Imagine you want to create a digital currency that does not rely on a centralized institution such as a bank or government. Why will people believe that your currency has any value when offered as payment for a transaction? How will you prevent users from creating an unlimited amount of the currency or copying a digital coin and spending it twice?

In today's economies, people assign value to money when they have some trust in the central bank of the issuing government. Governments need to be trusted not to flood the economy with lots of money, and generally to commit to pay wages to government workers in the government-backed currency.

Physical money such as coins and bills tends to be inconvenient for both consumers and merchants, and is rarely used for business-to-business transactions in developed countries. Yet, the status quo for completing electronic transactions involves costly fees, with merchants paying as much as 3% for credit card transactions, and with international wire transfers costing tens of dollars per transaction. The promise of a digital currency, where money is strings of bits, is that completing a transaction becomes no more costly than transferring a few bits from one computer to another.

In this chapter we are interested in the design of a system for digital currency that, unlike the U.S. dollar, the Euro, or systems such as PayPal, does not rely on a central institution such as a government or a firm to perform the role of a bank. The challenge is that people must still trust the currency and, if it is to be used for general transactions, the currency should be liquid enough that it can be exchanged for goods and services. Of particular concern with a digital currency is preventing users from easily creating new amounts of money, from copying and spending money more than once (called double spending).

Section 23.1 provides some background on the role of money in economies, and introduces gold-backed currencies and fiat currencies. A fiat currency has no intrinsic value but is associated with value because it comes to be trusted as a store of value. The digital currencies that we study will be fiat currencies.

Section 23.3 introduces a first approach to the design of a digital currency system, with the trade of promises between participants. The iOwe protocol is designed to be used in a setting with a standard unit of accounting, perhaps bandwidth in a peer-to-peer file sharing system or distance in a ride sharing network. Anyone can create a promise to provide some quantity of a resource or some quantity of work. By trading these promises they become a currency. iOwe uses cryptographic methods to generate proofs of double spending. The iOwe protocol also expects users to bootstrap trust through barter transactions and to limit transactions to those with trusted partners. This bootstrap phase limits the usefulness of its currency.
Section 23.4 introduces the Bitcoin digital currency and the peer-to-peer blockchain technology upon which it is built. Bitcoin was the first digital currency to succeed at a large scale and in August 2016 had a market cap of more than US $9 billion, with 1 Bitcoin trading for US $587 (having reached a peak price of over US $1,150 in November 2013), and with more than 15 million coins in circulation. Bitcoin addresses double spending by providing a decentralized, trusted ledger (called the blockchain) to record every Bitcoin transaction ever completed. Anyone can verify that no coins have been spent twice and that coins are legitimately owned. The Bitcoin protocol also uses a computational proof-of-work to control the creation of new Bitcoins. In a clever twist, this same scheme also makes the distributed ledger trustworthy and provides an incentive for participants to keep the ledger up-to-date (a process known as “bitcoin mining”).

Section 23.4.5 describes some potential weaknesses of Bitcoin, including the possibility of attacks on the blockchain technology along with political and economic threats. We also discuss alternative currencies that also use blockchain technology but vary in some of their design features.

23.1. The Role of Money in Economies

Trade is at the heart of all economies. One early form of trade was barter, where two interested parties directly exchanged goods or services for other goods or services. For example, one farmer might offer ten quarts of milk for two bundles of wheat. Barter markets are still in use in some places today. For example, students swap used textbooks with each other and people trade used furniture for other used furniture.

The big challenge with a barter market is the double coincidence of wants: for trade to occur, person A must want the goods of person B, and person B must want the goods of person A. Furthermore, this double coincidence of wants must occur at the same time and location. Another problem is that most goods are not easily divisible (anyone want to trade 1 sheep for 1.5 goat?) Last but not least, storing goods of value is difficult if the goods are non-durable such as wheat, or expensive to store over time, such as sheep that require constant feeding.

