. So at the end of the day, DigiCash lost and the credit card companies won.

**为什么DigiCash失败了？**

DigiCash的主要问题是：很难说服银行和商家采用它。

由于没有很多商家接受ecash，用户也不想用它。

更糟糕的是，它不支持“用户-用户交易”，或者至少不是很好。

它实际上集中在“用户-商家交易”。因此，如果商家不同意，就没有方法来激发对系统的兴趣。

所以，DigiCash最终失败了，而信用卡公司赢了。

As a side note, Bitcoin allows user-to-merchant and user-to-user transactions. In fact, the protocol doesn’t have a notion of merchant that’s separate from the notion of user. The support for user-to-user transactions probably contributed to Bitcoin’s success. There was something to do with your bitcoins right from the beginning: send it to other users, while the community tried to drum up support for Bitcoin and get merchants to accept it.

顺便说一下，比特币允许“用户-商家交易”和“用户-用户交易”。

事实上，这个协议并没有区分用户和商家的概念。

对“用户-用户交易”的支持可能促成了比特币的成功。

从一开始就有点关系：把比特币发给其他用户，而社区鼓动对比特币的支持，让商家接受它。

In the later years of the company, DigiCash also experimented with tamper-resistant hardware to try to ​ *prevent*​double-spending rather than just detecting it. In this system, you’d get a small hardware device that was usually called a wallet, or some sort of card. The device would keep track of your balance, which would decrease when you spent money and increase if you loaded the card with more money. The point of the device is that there should be no way to physically or digitally go in and tamper with its counter. So if the counter hits zero, then the card stops being able to spend money until it’s re-loaded.

在公司的后期，DigiCash也尝试了防篡改硬件，来防止双重支付，而不仅仅是检测它。

在这个系统中，你会得到一个小的硬件设备（称为钱包），或者某种卡片。

这个设备会记录你的余额，当你花钱的时候，它会减少，如果你把更多的钱充入卡片，它就会增加。

这个设备的要点是，没有办法物理或数字地进入和篡改它的计数器。

因此，如果计数器为零，则卡片就不能再花钱。

There were many other companies that had electronic cash systems based on tamper-resistant hardware. DigiCash later worked with a company called CAFE which was based in Europe. Another company formed around this idea was called Mondex and it was later acquired by Mastercard. Visa also had their own variant called VisaCash.

还有许多其它公司，也有基于防篡改硬件的电子现金系统。

DigiCash后来与CAFE公司合作，它的总部设在欧洲。

围绕这一想法而组建的另一家公司是Mondex，后来被Mastercard收购。

Visa也有自己的变体，称为VisaCash。



**Figure 3: Mondex system, showing user card and wallet.**

**图3：Mondex系统，显示了用户卡片和钱包**

Figure 3 shows the user side of the Mondex system. There’s a smart card and there’s a wallet unit, and you can load either of them with cash. And if you wanted to do user-to-user swap of money, the giver user would first put their card into the wallet and move money off of the card onto the wallet. Then the receiver would stick their card in the wallet then you’d move the money onto the second card. This was a way to exchange digital cash, and it was anonymous.

图3显示了Mondex系统的用户端。

有一张智能卡，有一个钱包，你可以向它们冲入现金。

如果你想进行用户对用户的货币交换，付款人首先将他的卡放入钱包，把钱从卡片移到钱包上。

然后，收款人把他的卡片放在钱包上，把钱移到这种卡片上。

这是一种交换数字现金的方式，它是匿名的。

Mondex trialled their technology in a bunch of communities. One community happened to be a city very close to where I grew up: Guelph, Ontario. You’ve probably already guessed that it didn’t really catch on. A major problem with Mondex cards is that they’re like cash — if you lose them or they get stolen, the money’s gone. Worse, if there’s some sort of malfunction with the card, if the card reader wouldn’t read it, there’s no way to figure out if that card had balance on it or not. In these scenarios, Mondex would typically eat the cost. They’d assume that the card was loaded and reimburse the user for that lost money. Of course, that can cost a company a lot of money.

Mondex在一些社区里试用他们的技术。

你可能已经猜到它并没有真正流行起来。一个主要问题是它就像现金一样，如果丢了或被盗，钱就没了。更糟的是，如果卡片出现了故障，如果读卡器不能读卡，就没有办法知道这张卡是否有余额。

在这些情况下，Mondex一般会支付成本。他们会假设卡片有钱，并赔偿用户的损失。当然，这会让公司花很多钱。

Further, the wallet was slow and clunky. It was much faster to pay with a credit card or with cash. And retailers hated having several payment terminals; they wanted just one for credit cards. All these factors together did Mondex in.

此外，钱包很慢，而且笨拙。用信用卡或现金支付要快得多。

零售商讨厌有许多付款终端，他们只想要一个信用卡终端。

所有这些因素共同影响了Mondex。

However, these cards were smart cards, which means that they have small microcontrollers on them, and that technology has proved successful. In many countries today, including Canada, where I live, every single credit card and every single debit card now has smart card technology in it. It’s used for a different purpose, though. It’s not used to prevent double-spending — the problem doesn’t arise since it’s not a cash-based technology. The bank, rather than your card, keeps track of your balance or available credit. Instead the chip is used for authentication, that is, to prove that you know the PIN that’s associated with your account. But Mondex was using it long before this technology was adopted widely by the banking industry.

但是，这些卡片是智能卡，这意味着，它们有小型微控制器，这个技术已证明是成功的。

在许多国家，包括加拿大，每张信用卡和每张借记卡现在都有智能卡技术。

然而，它被用于不同的目的。不是用来防止双花，因为它不是基于现金的技术，所以不会出现这个问题。

银行（不是卡片）跟踪余额或可用额度。芯片被用于认证，即，证明你知道与你的账户关联的PIN。

但在银行业广泛采用这项技术之前很久，Mondex就在使用它。

**【总结】**

1、**银行期票：签名**

如何解决“双重支付”问题：加个序列号，跟踪每个期票

**2、DigiCash公司的Ecash**

1983年，David Chaum最早提出了把“密码学”应用于“现金”的想法。

采用的方法：盲签名

仍然需要由中央机构（例如银行）运营一个服务器，并让每个人都信任这个结构。

服务器要参与每个交易。如果服务器出问题，支付就会停止。

1988年，Chaum与Fiat和Naor一起提出了“离线电子现金”。

他们的想法是：停止“为防止双花而担忧”，而是专注于“在事后检测双花”，在商家重新连接银行服务器时检测双花。

Chaum-Fiat-Naor检测双花的方法是：使用密码学。

在高层次上，它实现的是：

* 发给你的每个数字货币都编码了你的身份，但只有你能解码它，即使银行也无法解码。
* 当你花钱时，收款人要求你解码那个编码的一部分，然后记下来。这个解码不足以确定你的身份。
* 如果你做了双花，当两个收款人到银行兑换票据时，银行会把这两个信息放在一起，从而确定你的身份。

1989年，Chaum在成立了DigiCash公司，这可能是最早尝试解决网上支付问题的公司。

DigiCash系统中的现金被称为Ecash

* Ecash是基于Chaum的协议。
* 客户是匿名的，所以银行无法追踪客户的花钱方式。但ecash中的商家不是匿名的。
* Ecash不能分割货币

交易方法：

当你想做交易时，例如，你想给EPIC捐款，你点击一个捐赠链接，进入DigiCase网站。

然后打开一个反向Web连接，回到你的计算机。这意味着，你的计算机必须有能力作为服务器接受进入的连接。你必须有自己的IP地址，并且你的ISP还要允许进入连接。

如果连接成功，那么ecash软件将在你的计算机上启动，你就能够批准交易和转钱了。

**为什么DigiCash失败了？**

DigiCash的主要问题是：很难说服银行和商家采用它。

由于没有很多商家接受ecash，用户也不想用它。

更糟糕的是，它不支持“用户-用户交易”，或者至少不是很好。

它实际上集中在“用户-商家交易”。因此，如果商家不同意，就没有方法来激发对系统的兴趣。

所以，DigiCash最终失败了，而信用卡公司赢了。

在DigiCash的后期，也尝试了防篡改硬件，来防止双重支付，而不仅仅是检测它。

围绕这一想法而组建的另一家公司是Mondex，但并没有真正流行起来。

* 一个主要问题是它就像现金一样，如果丢了或被盗，钱就没了。

更糟的是，如果卡片出现了故障，如果读卡器不能读卡，就没有办法知道这张卡是否有余额。

在这些情况下，Mondex一般会支付成本。他们会假设卡片有钱，并赔偿用户的损失。当然，这会让公司花很多钱。

* 钱包很慢，而且笨拙。用信用卡或现金支付要快得多。
* 零售商讨厌有许多付款终端，他们只想要一个信用卡终端。

所有这些因素共同影响了Mondex。

但是，这些卡片所使用的智能卡技术已证明是成功的。

**3、对Chaum-Fiat-Naor方案的改进**

多年来，许多密码学家已经研究了这种结构，并以用各种方法来改进它。

* 在Chaum-Fiat-Naor方案中，不能分割货币。

Okamoto和Ohta写了一篇论文，使用“Merkle树”来创建一个系统，这个系统可以让你分割货币。

比特币也用到了Merkle树，我们将在第1章介绍。

* Chan-Fiat-Naor方案还在效率上留下了很大的改进空间。

特别是，对这个机制应用“零知识证明”成果显著。

比特币也用到了零知识证明，我们将在第6章介绍。

**4、新的尝试**

Chaum在DigiCash技术上有几项专利，特别是它使用的“盲签名”机制。

但是一群密码学者（他们在Cypherpunks邮件列表中）想要另一种选择。

Ben Laurie提出了一个提案Lucre。

Lucre试图用一个非专利方法替代电子现金中的“盲签名”机制，而其它部分都相同。

Ian Goldberg提出的另一个提议，试图解决不能分钱找零的问题。

为了解决Chaum-Fiat-Naor方案的商家匿名性问题，Goldberg提出了一个建议：有不同类型的货币（允许这些交易发生），允许你得到找零，并且仍然保持你的匿名性。