The use of money solves these problems. Money is transferable, meaning that when someone receives ten coins, she can use them to buy goods and services from another person. This solves the problem of the double coincidence of wants. Most importantly, money provides a standard form of value that can be understood by both parties to a trade. Money is also convenient; physical coins or bank notes, or credit cards, are easier to carry than sheep. Moreover, for all practical purposes, money is divisible, and except for countries with unstable governments and high inflation people generally accept that money will sustain its value into the future. When the value of a currency is relatively stable, then money can be used to store value over longer periods of time.

23.1.1. The Gold Standard

Until the 16th century, money typically took the form of metal coins, often gold. Around that time, paper money was slowly introduced, initially backed by debt instruments that would
23.1. The Role of Money in Economies

specify, for example, physical property that could be demanded in return for some amount of the paper money.

Later, it became common to adopt a monetary system referred to as the gold standard, with governments providing guarantees that paper money was backed by gold and redeemable for a specific amount of gold. Even though non-gold coins or paper money were used as currency, it was as if the currency was gold because a government promised to exchange this currency at a fixed rate for gold.

In 1944, most of the world’s industrial nations agreed upon the Bretton Woods System, an international monetary system that led to the establishment of the International Monetary Fund (IMF). Each participating country tied its exchange rate to the U.S. dollar, which was pegged to gold, and the U.S. government guaranteed that every U.S. dollar could be converted to gold at a fixed exchange rate, in effect putting the currencies of all other member countries on the gold standard.

Towards the end of the 1960s, however, the U.S. dollar started to be significantly overvalued, which created inflationary pressure. Many countries started exchanging large amounts of U.S. dollars for gold, or left the Bretton Woods System and allowed their currency to float in value. On August 15, 1971, the U.S. unilaterally ended the convertibility of U.S. dollars into gold, ending the Bretton Woods System and moving back to a system of floating exchange rates.

Today, no country is using the gold standard. The gold standard leads to inflexible monetary policies and governments have found this to be a challenge. Many governments increased the money supply during the economic downturn in the late 2000s, by printing more money. This would have required a large enough gold reserve for countries on the gold standard.

23.1.2. Fiat Money

All major currencies in use today are fiat money (fiat is Latin for “let it be done.”) Fiat money has no intrinsic value, and there is no guarantee provided by anyone that fiat currency will be worth anything. The value relies on trust in the entity that issues the currency, traditionally a government or central bank. By simple laws of supply and demand, if more money is printed then the value of each unit of currency in circulation falls, a process known as inflation. Governments may help to build the value of a currency by promising to accept the money as payment for certain services, by paying public workers in the currency, and by declaring it a legal medium of exchange.

It is typical that the central bank of a country is the only entity that can print money, and it is this bank that controls the amount of currency in circulation, thereby having some control over its value. If people lose trust that a currency will sustain its value, they may decide to exchange their money for durable goods or some other country’s currency.

Today, most monetary transactions in developed countries happen electronically. For example, 70% of all transactions between customers and businesses in the U.S. in 2011 happened electronically, up from 63% in 2001. Consumers pay with debit or credit cards, can use online services like PayPal to transfer funds, and wire services for large transactions that cross national borders. This ecosystem requires trust in third-party intermediaries such as banks. For example, if person A makes a wire transfer to person B, then A must trust her bank to deduct the...
proper amount of money and send it to $B$’s bank. Furthermore, $B$ must trust that it will be possible to access the transferred money. In this way, banks make sure that money is only spent by the rightful owner and only spent once. A similar trust relation must be established when using an electronic money service such as PayPal.

### 23.2. Digital Currency

Our interest is in the design of digital currency that does not rely on any centralized entity such as a government or a trusted intermediary such as a bank, and that is not backed by a commodity such as gold. In particular, we are interested in digital currency systems in which strings of bits take the role of coins. In this sense, a digital currency system operates quite differently from today’s electronic payment systems. Digital currencies are also sometimes called *e-cash* or *cryptocurrency*.