|  |  |  |
| --- | --- | --- |
|  | **DigiCash** | **改进** |
| 1 | “盲签名”是专利 | Ben Laurie提出了一个提案Lucre。  Lucre试图用一个非专利方法替代电子现金中的“盲签名”机制，而其它部分都相同。 |
| 2 | 需要中央机构 | 比特币是去中性化的 |
| 3 | 不能分割货币 | Okamoto和Ohta写了一篇论文，使用“Merkle树”来创建一个系统，这个系统可以让你分割货币。  比特币也用到了Merkle树（第1章） |
| 4 | 商家匿名性问题 | Ian Goldberg提出的另一个提议，试图解决不能分钱找零的问题。  为了解决Chaum-Fiat-Naor方案的商家匿名性问题，Goldberg提出了一个建议：有不同类型的货币（允许这5些交易发生），允许你得到找零，并且仍然保持你的匿名性。  在比特币中，不区分用户和商家，都是匿名的。 |
| 5 | 效率 | 特别是，对这个机制应用“零知识证明”成果显著。  比特币也用到了零知识证明（第6章） |
| 6 | 很难说服银行和商家采用它 | 比特币的策略：把比特币发给其他用户，而社区鼓动对比特币的支持，让商家接受它。 |
| 7 | 防篡改硬件 | ddk：比特币在硬件钱包中用到了这个技术？ |
| 8 | 智能卡技术 | 现在，每张信用卡和借记卡现在都有智能卡技术。  然而，它被用于不同的目的。不是用来防止双花，因为它不是基于现金的技术，所以不会出现这个问题。  银行（不是卡片）跟踪余额或可用额度。芯片被用于认证，即，证明你知道与你的账户关联的PIN。 |

## 0.4凭空发行的货币

**Minting Money out of Thin Air**

In the DigiCash system, if you have a digital cash object that’s worth $100, what makes it actually worth $100? The answer is simple: in order to obtain ecash worth $100, you’d have to take $100 out of your bank account and give it to the bank that was issuing you the ecash. But there were a bunch of different proposals for how to do this and different companies did it differently. One far-fetched possibility: what if the government of a particular country actually authorized services to mint digital money, creating new cash out of thin air? That was the idea behind NetCash, although it never got beyond the proposal stage. A different system, used by e-Gold, was to put a pile of gold in a vault and to issue digital cash only up to the value of the gold. Another company called Digigold wasn’t fully backed by gold, but had partial reserves.

在DigiCash系统中，如果你有一个价值100美元的数字现金，是什么使它真得值100美元呢？

答案很简单：为了获得价值100美元的ecash，你必须拿出100美元，把它交给发行ecash的银行。

但是有一些不同的提案来实现这个，不同的公司也有不同的做法。

* 一个天方夜谭的想法是：如果政府批准铸造数字货币，凭空创建新的现金，会怎么样？

这是NetCach背后的想法，尽管它只是一个建议，从未实现过。

* e-Gold使用了另一个系统：把一堆黄金放在金库中，然后发行与黄金等值的数字现金。

Digigold公司并不完全由黄金支持，但它有部分储备。

All of these ideas ultimately peg the value of digital cash to the dollar or a commodity. If the dollar’s value goes up or down, the value of your digital money holdings will change along with it. A radically different possibility is to allow digital money to be it own currency, issued and valued independently of any other currency.

所有这些想法最终是把“数字现金”的价值与美元或商品锚定。

如果美元的升值或贬值，则数字货币也会随之改变。

一种完全不同的想法是：让数字货币本身成为货币，它的发行和价值独立于任何其它货币。

To create a free-floating digital currency that is likely to acquire real value, you need to have something that’s scarce by design. In fact, scarcity is also the reason why gold or diamonds have been used as a backing for money. In the digital realm, one way to achieve scarcity is to design the system so that minting money requires solving a computational problem (or “puzzle”) that takes a while to crack. This is what happens in Bitcoin “mining”, which we’ll look at in Chapter 5.

为了创造一种能够获得真正价值的自由浮动的数字货币，你需要设计一些稀缺的东西。

实际上，“稀缺性”也是黄金或钻石作为货币支持的原因。

在数字领域中，实现稀缺性的一种方法是：为了得到货币，必须解决一个计算难题，它要花一些时间。

这就是比特币中的“挖矿”，我们将在第5章中介绍

The basic idea — that solutions to computational puzzles could be digital objects that have some value — is pretty old. It was first proposed by cryptographers Dwork and Naor as a potential solution to email spam back in 1992. What if, every time you sent an email, your computer would have to solve one of these puzzles that would take a few seconds to solve? To enforce this requirement, the recipient’s email program would simply ignore your email you didn’t attach the solution to the computational puzzle. For the average user, it wouldn’t be that much of a barrier to sending emails because you’re not sending emails very frequently. But if you’re a spammer, you’re trying to send out thousands or millions of emails all at once, and solving those computational puzzles could become prohibitive. A similar idea was later discovered independently by Adam Back in 1997 in a proposal called Hashcash.

基本的想法相当古老：计算难题的解可以是有一定价值的数字对象。

这是密码学着Dwork和Naor在1992年首次提出的，用于解决垃圾邮件的一个可能方案。

如果你每次发一封电子邮件，你的计算机就要解决一个难题，这需要花几秒钟，会怎样呢？

如果你在电子邮件中没有附上这个难题的解，收件人的电子邮件程序就会忽略你的电子邮件。

对于一般用户来说，发送电子邮件不是很大的障碍，因为你不经常发送电子邮件。

但对于垃圾邮件发送者，他会同时发送成千上万封电子邮件，解决这些计算难题就会是个麻烦。

一个类似的想法后来在1997被Adam Back独立地发现，用在一个名为Hashcash的建议中。

These computational puzzles need to have some specific properties to be a useful spam deterrent. First, it should be impossible for a spammer to solve one puzzle and attach the solution to every email he sends. To ensure this, the puzzle should be specific to the email: it should depend on the sender and receiver, the contents of the email, and the approximate time at which it’s sent. Second, the receiver should be able to easily check the puzzle solution without having to repeat the process of solving the puzzle. Third, each puzzle should be totally independent of the others, in the sense that solving one puzzle does not decrease the amount of time it takes to solve any other puzzle. Finally, since hardware improves with time and solving any given computational puzzle gets faster and cheaper, recipients should be able to adjust the difficulty of the puzzle solutions that they will accept. These properties can be achieved by using cryptographic hash functions to design the puzzles, and we’ll study this in Chapter 1.

这些计算难题需要有一些特定的属性，才能有效地威慑垃圾邮件。

1. 垃圾邮件发送者不能只解决一个难题，并把这个解附在每封电子邮件中。

为了确保这一点，难题应该针对每个电子邮件：它应该取决于发送者和接收者、内容、发送时间。

1. 接收者应该能够很容易检查难题的解，而不必重复解决难题的过程。
2. 每个难题应该完全独立于其它难题，这样解决一个难题不会减少解决其它难题所需的时间。
3. 由于硬件的升级，导致解决难题会变得更快、更容易，收件人应该能够调整难题的难度。

通过使用密码哈希函数来设计难题，能够获得这些属性。我们将在第1章中介绍。

Bitcoin uses essentially the same computational puzzle as Hashcash, but with some minor improvements. Bitcoin does a lot more than Hashcash does, though — after all, it takes a whole book to explain Bitcoin! I only mention this because Hashcash inventor Adam Back has said, “Bitcoin is Hashcash extended with inflation control.” I think that’s overreaching a bit. It’s sort of like saying, “a Tesla is just a battery on wheels.”

比特币基本上使用与Hashcash相同的计算难题，但有一些小的改进。

比特币比Hashcash做的多了很多，但是，因为本书全书都是解释比特币，所以这里不再多说！

我只提到这点是因为Hashcash的发明者Adam Back说：“比特币是扩展了Hashcash，控制了通货膨胀。”

我认为这有点儿过头了，这就像说：“特斯拉只是车轮上的电池。”

As with any good idea in cryptography, there are many variants of computational puzzles that aim to achieve slightly different properties. One proposal comes from Rivest and Shamir, the R and the S in the RSA cryptosystem. Observe that in Hashcash, your cost to solve a number of puzzles is simply the sum of the individual costs, by design. But this is different from the cost structure for a government to mint money. If you think about how anti-counterfeiting technology in a paper currency, there’s a huge initial cost to acquire all the equipment, create the security features, and so on. But once they’ve done all that, their costs go down, and it doesn’t matter much if they print one bill or a hundred bills. In other words, minting paper money has a huge fixed cost but low marginal cost.