The primary benefits from a successful digital currency are:

- **Lower cost:** The internet has made it extremely cost-effective to transfer bits between people in different geographic locations and smart mobile devices made it easy to transfer bits in close physical proximity. This contrasts with the high fees for electronic transactions such as credit cards, debit cards, and wire transfers, all of which involve costly intermediaries.

- **Resilience:** Whereas systems that rely on intermediaries such as banks and credit card companies can be blocked by governments through legal rulings or pressure, a digital currency system that is fully decentralized and operates out of multiple countries would not be easy for a government to control. For example, it is illegal for banks to transfer money from a U.S. bank account to an online poker site, but no such limit could be imposed on payment through a digital currency. For example, Visa, MasterCard, and PayPal stopped making monetary transfers to Wikileaks from late 2010 to mid-2013, but payments in digital currency were not blocked.

- **Control of Money Supply:** A digital currency protocol can constrain by design the rate of creation of new money, and does not require trust in a central bank or government.

On the other hand, some potential vulnerabilities from a digital currency include:

- **Bugs, security problems, or unintended behaviors:** A user must trust that the digital currency system will remain operational and not be subject to bugs, crashes, attacks, or other kinds of outages. This extends not just to the core protocol but also to ancillary technology such as digital wallets and currency exchange.

- **Money printing:** When money is just strings of bits, it may become easy for anyone to create more money by creating new strings of bits. Counterfeiting is difficult with well-designed physical currency, such as metal coins and bank notes, and must be made similarly difficult for digital currency.
23.3. Credit Networks and the Trade of IOUs

- **Double spending:** A challenge with digital currency is to prevent a user spending money more than once, for example through copying, where the bits representing money are copied before the money is spent so that the money can be spent a second time.

- **Theft:** It may be easier to hack into a bank or exchange containing a digital currency and steal the bits that correspond to the money, or the right to transfer the money, than it is to break into a physical bank and steal physical money.

Over the last 20 years or more, researchers have proposed many different forms of digital currency, using a number of different technologies; e.g., cryptographic algorithms, trusted hardware that cannot be manipulated, credit networks, and proof-of-work. Bitcoin is the first digital currency to have gained wide-spread acceptance, and represents as of August 2016 more than 80% of the total market capitalization of digital currencies.

23.3. Credit Networks and the Trade of IOUs

A digital currency system based on IOUs can be useful in a setting with a common unit of accounting. In a peer-to-peer video streaming system for example, the unit of measure could be bandwidth. In a ride sharing network, the unit of measure could be distance (either consuming, as a passenger, or contributing, as a driver.)

Any user can issue any number of IOUs to other users. An IOU is a promise to provide the associated quantity of a good or service in the future. For now we assume that an IOUs is specific to a particular pair of users (e.g., “I will drive you for 5 miles”). But more generally an IOU could be transferable (e.g., “I will drive anyone in the network for 5 miles”), so that the IOUs begin to function similarly to a currency.

An unusual feature of IOUs relative to traditional money is that anyone can issue any quantity of IOUs. In response, others may choose to accept only a certain quantity of IOUs from a single user or may accept only IOUs from users whom they already trust. In this sense, every user can “print her own type of money” (e.g., Seuken dollars) and every user must decide how much trust to place on these various kinds of money. Consider for example a simple transaction, where user $B$ receives some goods from user $A$. In exchange, $B$ can give $A$ an IOU, which $A$ will only accept if she trusts $B$. Accepting this IOU corresponds to $A$ extending a certain amount of credit to user $B$.

A **credit network** is a weighted, directed graph that records the bilateral credit relationships between users. Each user is associated with a vertex, and a vertex is associated with a record of any IOUs that are held from other users and not redeemed or expired. A directed edge from user $B$ to user $A$ with weight, $w_{BA} > 0$, is a **credit edge** and indicates the maximum quantity of IOUs that $B$ can spend at $A$. Equivalently, this is the amount of credit $A$ is willing to extend to $B$.