正如在密码学中的任何好想法一样，也有许多计算难题的变体，它们是为了实现稍微不同的属性。

一个建议来自Rivest和Shamir（RSA公司）。

可以看出，在Hashcash的设计中，解决许多难题的成本仅仅是单个成本的总和。

但这与政府发行货币的成本结构不同。纸币有防伪技术，需要巨大的初始成本，创造出安全特性，等等。但是，一旦实现了这些特性，造币成本就会下降，发行一张纸币和一百张纸币并没有太大区别。

换言之，发行纸币有很大的固定成本，但边际成本较低。

Rivest and Shamir wanted to design computational puzzles that would mimic these properties, so that minting the first coin is massively computationally challenging, but minting subsequent coins is a lot cheaper. Their proposal also utilized hash functions, but in a different way. We won’t get into the details of their solution, but the problem they were trying to solve is interesting at a high level.

Rivest和Shamir想设计这样的计算难题：模仿纸币的属性，使铸造第一个货币很难，但后续铸造能简单得多。

他们的建议也是利用哈希函数，但方式不同。

我们不会深入他们的方案细节，但是，他们试图解决的问题在高层次上是很有趣的。

Why did Hashcash never catch on for its intended purpose of preventing spam? Perhaps spam just wasn’t a big enough problem to solve. For most people spam as a nuisance, but not something that they want to spend their computing cycles on combatting. We have spam filters today that work pretty well at keeping spam out of our inboxes. It’s also possible Hashcash wouldn’t have actually stopped spammers. In particular, most spammers today send their spam using ‘botnets’, large groups of of other people’s computers that they take control of using malware. They might just as well use those computers to harvest Hashcash. That said, the idea of using computational puzzles to limit access to resources is still an idea that’s kicking around. You can see it in some proposals for replacing network protocols, such as MinimaLT.

为什么Hashcash没在实际中用于防止垃圾邮件？

也许垃圾邮件问题还不是一个大到必须解决的问题。对于多数人来说，垃圾邮件是个讨厌的东西，但还不是他们想要花时间来解决的问题。我们现在有垃圾邮件过滤器，它能很好地防止垃圾邮件进入收件箱。

也有可能是Hashcash并没有能够真正阻止垃圾邮件发送者。特别是，现在多数垃圾邮件发送者使用“僵尸网络”来发送垃圾邮件，他们使用恶意软件控制别人的计算机。他们也可以使用这些计算机来收割Hashcash。

也就是说，使用计算难题来限制对资源的访问的想法，仍然是一个非正式的提议。你可以在一些替代网络协议的建议中看到它，例如MelimalT。

**【总结】**

解题：凭空发行货币

* 即使数字货币本身独立的价值
* 方法是：解决难题，哈希函数、挖矿

在DigiCash系统中，如果你有一个价值100美元的数字现金，是什么使它真得值100美元呢？

有一些不同的提案来实现这个，不同的公司也有不同的做法。

所有这些想法最终是把“数字现金”的价值与美元或商品锚定。

一种完全不同的想法是：让数字货币本身成为货币，它的发行和价值独立于任何其它货币。

为了创造一种能够获得真正价值的自由浮动的数字货币，你需要设计一些稀缺的东西。

在数字领域中，实现稀缺性的一种方法是：为了得到货币，必须解决一个计算难题，它要花一些时间。

这就是比特币中的“挖矿”。（第5章）

基本的想法相当古老：计算难题的解可以是有一定价值的数字对象。

这是密码学着Dwork和Naor在1992年首次提出的，用于解决垃圾邮件的一个可能方案。

一个类似的想法后来在1997被Adam Back独立地发现，用在一个名为Hashcash的建议中。

比特币基本上使用与Hashcash相同的计算难题，但有一些小的改进。

这些计算难题需要有一些特定的属性，才能具有价值：

1. 发送者不能只解决一个难题
2. 接收者应该能够很容易检查难题的解，而不必重复解决难题的过程。
3. 每个难题应该完全独立于其它难题，这样解决一个难题不会减少解决其它难题所需的时间。
4. 由于硬件的升级，导致解决难题会变得更快、更容易，收件人应该能够调整难题的难度。

通过使用“密码哈希函数”来设计难题，能够获得这些属性。（第1章）

正如在密码学中的任何好想法一样，也有许多计算难题的变体，它们是为了实现稍微不同的属性。

一个建议来自Rivest和Shamir（RSA公司）。

他们想设计这样的计算难题：模仿纸币的属性，使铸造第一个货币很难，但后续铸造能简单得多。

他们的建议也是利用哈希函数，但方式不同。

## 0.5把所有东西记录在一个账本中

Recording Everything in a Ledger

Another key component of Bitcoin is the block chain: a ledger in which all Bitcoin transactions are securely recorded. The ideas behind the block chain are again quite old, and trace back to a paper by Haber and Stornetta in 1991. Their proposal was a method for secure timestamping of digital documents, rather than an digital money scheme. The goal of timestamping is to give an approximate idea of when a document came into existence. More importantly, timestamping accurately conveys the order of creation of these documents: if one came into existence before the other, the timestamps will reflect that. The security property requires that a document’s timestamp can’t be changed after the fact.

比特币的另一个重要组成部分是“区块链”：它是一个账目，所有比特币交易都被安全地记录在里面。

区块链背后的思想很久之前就有了，可追溯到Haber和Surneta在1991年发表的一篇论文。

他们提议的方法是针对数字文档的“安全时间戳”，而不是针对数字货币方案。

* 时间戳：
  + 时间戳的目的是为了标记一个文档是在何时出现的。
  + 更重要的是，时间戳准确地记录了先后顺序：如果一个文档比另一文档先出现，时间戳能够反映出这个先后顺序。
* 安全属性：
  + 要求：在事实发生之后，文档的时间戳不能被更改。

In Haber and Stornetta’s scheme, there’s a timestamping service to which clients send documents to timestamp. When the server receives a document, it signs the document together with the current time and as well as a link or a pointer to the previous document, and issues a “certificate” with this information. The pointer in question a special type pointer which links to a piece of data instead of a location. That means that if the data in question changes, the pointer automatically become invalid. In Chapter 1 we’ll study how we can create such pointers using hash functions.

在Haber和SturnTa的方案中，有一个“时间戳服务器”：

* 客户端：把文档发给时间戳服务器
* 服务器：
  + 收到一个文档时，它对这些进行签名：文档、当前时间、指向前一个文档的链接或指针。
  + 然后用这个信息发布一个“证书”。

这里所说的指针是一个特殊类型指针，它与一个数据关联起来，而不是与位置关联起来。这意味着，如果数据发生了变化，指针会自动变为无效。

在第1章中，我们将研究如何使用“哈希函数”来创建这种指针。

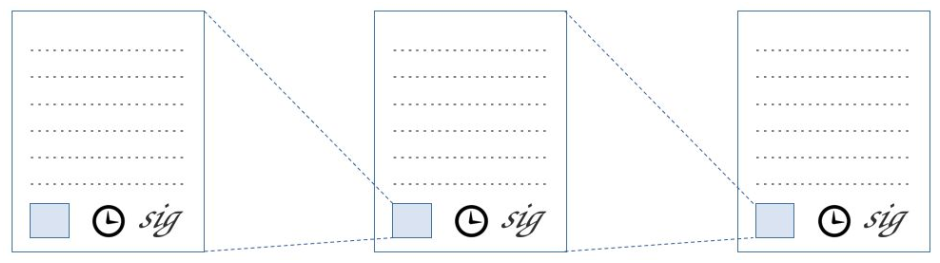
What this achieves is that each document’s certificate ensures the integrity of the contents of the previous document. In fact, you can apply this argument recursively: each certificate essentially fixes the entire history of documents and certificates up until that point. If we assume that each client in the system keeps track of at least a few certificates — their own documents’ certificates, and those of the previous and following documents — then collectively the participants can ensure that the history cannot be changed after the fact. In particular, the relative ordering of documents is preserved.

所实现的是：每个文档的证书保证了前一个文档内容的完整性。（即，如果前一个文档被修改，后一个文档的证书就会无效，所以无法修改前一个文档。）

实际上，你可以递归地应用这个参数：每个证书都固定了到这个时间点为止的文档和证书的全部历史。

（即，在这个时间之前的文档都无法被修改，如果被修改，就会通过证书被发现。）

如果我们假设系统中的每个客户都跟踪至少一些证书（他们自己文档的证书，以及之前和之后的文档），那么，参与者就可以共同保证：在事实发生后，历史就不能被改变了。特别是不能改变文档的先后排序。



**Figure 4: linked timestamping**​ .

To create a certificate for a document, the timestamp server includes a hash pointer to the previous document’s certificate, the current time, and signs these three data elements together.

为了给一个文档创建一个证书，时间戳服务器使用下列信息：当前文档、当前时间、前一个文档的证书的哈希指针。它将这三个数据元素合在一起进行签名。

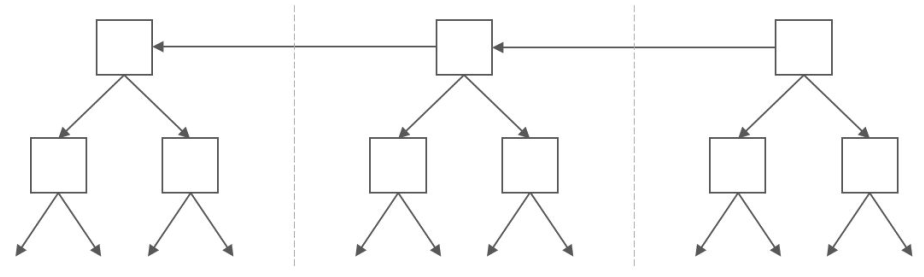
A later paper proposed an efficiency improvement: instead of linking documents individually, we can collect them into blocks and link blocks together in a chain. Within each block, the documents would again be linked together, but in a tree structure instead of linearly. This decreases the amount of checking needed to verify that a particular document appears at a particular point in the history of the system. Visually, this hybrid scheme looks like Figure 5.

后来的一篇论文提出了一个有效的改进：不是单独链接每个文档，而是将一些文档集中到区块中，再把区块串成一个链。

在每个区块中，文档被链接在一起，使用的是树形结构，而不是线性结构。

这减少了检查的数量：验证一个特定文档出现在历史中的某个特定点。

图5显示了这种混合方案。



**Figure 5: efficient linked timestamping**​ .

Arrows represent hash pointers and dotted vertical lines indicate time intervals.

箭头表示哈希指针，虚线表示时间间隔。

This data structure forms the skeleton of Bitcoin’s block chain, as we’ll see in Chapter 3. Bitcoin refines it a subtle but important way: a Hashcash-esque protocol is used to delay how fast new blocks are added to the chain. This modification has profound and favorable consequences for Bitcoin’s security model. There is no longer the need for trusted servers; instead, events are recorded by a collection of untrusted nodes called “miners”. Every miner keeps track of blocks, rather than having to rely on regular users to do it. Anyone can become a miner by solving computational puzzles to create blocks. Bitcoin also gets rid of signatures, relying only on hash pointers to ensure the integrity of the data structure. Finally, the actual timestamps aren’t of much importance in Bitcoin, and the point of the system is to record the relative ordering of transactions in a tamper-resistant way. In fact, Bitcoin blocks aren’t created in a fixed schedule. The system ensures that a new one is created every 10 minutes on average, but there’s considerable variation in the time between successive blocks.

这个数据结构形成了比特币区块链的骨架。

比特币对它做了一个微妙但重要的修改：使用了一个Hashcash-esque协议来确保了新区块加入区块链的速度。

这个修改对比特币的“安全模型”有着深刻而有利的影响。

* 不再需要可信服务器，事件是由一组不可信节点（矿工）记录的。
* 每个矿工都在跟踪区块，而不是依靠普通用户来做。
* 任何人都可以成为矿工，方法是：通过解决计算难题来创建区块。
* 比特币也去掉了签名，只依赖哈希指针来保证数据结构的完整性。

最后，实际的时间戳在比特币中并不太重要，系统的要点是以防篡改的方式记录交易的相对排序。

实际上，比特币区块不是按照固定的时间表创建的。该系统确保平均每10分钟创建一个新的区块，但在连续的区块之间的时间有相当大的变化。

In essence, Bitcoin combines the idea of using computational puzzles to regulate the creation of new currency units with the idea of secure timestamping to record a ledger of transactions and prevent double spending. There were earlier, less sophisticated proposals that combined these two ideas. The first is called b-money, and it was by Wei Dai in 1998. In b-money, anyone can create money using a hashcash-like system. There’s a peer-to-peer network, sort of like in Bitcoin. Each node maintains a ledger, but it’s not a global ledger like in the Bitcoin block chain. Each node has its own ledger of what it thinks everyone’s balance is.

本质上，比特币结合了这两种思想：

* 计算难题：使用计算难题来控制新货币的创建
* 安全时间戳：使用安全时间戳来记录交易的账目，防止双重支付。

在比特币之前，也有一些不太成熟的提议综合了这两种思想。

第一种是b-money，它是Wei Dai在1998年提出的。在b-money中，任何人都可以使用一个类似hashcash的系统创建货币。有一个P2P网络，有点儿像比特币。每个节点维护一个账本，但它不是一个像比特币区块链那样的全球账本。每个节点有它自己的账本：它认为每个人的余额是多少。

Another similar proposal, by Nick Szabo, is called Bitgold. Szabo says he had the idea for Bitgold as early as 1998, but didn’t get around to blogging about it until 2005. The reason I mention this is that there’s a minor conspiracy theory popularized by Nathaniel Popper, a New York Times reporter who wrote a very good book on the history of Bitcoin. Popper notes that the blog post timestamps were changed after Satoshi posted the Bitcoin whitepaper so that the Bitgold proposal looks like it was written up about two months after Bitcoin was released. Popper believes, like many other observers, that Szabo could be Satoshi, and he cites the timestamp change as evidence of Szabo/Satoshi trying to cover up the fact that he invented Bitgold before he knew about Bitcoin.

另一个类似的提议是Bitgold，由Nick Szabo。

Szabo说他早在1998年就有了Bitgold的想法，但直到2005年才在博客上公开发布。

我之所以提到这一点，是因为纽约时报记者Nathaniel Popper曾写了报道了一个广为人知的阴谋论，他写了一本关于比特币历史的好书。

Popper注意到，在中本聪发布了比特币白皮书后，这个博客帖子的时间戳被修改了，因此，Bitgold提议像是在比特币发布两个月后写的。

像许多其他观察者一样，Popper相信，Szabo可能是中本聪，他把时间戳修改作为证据，Szabo试图掩盖在它了解比特币之前就发明了Bitgold的事实。

The problem with this explanation is that if you actually read the contents of the blog posts, Szabo is very clear about having had this idea in 1998, and he doesn’t try to change those dates. So a more reasonable explanation is that he just bumped the post to the top of his blog after Bitcoin popularized similar ideas, to make sure that people were aware of his prior proposal.

这个解释的问题是，如果你真地读了这个博客文章的内容，那么Szabo很清楚在1998年已经有了这个想法，他并没有试图改变那些日期。

因此，一个更合理的解释是，在比特币流行之后，他把他的博客置顶，这样，人们就可以看到他以前的提议。

Bitcoin has several important differences from b-money and Bitgold. In those proposals, computational puzzles are used directly to mint currency. Anyone can solve a puzzle and the solution is a unit of money itself. In Bitcoin, puzzle solutions themselves don’t constitute money. They are used to secure the block chain, and only indirectly lead to minting money for a limited time. Second, b-money and Bitgold rely on timestamping services that sign off on the creation or transfer of money. Bitcoin, as we’ve seen, doesn’t require trusted timestamping, and merely tries to preserve the relative order of blocks and transactions.

比特币与b-money和Bitgold有几个重要的区别。

* 第一，在这些提议中，计算难题被直接用于铸币。任何人都可以解决一个难题，难题的解就是一个货币单位。而在比特币中，难题的解本身并不构成货币。它们被用于保证区块链的安全，只是间接地导致了在一个有限时间内进行铸币。
* 第二，b-money和Bitgold依靠时间戳服务来签署货币的创造或转移。比特币不要求可信的时间戳，只是试图保留区块和交易的相对顺序。

Finally, in b-money and Bitgold, if there is disagreement about the ledger among the servers or nodes, there isn’t a clear way to resolve it. Letting the majority decide seems to be implicit in both authors’ writings. But since anyone can set up a node — or a hundred, hiding behind different identities — these mechanisms aren’t very secure, unless there is a centralized gatekeeper who controls entry into the network. In Bitcoin, by contrast, for an attacker to change history, they must solve computational puzzles at a faster rate than the rest of the participants combined. This is not only more secure, it allows us to quantify the security of the system.

最后，在b-money和Bitgold中，如果在服务器或节点之间存在账目的不一致，则没有明确的解决方法。

让多数决定似乎隐含在两位作者的著作中。但是，由于任何人都可以建立一个节点（或一百个节点，用不同的身份隐藏），这些机制并不十分安全，除非有一个集中式的看门人，他控制网络的入口。

在比特币中，攻击者为了修改历史，必须比其他参与者结合在一起还要快地解决计算难题。这不仅更安全，还允许我们量化系统的安全性。

B-money and Bitgold were informal proposals — b-money was a post on a mailing list and Bitgold was a series of blog posts. Neither took off, or was even implemented directly. Unlike the Bitcoin white paper, there wasn’t a full specification or any code. The proposals gloss over issues that may or may not be solvable. The first, as we’ve already mentioned, is how to resolve disagreements about the ledger. Another problem is determining how hard the computational puzzle should be in order to mint a unit of currency. Since hardware tends to get dramatically cheaper over time for a fixed amount of computing power, Bitcoin incorporates a mechanism to automatically adjust the difficulty of the puzzles periodically. B-money and Bitgold don’t include such a mechanism, which can result in problems since coins may lose their value if it become trivially easy to create new ones.