An example is shown in Figure 23.1 (a): $B$ can pay an IOU to $A$ for up to three units and $D$ can pay an IOU to $B$ for up to two units. In the example, user $B$ may also **vouch** for user $D$ to user $A$, meaning that $B$ will cover any debt if $D$ does not redeem IOUs as promised. If users are willing to vouch for each other and receive IOUs over longer paths when facilitated.
Figure 23.1.: A transaction on a credit network in the presence of strong trust. (a) Initial credit network. (b) Flow of goods (double arrow), payment in IOUs (dashed arrow). (c) Final credit network.

by a trusted intermediary, we say the network exhibits strong trust. If this is not possible then we say that the network exhibits weak trust. With strong trust, the state of a credit network also records when one or more other users have vouched for an outstanding IOU.

**Example 23.1.** Consider the credit network in Figure 23.1 (a). In addition to the credit edges, we see that B is holding an IOU from A in the amount of 3 units. With only weak trust, C and A but not D can buy from A. With strong trust, B and C will vouch for D with A. B will vouch for up to 2 units and A will accept up to 3 units via B. C will vouch for up to 5 units and A will accept up to 4 units via C. The effect is that D can purchase up to 6 units.

Figure 23.1 (b) illustrates a purchase by D of 4 units from A with payment routed via C. Figure 23.1 (c) illustrates a possible state of the credit network after this transaction. A has an IOU from D, also indicating that C vouched for D. The weight on edges DC and CA have been decremented (CA is now not shown because its weight is now 0.) New credit edges have been added by agents to indicate willingness to reciprocate. D now provides 2 units of credit to C in return for C vouching for D, along with 3 units of credit to A. C also chooses to provide 1 unit of credit to A.

Credit networks do not overcome the problem of double coincidence of wants. In Example 23.1, both D and C need to offer something of value to A. We do not describe an exact way in which credit edges are updated after a transaction. This would likely be a result of norms, suggested policies, and self-interested behavior by participants.

**Example 23.2.** Figure 23.2 illustrates a transaction with two intermediary users. D buys 3 units from A, with C vouching to B for D and B vouching to A for C. The effect is that if D defaults, A will go to B to collect the debt, and will go to C to collect the debt. If C defaults then B should provide the back-stop to A. The credit edges after the transaction reflect...
one possible way in which users may choose to recipricate for the role of A, B and C in the transaction.

Whether strong or weak trust is a more appropriate model will depend on the application. A credit network that is built on top of a social network may exhibit strong trust: friends will lend each other money up to some limit, and vouch for the trustworthiness of friends up to some limit. A network built on very new business relationships may not exhibit strong trust: a bank who has only done one transaction with another bank may not be willing to vouch for the bank.

In a network with strong trust, the maximum quantity that B can purchase from A, for any pair of users A and B, is called the maximum flow on the credit network from B to A. In Example 23.1, the maximum flow from D to A is six units. Even if the trust weights on edges are small, large transactions can be enabled by considering all possible paths that connect a pair of users. In a practical setting, we might expect a limit on the length of paths in order to reduce risk. See the chapter notes for additional discussion.

23.3.1. The iOwe Digital Currency

The iOwe digital currency is designed to facilitate the trade of IOUs on a credit network. The unit of currency in iOwe is an iota, and an iota is a promise to repay a certain amount of debt in the future. We call this a “promise of work” but really it can be a promise of some quantity of any service or resource for which there is a standard unit of accounting.

The departure from IOUs is that an iota is not issued to a specific user. Rather, the promise is made to the first person who redeems the iota with the issuing user. This makes the IOUs transferable between users, giving them the flavor of a currency. Because an iota is backed by work, and thus something tangible that has intrinsic value, it is more like a gold-standard currency than a fiat currency.