B-money和Bitgold都是非正式提议：B-money是邮件列表中的一个帖子，Bitgold是一篇博客文章。

他们既没有开始，也没有直接实施。不像比特币白皮书，它们没有一个完整的规范或任何代码。这些提议掩盖了可能无法解决的问题。

第一个问题是，如何解决账目的不一致。

第二个问题是，决定为了铸币，计算难题应该有多难。由于硬件会越来越便宜，而计算能力越来越强，所以比特币包含一种机制，能周期性地自动调整谜题的难度。b-money和Bitgold没有这样的机制，这可能导致问题：因为造币越来越容易，所以货币会失去它们的价值。

**【总结】**

解题：把所有东西记录在一个账本中

* 介绍了比特币的核心：计算难题和区块链

比特币的另一个重要组成部分是“区块链”：它是一个账目，所有比特币交易都被安全地记录在里面。

本质上，比特币结合了这两种思想：

* 计算难题：使用计算难题来控制新货币的创建
* 区块链：使用安全时间戳来记录交易的账目，防止双重支付。

1、计算难题

比特币使用了一个Hashcash-esque协议来确保了新区块加入区块链的速度。

2、区块链

区块链背后的思想可追溯到Haber和Surneta在1991年发表的一篇论文。

他们提议的方法是针对数字文档的“安全时间戳”，而不是针对数字货币方案。

* 时间戳：时间戳的目的是为了标记一个文档是在何时出现的。更重要的是，时间戳准确地记录了先后顺序：如果一个文档比另一文档先出现，时间戳能够反映出这个先后顺序。
* 安全属性：要求在事实发生之后，文档的时间戳不能被更改。

后来的一篇论文提出了一个有效的改进：不是单独链接每个文档，而是将一些文档集中到区块中，再把区块串成一个链。

在每个区块中，文档被链接在一起，使用的是树形结构，而不是线性结构。

这个数据结构形成了比特币区块链的骨架。

3、其它思想

在比特币之前，也有一些不太成熟的提议综合了这两种思想。

第一种是b-money，它是Wei Dai在1998年提出的。

另一个类似的提议是Bitgold，由Nick Szabo。Szabo说他早在1998年就有了Bitgold的想法，但直到2005年才在博客上公开发布。

B-money和Bitgold都只是非正式提议，这些提议掩盖了可能无法解决的问题，而比特币解决这些问题。

* 计算难题：决定为了铸币，计算难题应该有多难。
* 区块链：如何解决账目的不一致。

## 0.6关于中本聪

Hints about Satoshi

You may know that Satoshi Nakamoto is the pseudonym adopted by the creator of Bitcoin. While his identity remains a mystery, he communicated extensively in Bitcoin’s early days. Let’s use this to dig a little bit into questions like when he started working on Bitcoin, to what extent he was influenced by the prior ideas we’ve looked at, and what motivated him.

你可能知道中本聪（Satoshi Nakamoto）是比特币的创造者所使用的笔名。

虽然他的身份仍然是个谜，但在比特币的早期，他还是广泛地与他人进行过交流。

我们利用这一点来挖掘一些问题，例如：

* 他什么时候开始研究比特币？
* 他受到了我们前面提到的那些想法的多大影响？
* 是什么激励了他？

Satoshi says he started coding Bitcoin around May 2007. I’ll take him at his word; the fact that he’s anonymous is not a reason to think he’d lie about things like that. He registered the domain bitcoin.org in August 2008. And at that time, he started sending private emails to a few people who he thought might be interested in the proposal. Then a little later in October 2008, he publicly released a white paper that described the protocol, and then soon after, he released the initial code for Bitcoin as well. Then he stuck around for about two years, during which he posted lots of messages on forums, emailed with lots of people, and responded to people’s concerns. On the programming side, he submitted patches to the code. He maintained the source code in conjunction with other developers, fixing issues as they arose. By December 2010, others had slowly taken over the maintenance of the project, and he stopped communicating with them.

中本聪说，他大约在2007年5月开始编写比特币代码。

* 2008年8月,他注册了域名bitcoin.org。那个时候，他开始发送电子邮件给一些他认为可能对这个提议感兴趣的人。
* 2008年10月，他公布了比特币白皮书，描述了这个协议。
* 2009年1月，他发布了比特币的初始代码。
* 之后两年，他在论坛上张贴了大量的信息，给很多人发电子邮件，并回应了人们的关切。

在编程方面，他提交代码补丁，与其他开发人员一起维护源代码，解决出现的问题。

* 2010年12月，其他人已经逐渐接管了项目的维护，他也停止了与他们的交流。

I’ve been referring to Satoshi Nakamoto as a “he,” but I have no particular reason to believe Satoshi is a man and not a woman. I’m just using the male pronoun since Satoshi is a male name. I’ve also been referring to him as a single individual. There is a theory that Satoshi Nakamoto might be a collection of individuals. I don’t buy this theory — I think Satoshi is probably just one person. The reason is that if we look at the entirety of the online interactions undertaken under the Satoshi pseudonym, if we think about the two years that Satoshi spent replying to emails and patching code, it’s hard to imagine that this could be multiple people sharing user accounts and passwords, responding in a similar style and a similar voice, and making sure they didn’t contradict each other. It just seems a much simpler explanation that at least this portion of Satoshi’s activity was done by a single individual.

我认为中本聪是一个人。有一种看法认为中本聪可能是一群人，我不相信这个看法，因为，我们可以看看中本聪进行的所有在线交流，如果我们考虑到中本聪花了两年时间回复电子邮件和修补代码，很难想象这可能是多人共享的帐户和密码，他们还要用相似的风格和相似的声音，以确保他们之间没有互相矛盾。一个更简单的解释似乎是，至少中本聪的活动这一部分是由一个人完成的。

Furthermore, it’s clear from his writings and patches that this individual understood the full code base of Bitcoin and all its design aspects. So it’s very reasonable to assume that the same individual wrote the original code base and the white paper as well. Finally, it’s possible that Satoshi had help with the original design. However, after Bitcoin’s release, we can see for ourselves that Satoshi was quick to attribute any help he received from other contributors. It would be out of character for him to mislead us about inventing something by himself if he had had help from other people.

此外，从他的著作和补丁中可以清楚地看出，这个人理解比特币的全部代码和所有设计。

因此，这个假设是合理的：一个人写了原始代码和白皮书。

最后，有人可能在开始时帮助过中本聪。然而，在比特币发布后，我们可以看到，中本聪很快就对其他贡献者表示感谢。

从这一点来说，他应该不会在接受别人的帮助后闭口不言，故意误导我们想象比特币是他一个人的作品。

Next, we might ask ourselves, “What did Satoshi know about the history of ecash?” To understand this better, we can start by looking at what he cites in his white paper as well as the references that existed on early versions of the Bitcoin website. In the white paper he cites some papers on basic cryptography and probability theory. He also cites the time-stamping work that we saw earlier, and it’s very natural to think that he based the design of the block chain on these references since the similarities are so apparent. He also cites the Hashcash proposal whose computational puzzle is very similar to the one that’s used in Bitcoin. He also has a reference to b-money. Later, on the website, he added references to Bitgold and as well to a scheme by Hal Finney for reusing computational puzzle solutions.

接下来，我们可能会问：中本聪对ecash的历史了解多少？

为了更好地理解这一点，我们可以从这里开始：他在白皮书中引用的内容，以及比特币网站早期版本上的参考。

在白皮书中，他引用了一些论文，是关于基本的密码学和概率论。

他还引用了我们前面看到的时间戳工作，很自然地认为他是基于这些参考做了区块链的设计，因为它们的相似性是显而易见的。

他还引用了Hashcash提议，其计算难题与比特币中使用的计算难题非常相似。

他还引用了b-money。

后来，在网站上，他增加了对Bitgold的引用，以及Hal Finney的方案（重用计算难题）。

But, if we look at the email exchanges that were made public by people who corresponded with Satoshi Nakamoto in the early days, we find that the b-money proposal was actually added after-the-fact, at the suggestion of Adam Back. Satoshi then emailed Wei Dai who created b-money and apparently, Dai was the one that told him about Bitgold. So these proposals weren’t probably inspirations for the original design. He later corresponded a lot with Hal Finney, and that’s quite a reasonable explanation for why he cites Finney’s work, at least on the website.

但是，如果我们看看那些在早期与中本聪通信的人所公开的电子邮件交流，我们发现b-money提议实际上是根据Adam Back的建议在事后添加的。

然后，中本聪发电子邮件给Wei Dai（他创造了b-money），显然，Wei Dai向他介绍了Bitgold。

因此，这些建议可能不是最初设计的灵感。

后来他和Hal Finney进行了大量的交流，这也解释了为什么他引用了Finney的作品，至少在网站上。

Based on this, it seems plausible that when creating Bitcoin, Hashcash and time-stamping were the only things from the history of ecash that Satoshi knew about or thought were relevant. After he came to know of b-money and Bitgold, however, he seems to have appreciated their relevance. In mid-2010, the Wikipedia article on Bitcoin was flagged for deletion Wikipedia’s editors because they thought it wasn’t noteworthy. So there was some discussion between Satoshi and others about how to word the article so that Wikipedia would accept it. To that end, Satoshi suggested this description of Bitcoin: “Bitcoin is an implementation of Wei Dai’s b-money proposal on Cypherpunks in 1998 and Nick Szabo’s Bitgold proposal.” So Satoshi, by this point, did see positioning Bitcoin as an extension of these two ideas or an implementation of these two prior systems as a good explanation of how it worked.

基于此，这似乎是有道理的：当创建比特币时，Hashcash和时间戳是来自ecash的历史的唯一的东西，中本聪了解它们或认为与之相关。

在他了解了b-money和Bitgold之后，他似乎已经认识到了它们的相关性。

在2010年中期，维基百科关于比特币的文章被标记为删除维基百科的编辑，因为他们认为它没有价值。

因此，中本聪和其他人进行了一些讨论“如何写这篇文章”，以便使维基百科能够接受。

为此，中本聪提出了对比特币这样描述：比特币是基于下列的一个实现，Wei Dai 1998年在Cypherpunks上的b-money提议，以及Nick Szabo的Bitgold提议。

因此，在这一点上，中本聪确实把比特币定位为这两个提议的扩展，或这前两个系统的实现，以便很好地解释它如何工作。

But, what about everything else — the Chaumian ecash schemes and the credit card proposals that we looked at? Did Satoshi know any of that history when designing Bitcoin? It’s hard to tell. He didn’t give any indication of knowing that history, but it’s just as likely that he didn’t reference this because it wasn’t relevant to Bitcoin. Bitcoin uses a completely different decentralized model and so there’s no compelling reason to dwell on old centralized systems that failed.

但是，其它东西呢？

我们提到过的Chaumian ecash方案和信用卡提议？

中本聪在设计比特币时知道它的历史吗？

很难说清楚，他没有给出任何知道历史的迹象，但他没有提到这一点，很可能是因为这与比特币无关。比特币使用了完全不同的去中心化模型，因此没有足够的理由去考虑旧的已经失败的中心化系统。

Satoshi himself makes this point, by mentioning Chaumian ecash in passing, in one of his posts to the Bitcoin forums. Writing about another proposal called opencoin.org, he notes that they seem to be “talking about the old Chaumian central mint stuff, but maybe only because that was the only thing available. Maybe they would be interested in a new direction. A lot of people automatically dismiss e-currency as a lost cause because of all the companies that failed since the 1990’s. I hope it’s obvious it was only the centrally controlled nature of those systems that doomed them. I think this is the first time we’re trying a decentralized, non-trust-based system.” That gives us a pretty good idea what Satoshi thought of the earlier proposals, and specifically how he felt Bitcoin was different. Bitcoin’s decentralization is indeed a defining feature that sets it apart from almost everything we’ve look at.

中本聪本人也表明了这一点，他在Bitcoin论坛上发表的文章时提到了Chaumian ecash。

他写了另一个叫opencoin.org的提议，他说他们似乎在“讨论老的Chaumian中心化造币体系，但可能只是因为这是唯一可用的东西。也许他们会对一个新的方向感兴趣。许多人并不看好电子货币，因为1990年以来所有的公司都失败了。我希望很明显，这些系统的中心化控制性质注定了它们的失败。我认为这是我们第一次尝试一个去中心化的、基于非信任的系统。”

这让我们很好地理解了中本聪对早期提议的看法，特别是他觉得比特币与众不同的地方是什么。

比特币的去中心化确实是它与众不同的特征。

Another interesting quote from Satoshi suggests that he might not be an academic. Most academic researchers think about ideas and write them down immediately, before they build the system. Satoshi says that he took an opposite approach: “I actually did Bitcoin kind of backwards. I had to write all the code before I could convince myself that I could solve every problem, then I wrote the paper. I think I will be able to release the code sooner than I could write a detailed specification.”

另一个来自中本聪的话，可能暗示中本聪不是一个学者。

多数学术研究者思考ideas，立刻写下来，然后才构建系统。

中本聪说他采取了相反的做法：“我在做比特币时是倒着来的。我先写了所有代码，以说服自己能够解决所有问题，然后才写了论文。我想我写代码并我写详细规范要快。”

Since there’s bit of myth around Satoshi, it’s worth mentioning that he made mistakes like everyone else and that wasn’t a perfect oracle of the future. There are bugs and questionable design choices in the original Bitcoin code as well as in its design. For example, there was a feature to send bitcoins to IP addresses that never caught on and, in retrospect, was a bad idea. When he described what Bitcoin was useful for, his scenarios were centered on the idea of using it across the internet. That use case is central to Bitcoin, of course, but it’s not the only one. He didn’t indicate a vision of going into a coffee shop and being able to pay for your coffee with Bitcoin, for example.

因为中本聪有点儿神秘，值得一提的是：他像其他人一样也会犯错，这不是未来的完美预言。

在最初的比特币代码和设计中，都存在错误和可疑的设计选择。

例如，有一个特性将比特币发送到IP地址，回想起来，这是一个坏主意。

当他描述了比特币的用处时，他的情景集中在互联网上的使用。

那个用例是比特币的核心，当然，并不是唯一的。

例如，他没有表示这样的摄像：到咖啡店用比特币买咖啡。

A final question we may ask ourselves, colored by what we understand from the history of digital cash, is, “Why does Satoshi maintain his anonymity?” There are many possible reasons. To begin with, it might be just for fun. Many people write novels anonymously, and there are graffiti artists like Banksy who maintain their anonymity. In fact, in the community that Satoshi was involved in at that time, the Cypherpunk community and the cryptography mailing list, it was common practice for people to post anonymously.

从我们对数字现金的历史的理解来看，我们问的最后一个问题是：为什么中本聪要保持匿名？

有很多可能的原因。

首先，可能只是好玩。许多人匿名写小说，还有像Banksy这样的涂鸦艺术家，他们保持匿名。事实上，在当时，中本聪参与的社区、Cypherpunk社区和密码学邮件列表中，人们经常匿名发帖。

On the other hand, there could have been legal worries behind Satoshi’s choice. Two U.S. companies, Liberty Reserve and e-Gold, ran into legal trouble for money laundering. In 2006, one of the founders of Liberty Reserve fled the United States, fearing that he would be indicted on money laundering charges. E-Gold’s founders, on the other hand, stayed in the United States, and one was actually indicted and eventually pled guilty to the charges. This guilty plea was registered just right before Satoshi set up the Bitcoin website and started emailing people about his proposal. That said, numerous people have invented ecash systems, and nobody else was scared of the legal implications or has chosen to remain anonymous. So this may have been the reason, it may not have been the reason.

另一方面，中本聪的选择背后可能存在法律上的担忧。

两个美国公司，自由储备和e-Gold，因洗钱而陷入法律麻烦。

2006，自由储备的创始人之一逃离美国，他担心被指控洗钱罪。

而e-Gold的创始人留在美国，其中一人被起诉并最终被指控有罪。

在中本聪建立比特币网站并开始向人们发送他的提议之前，这一认罪请求刚刚被登记。也就是说，很多人已经发明了ecash系统，但没有人害怕法律暗示，或者选择匿名。所以这可能是原因，也可能不是原因。

It’s also worth recalling that certain aspects of ecash were patented, and that members of the Cypherpunk movement were concerned about implementing ecash systems due to these patents. In fact, one post to the cypherpunks mailing list proposed that a group of anonymous coders implement ecash so that if someone were to sue, they wouldn’t be able to find the coders. While it is difficult to think that Bitcoin would violate the ecash patents given how different its design is, perhaps Satoshi was being extra cautious. Or maybe he was just inspired by the idea of an anonymous coder from the cypherpunk community.

值得一提的是，ecash的某些方面获得了专利，而Cypherpunk 运动的成员担心实现ecash系统会侵犯这些专利。

事实上，cypherpunks邮件列表的一个帖子建议，一组匿名编码者实现ecash，这样如果有人起诉，他们将无法找到编码者。

虽然很难想象比特币会侵犯ecash的专利，它的设计是多么不同，但也许中本聪对此格外谨慎。

或许他的灵感只是来自cypherpunk 社区的一名匿名程序员的想法。

A final reason that’s often cited is personal security. We know that Satoshi has a lot of bitcoins from his mining early on, and due to Bitcoin’s success these are now worth a lot of money. I think this is a plausible reason. After all, choosing to be anonymous isn’t a decision you make once, it’s something that you do on a continual basis. That said, it probably wasn’t Satoshi’s original reason. The first time Satoshi used the name Satoshi Nakamoto, he hadn’t even released the whitepaper or the codebase for Bitcoin, and it’s hard to imagine that he had any idea that it would be as successful as it was. In fact, at many points in its early history, Satoshi was optimistic but cautious about Bitcoin’s prospects. He seems to have understood that many previous efforts had failed and that Bitcoin might fail as well.

人们经常引用的最后一个原因是人身安全。

我们知道中本聪从他早期的挖矿中得到了大量的比特币，而且由于比特币的成功，这些已经很值钱了。

我认为这是一个合理的理由。毕竟，选择匿名是不是你做出的一次决定，而是你在连续基础上做的事情。

尽管如此，这可能不是中本聪最初的原因。中本聪第一次使用这个名字时，他甚至没有发布比特币白皮书或代码，很难想象他会想到比特币会如此成功。事实上，在其早期历史的许多时刻，中本聪对比特币的前景持乐观而谨慎的态度。他似乎已经明白，以前的许多努力都失败了，比特币也可能失败。

## 0.7结语

Concluding remarks

The success of Bitcoin is quite remarkable if you consider all the ventures that failed trying to do what it does. Bitcoin has several notable innovations including the block chain and a decentralized model that supports user-to-user transactions. It provides a practically useful but less-than-perfect level of anonymity for users. In Chapter 6 we’ll take a detailed look at anonymity in Bitcoin. In one sense it’s weaker than the strong anonymity in DigiCash, but in another sense it’s stronger. That’s because in DigiCash, it was only the senders of the money that maintained their anonymity, and not the merchants. Bitcoin gives both senders and merchants (whether users or merchants) the same level of anonymity.

比特币的成功非常显著，如果你考虑了所有尝试失败的冒险家。

比特币有几个显著的创新，包括区块链和一个去中心化模型，它支持用户对用户的交易。

它为用户提供了实用，但不完美的匿名级别。在第6章中，我们将详细研究比特币中的匿名性。

从某种意义上说，它比DigiCash中的强匿名性要弱，但在另一个意义上，它更强大。这是因为在DigiCash中，只有钱的发送者保持了他们的匿名性，而商人没有匿名性。比特币给发送者和商家（无论是用户还是商家）的匿名程度相同。

Let me conclude with some lessons that we can learn from Bitcoin through the lens of the previous systems that we’ve looked at. The first is to not give up on a problem. Just because people failed for 20 years in developing digital cash doesn’t mean there isn’t a system out there that will work. The second is to be willing to compromise. If you want perfect anonymity or perfect decentralization you’ll probably need to worsen other areas of your design. Bitcoin, in retrospect, seems to have made the right compromises. It scales back anonymity a little bit and requires participants to be online and connected to the peer-to-peer network, but this turned out to be acceptable to users.

我来总结一下我们可以从比特币中吸取的经验教训，通过了解之前的一些系统。

* 第一是对一个问题不要放弃。仅仅因为人们在开发数字现金方面失败了20年，并不意味着没有一个可以工作的系统。
* 第二是愿意妥协。如果你想要完美的匿名或完美的去中心化，你可能需要弱化你的设计的其它领域。比特币似乎做了正确的妥协。它稍微减少了匿名性，要求参与者在线并连接到P2P网络，但这对用户来说是可以接受的。

A final lesson is success through numbers. Bitcoin was able to build up a community of passionate users as well as developers willing to contribute to the open-source technology. This is a markedly different approach than previous attempts at digital cash, which were typically developed by a company, with the only advocates for the technology being the employees of the company itself. Bitcoin’s current success is due in large part to the vibrant supporting community who pushed the technology, got people using it, and got merchants to adopt it.

最后一个经验是通过数字取得成功。

比特币能够建立一个社区，这里不仅有充满激情的用户，还有愿意为开源技术做出贡献的开发者。

这是一个明显不同于以前的数字现金的尝试，以前通常是由一家公司开发，该技术的唯一倡导者是这家公司。比特币目前的成功很大程度上是由于活跃的支持社区，他们推动技术、让人们使用它，并让商家采用它。

## 0.8延伸阅读

Further Reading

An accessible overview of digital cash schemes focused on practical issues:

一篇关于“数字现金”的综述，浅显易懂，注重实际问题：

**P. Wayner. Digital Cash: commerce on the net (2nd ed). Morgan Kaufmann, 1997.**

A cryptographically-oriented overview of e-cash systems (Chapter 1) and micropayments (Chapter 7):

从密码学角度看e-cash系统（第1章）和小额支付（第7章）  
**B. Rosenberg (ed.) Handbook of Financial Cryptography and Security. CRC Press, 2011.**

Although not Chaum’s earliest paper on e-cash, this is arguably the most innovative, and it formed a template replicated by many other papers:

虽然不是Chaum的关于e-cash的最早论文，但是最有创新性的，它被许多其它论文引用。  
**D. Chaum, A. Fiat, M. Naor. Untraceable electronic cash. CRYPTO 1998.**

Many papers improved the efficiency of Chaum-Fiat-Naor using modern cryptographic techniques, but arguably the most significant is:

许多论文使用现代密码学技术改进了Chaum-Fiat-Naor的效率，但是最有影响的是：  
**J. Camenisch, S. Hohenberger, A. Lysyanskaya, Compact e-cash. Theory and Applications of  
Cryptographic Techniques, 2005**

Some practical security observations on the financial industry and proposals, including Mondex:

对金融行业和提案（包括Mondex）的一些实际的安全观察  
**R. Anderson. Security Engineering (2nd ed). Wiley, 2008.**

An overview of the implementation of Chaum’s ecash proposal:

Chaum的ecash提议的实现的概述  
**B. Schoenmakers. Basic security of the ecash payment system. State of the Art in Applied  
Cryptography, 1997.**

Two papers cited by Satoshi Nakamoto in the Bitcoin whitepaper that are integral to Bitcon’s design:

被中本聪和比特币白皮书引用的两篇论文，它们是比特币设计的一部分。  
**A. Back. Hashcash - A Denial of Service Counter-Measure, Online, 2002.  
S. Haber, W. S. Stornetta. Secure names for bitstrings. CCS, 1997.**

## 0.9前言的总结

本节对前言原文做了总结。

### 0.1传统金融体系

|  |  |  |
| --- | --- | --- |
| **交易系统** | **优点** | **缺点** |
| **物物交换** |  | 需要相互协调：在同一时间、同一地点，安排一组人，他们的供需能够一致 |
| **信用** | 不需要启动运转 | 债权人要冒一些风险，因为债务人可能不偿还债务。 |
| **现金** | 现金能让我们准确衡量物品的价值。  避免了买家拖欠债务的可能性。 | 需要“初始分配一些现金”来启动这个系统的运转，没有“现金”就没法做交易。 |
| **混合系统** | 我们今天使用的是一种混合系统：  即使我们在使用信用时，也用现金来计算要偿还的债务。 | |

这些想法（物物交换、现金、信用）出现在许多情况下，尤其是网上系统，用户交易的是某种虚拟商品。

例如：“P2P文件共享网络”使用“虚拟现金”来实现文件交换。

### 0.2信用类电子支付

|  |  |  |
| --- | --- | --- |
| **信用类电子支付方法** | | |
| **1非中介体系结构**  **例如：**  **Amazon** | **处理** | 1. 你输入信用卡信息，发送给Amazon 2. Amazon用这些信息与“系统”进行对话，它是一个金融系统，涉及processor、银行、信用卡公司、其它中介机构。 |
| **优点** | 直接与卖家互动 |
| **缺点** | 信用卡信息提供给商家可能是一个风险 |
| **2中介体系结构**  **例如：FirstVirtual**  **PayPal** | **处理** | 有一个中介在你和卖家之间，你把信用卡信息发给这个中介，它批准交易并通知卖家。这个中介每天会与卖家进行结算。  FirstVirtual公司成立于1994年，是一家早期的支付中介公司。FirstVirtual提出的系统有点像PayPal，但比它早很多年。  用户向FirstVirtual注册，并提供信用卡信息。当你要买东西时，卖家会联系FirstVirtual，告知支付信息，FirstVirtual向你确认这些信息，以及你是否同意使用信用卡进行支付。  两个细节：所有沟通都是通过email进行的。 客户有90天时间来质疑收费，商家只有在90天后才收到这笔钱！ |
| **优点** | 不必向卖家提供你的身份，这也会提高你的隐私。 |
| **缺点** | 你和卖家都必须在中介那里有一个账户。  你失去了与卖家直接互动的简单性。 |
| **3 SET体系结构**  **例如：CyberCash** | **处理** | 90年代中期，有一种与“中介体系结构”竞争的方法，称为“SET体系结构”。  CyberCash公司实现了SET，CyberCash在2001年破产，知识产权被Verisign收购，Verisign又转手卖给了PayPal，PayPal今天还活着。  SET的购物流程：   * 用户操作：   + 浏览器把你的“交易细节视图”传给本机的“购物应用程序”   + 购物应用程序再加上你的信用卡信息，对这些信息进行加密（只有中介可以解密）。   + 你把加密数据发送给商家。 * 商家操作：商家把数据转给中介，包括商家自己的交易细节视图。 * 中介操作：中介解密你的数据，只有当你和商家的视图匹配时，才批准这个交易。 |
|  | **优点** | SET避免了客户向商家提供信用卡信息。  也不需要用户向中介进行注册。 |
|  | **缺点** | 根本的问题是与“证书”有关。证书是一种方法，它将“加密身份（即公钥）”安全地与“实际身份”关联起来。  网站需要从Verisign这类公司（称为“认证机构”）获取证书，目的是在你的浏览器判断它是安全的（通常由一个锁图标指示）。  CyberCash和SET认为安全性重于可使用性，所以他们要求，不仅processors和商家要获得证书，所有用户也要获得证书。  获得证书就像纳税一样繁琐，所以这个系统是一场灾难。在过去几十年里，主流用户都排斥任何需要最终用户证书的系统，这些提案现在已被沦为了学术论文。 |
| **4 Web联盟** | **处理** | 90年代中期，当SET被标准化时，Web联盟也想对金融支付进行标准化。  他们想通过扩展HTTP协议来实现，这样用户在做交易时就不需要额外的软件，只需要用浏览器。实际上，他们有一个更通用的提案“如何扩展这个协议”，其中一个用例就是支付。  2015年，Web联盟宣布它想采用另一种方法，这次，比特币将成为标准化的一部分。 |
|  | **优点** | 用户在做交易时就不需要额外的软件，只需要用浏览器。 |
|  | **缺点** | 从未实现过，整个扩展框架从未部署在任何浏览器中。 |

### 0.3现金类电子支付

1、**银行期票：签名**

如何解决“双重支付”问题：加个序列号，跟踪每个期票

**2、DigiCash公司的Ecash**

1983年，David Chaum最早提出了把“密码学”应用于“现金”的想法。

采用的方法：盲签名

仍然需要由中央机构（例如银行）运营一个服务器，并让每个人都信任这个结构。

服务器要参与每个交易。如果服务器出问题，支付就会停止。

1988年，Chaum与Fiat和Naor一起提出了“离线电子现金”。

他们的想法是：停止“为防止双花而担忧”，而是专注于“在事后检测双花”，在商家重新连接银行服务器时检测双花。

Chaum-Fiat-Naor检测双花的方法是：使用密码学。

在高层次上，它实现的是：

* 发给你的每个数字货币都编码了你的身份，但只有你能解码它，即使银行也无法解码。
* 当你花钱时，收款人要求你解码那个编码的一部分，然后记下来。这个解码不足以确定你的身份。
* 如果你做了双花，当两个收款人到银行兑换票据时，银行会把这两个信息放在一起，从而确定你的身份。

1989年，Chaum在成立了DigiCash公司，这可能是最早尝试解决网上支付问题的公司。

DigiCash系统中的现金被称为Ecash

* Ecash是基于Chaum的协议。
* 客户是匿名的，所以银行无法追踪客户的花钱方式。但ecash中的商家不是匿名的。
* Ecash不能分割货币

交易方法：

当你想做交易时，例如，你想给EPIC捐款，你点击一个捐赠链接，进入DigiCase网站。

然后打开一个反向Web连接，回到你的计算机。这意味着，你的计算机必须有能力作为服务器接受进入的连接。你必须有自己的IP地址，并且你的ISP还要允许进入连接。

如果连接成功，那么ecash软件将在你的计算机上启动，你就能够批准交易和转钱了。

**为什么DigiCash失败了？**

DigiCash的主要问题是：很难说服银行和商家采用它。

由于没有很多商家接受ecash，用户也不想用它。

更糟糕的是，它不支持“用户-用户交易”，或者至少不是很好。

它实际上集中在“用户-商家交易”。因此，如果商家不同意，就没有方法来激发对系统的兴趣。

所以，DigiCash最终失败了，而信用卡公司赢了。

在DigiCash的后期，也尝试了防篡改硬件，来防止双重支付，而不仅仅是检测它。

围绕这一想法而组建的另一家公司是Mondex，但并没有真正流行起来。

* 一个主要问题是它就像现金一样，如果丢了或被盗，钱就没了。

更糟的是，如果卡片出现了故障，如果读卡器不能读卡，就没有办法知道这张卡是否有余额。

在这些情况下，Mondex一般会支付成本。他们会假设卡片有钱，并赔偿用户的损失。当然，这会让公司花很多钱。

* 钱包很慢，而且笨拙。用信用卡或现金支付要快得多。
* 零售商讨厌有许多付款终端，他们只想要一个信用卡终端。

所有这些因素共同影响了Mondex。

但是，这些卡片所使用的智能卡技术已证明是成功的。

**3、对Chaum-Fiat-Naor方案的改进**

多年来，许多密码学家已经研究了这种结构，并以用各种方法来改进它。

* 在Chaum-Fiat-Naor方案中，不能分割货币。

Okamoto和Ohta写了一篇论文，使用“Merkle树”来创建一个系统，这个系统可以让你分割货币。

比特币也用到了Merkle树，我们将在第1章介绍。

* Chan-Fiat-Naor方案还在效率上留下了很大的改进空间。

特别是，对这个机制应用“零知识证明”成果显著。

比特币也用到了零知识证明，我们将在第6章介绍。

**4、新的尝试**

Chaum在DigiCash技术上有几项专利，特别是它使用的“盲签名”机制。

但是一群密码学者（他们在Cypherpunks邮件列表中）想要另一种选择。

Ben Laurie提出了一个提案Lucre。

Lucre试图用一个非专利方法替代电子现金中的“盲签名”机制，而其它部分都相同。

Ian Goldberg提出的另一个提议，试图解决不能分钱找零的问题。

为了解决Chaum-Fiat-Naor方案的商家匿名性问题，Goldberg提出了一个建议：有不同类型的货币（允许这些交易发生），允许你得到找零，并且仍然保持你的匿名性。

|  |  |  |
| --- | --- | --- |
|  | **DigiCash** | **改进** |
| 1 | “盲签名”是专利 | Ben Laurie提出了一个提案Lucre。  Lucre试图用一个非专利方法替代电子现金中的“盲签名”机制，而其它部分都相同。 |
| 2 | 需要中央机构 | 比特币是去中性化的 |
| 3 | 不能分割货币 | Okamoto和Ohta写了一篇论文，使用“Merkle树”来创建一个系统，这个系统可以让你分割货币。  比特币也用到了Merkle树（第1章） |
| 4 | 商家匿名性问题 | Ian Goldberg提出的另一个提议，试图解决不能分钱找零的问题。  为了解决Chaum-Fiat-Naor方案的商家匿名性问题，Goldberg提出了一个建议：有不同类型的货币（允许这5些交易发生），允许你得到找零，并且仍然保持你的匿名性。  在比特币中，不区分用户和商家，都是匿名的。 |
| 5 | 效率 | 特别是，对这个机制应用“零知识证明”成果显著。  比特币也用到了零知识证明（第6章） |
| 6 | 很难说服银行和商家采用它 | 比特币的策略：把比特币发给其他用户，而社区鼓动对比特币的支持，让商家接受它。 |
| 7 | 防篡改硬件 | ddk：比特币在硬件钱包中用到了这个技术？ |
| 8 | 智能卡技术 | 现在，每张信用卡和借记卡现在都有智能卡技术。  然而，它被用于不同的目的。不是用来防止双花，因为它不是基于现金的技术，所以不会出现这个问题。  银行（不是卡片）跟踪余额或可用额度。芯片被用于认证，即，证明你知道与你的账户关联的PIN。 |

### 0.4比特币：让货币本身具有独立的价值

解题：凭空发行货币

* 即使数字货币本身独立的价值
* 方法是：解决难题，哈希函数、挖矿

在DigiCash系统中，如果你有一个价值100美元的数字现金，是什么使它真得值100美元呢？

有一些不同的提案来实现这个，不同的公司也有不同的做法。

所有这些想法最终是把“数字现金”的价值与美元或商品锚定。

一种完全不同的想法是：让数字货币本身成为货币，它的发行和价值独立于任何其它货币。

为了创造一种能够获得真正价值的自由浮动的数字货币，你需要设计一些稀缺的东西。

在数字领域中，实现稀缺性的一种方法是：为了得到货币，必须解决一个计算难题，它要花一些时间。

这就是比特币中的“挖矿”。（第5章）

基本的想法相当古老：计算难题的解可以是有一定价值的数字对象。

这是密码学着Dwork和Naor在1992年首次提出的，用于解决垃圾邮件的一个可能方案。

一个类似的想法后来在1997被Adam Back独立地发现，用在一个名为Hashcash的建议中。

比特币基本上使用与Hashcash相同的计算难题，但有一些小的改进。

这些计算难题需要有一些特定的属性，才能具有价值：

1. 发送者不能只解决一个难题
2. 接收者应该能够很容易检查难题的解，而不必重复解决难题的过程。
3. 每个难题应该完全独立于其它难题，这样解决一个难题不会减少解决其它难题所需的时间。
4. 由于硬件的升级，导致解决难题会变得更快、更容易，收件人应该能够调整难题的难度。

通过使用“密码哈希函数”来设计难题，能够获得这些属性。（第1章）

正如在密码学中的任何好想法一样，也有许多计算难题的变体，它们是为了实现稍微不同的属性。

一个建议来自Rivest和Shamir（RSA公司）。

他们想设计这样的计算难题：模仿纸币的属性，使铸造第一个货币很难，但后续铸造能简单得多。

他们的建议也是利用哈希函数，但方式不同。

### 0.5比特币：把所有交易记录在区块链中

解题：把所有东西记录在一个账本中

* 介绍了比特币的核心：计算难题和区块链

比特币的另一个重要组成部分是“区块链”：它是一个账目，所有比特币交易都被安全地记录在里面。

本质上，比特币结合了这两种思想：

* 计算难题：使用计算难题来控制新货币的创建
* 区块链：使用安全时间戳来记录交易的账目，防止双重支付。

1、计算难题

比特币使用了一个Hashcash-esque协议来确保了新区块加入区块链的速度。

2、区块链

区块链背后的思想可追溯到Haber和Surneta在1991年发表的一篇论文。

他们提议的方法是针对数字文档的“安全时间戳”，而不是针对数字货币方案。

* 时间戳：时间戳的目的是为了标记一个文档是在何时出现的。更重要的是，时间戳准确地记录了先后顺序：如果一个文档比另一文档先出现，时间戳能够反映出这个先后顺序。
* 安全属性：要求在事实发生之后，文档的时间戳不能被更改。

后来的一篇论文提出了一个有效的改进：不是单独链接每个文档，而是将一些文档集中到区块中，再把区块串成一个链。

在每个区块中，文档被链接在一起，使用的是树形结构，而不是线性结构。

这个数据结构形成了比特币区块链的骨架。

3、其它思想

在比特币之前，也有一些不太成熟的提议综合了这两种思想。

第一种是b-money，它是Wei Dai在1998年提出的。

另一个类似的提议是Bitgold，由Nick Szabo。Szabo说他早在1998年就有了Bitgold的想法，但直到2005年才在博客上公开发布。

B-money和Bitgold都只是非正式提议，这些提议掩盖了可能无法解决的问题，而比特币解决这些问题。

* 计算难题：决定为了铸币，计算难题应该有多难。
* 区块链：如何解决账目的不一致。

### 0.6中本聪

中本聪说，他大约在2007年5月开始编写比特币代码。

* 2008年8月,他注册了域名bitcoin.org。那个时候，他开始发送电子邮件给一些他认为可能对这个提议感兴趣的人。
* 2008年10月，他公布了比特币白皮书，描述了这个协议。
* 2009年1月，他发布了比特币的初始代码。
* 之后两年，他在论坛上张贴了大量的信息，给很多人发电子邮件，并回应了人们的关切。

在编程方面，他提交代码补丁，与其他开发人员一起维护源代码，解决出现的问题。

* 2010年12月，其他人已经逐渐接管了项目的维护，他也停止了与他们的交流。

中本聪说：“讨论老的Chaumian中心化造币体系，但可能只是因为这是唯一可用的东西。也许他们会对一个新的方向感兴趣。许多人并不看好电子货币，因为1990年以来所有的公司都失败了。很明显，这些系统的中心化控制性质注定了它们的失败。我认为这是我们第一次尝试一个去中心化的、基于非信任的系统。”

这让我们很好地理解了中本聪对早期提议的看法，特别是他觉得比特币与众不同的地方是什么。

比特币的“去中心化”确实是它与众不同的特征。

### 0.7结语

比特币有几个显著的创新：

* 区块链和去中心化模型，它支持用户对用户的交易。
* 匿名性：它为用户提供了实用，但不完美的匿名级别。