The main design advantage of iOwe over traditional IOUs is that because iotas can be traded, it mitigates the double coincidence of wants without requiring strong trust. User B receiving an iota from A does not need to believe that A has something of value to B, only that someone
else may find this iota valuable. Rather than embodying the idea of “vouching” for other users, iOwe relies on direct credit edges between two parties transacting in an iota.

iOwe has been applied to peer-to-peer video streaming, where the common unit of accounting is bandwidth. Whereas the BitTorrent file sharing protocol described in Chapter 5 is inherently based on barter— a user needs to first acquire pieces of a file to “trade” before she can fully participate in a swarm —iOwe allows a user to immediately download a video from peers, spending credit that has accumulated from contributions of bandwidth when sharing another video.

iOwe uses public-key cryptography. Each user $i$ has a public key, $PK_i$, that is known to all other users, and a secret key, $SK_i$, only known to user $i$. For our purposes, it is enough to know that public-key cryptography provides the following properties:

- User $A$ can sign bits $X$ using her secret key $SK_A$, denoted as
  \[
  [X]_{SK_A}.
  \]  
  (23.1)
  This represents both $X$ and $A$’s signature of $X$. Any other user can use public key $PK_A$ to verify that this was signed by user $A$, and will believe that this could only have been signed by $A$ because only $A$ is assumed to have access to the secret key $SK_A$.

- User $A$ can associate bits $X$ with user $B$’s public key $PK_B$, forming the pair $(X, PK_B)$, and then sign this. We denote this as
  \[
  [X, PK_B]_{SK_A}.
  \]  
  (23.2)
  This represents both $X$ and $B$’s public key, as well as the signature by $A$ of the concatenation of $X$ with $PK_B$. User $B$ can later use her secret key $SK_B$ to prove that she is the user associated by $A$ with $X$, because only her secret key $SK_B$ corresponds to the public key $PK_B$.

Below, we describe the functional way in which creating iotas, spending iotas, and redeeming iotas occur. However, we will not discuss what a user would accept an iota as payment for. In an application, iOwe could be used together with a credit network to quantify the number of iotas users are willing to accept from each other.

**Creating iotas.** An iota for a promise of work of quantity $q$ from user $A$, with the latest time it can be used $expiry$-time, can be created by calling the function $issue_A(q, expiry$-time). This returns an *iota* of the form

\[
I = (A, q, expiry$-time, nonce),
\]  
(23.3)
where $nonce$ is a unique number associated with this iota that plays an important role in preventing double-spending attacks. This iota $I$ identifies the *issuing user* $A$, and states that whoever redeems the iota first, before it expires, should receive $q$ units of work from $A$. 

Copyright © 2017 Parkes & Seuken. Draft: Do not distribute without permission.
Spending iotas. Once created, the iota can be transferred to another user. If user A wants to pay user B, for example, then she will create the iota I and then use B’s public key PK_B to transfer I to B. By invoking function $\text{transfer}_A(I, PK_B)$ this transfers the iota to B, denoted as

$$I_{A\to B} = [I, PK_B, \text{time}]_{SK_A}, \quad (23.4)$$

where time is the current time. $I_{A\to B}$ denotes an iota that was issued by A and transferred to B. B can now redeem the iota in return for work from A. User B can also choose to transfer this iota to user C by invoking $\text{transfer}_B(I_{A\to B}, PK_C)$, denoted as

$$I_{A\to B\to C} = [I_{A\to B}, PK_C, \text{time}]_{SK_B}. \quad (23.5)$$

By transferring iotas from one user to the next, chains of signatures are created. These signature chains play an important role in preventing double-spending attacks.

Redeeming iotas. A user can redeem an iota by transferring it to the issuing user, and demanding the specified quantity of work. For example, user C can redeem the iota received from B by invoking function $\text{redeem}_C(I_{A\to B\to C}, PK_A)$. This transfers the iota back to A and requests work from A, the transfer denoted as

$$I_{A\to B\to C \to A} = [I_{A\to B\to C}, PK_A, \text{time}]_{SK_C}. \quad (23.6)$$

Anyone, including A and C can use the signature chain to verify that C was transferred the iota through a chain from A. In particular, PK_B along with B’s signature in the chain can be used to verify that the iota was transferred to C by B and PK_A along with A’s signature in the chain can be used to verify that the iota was transferred to B by A.

According to the protocol, A can refuse to provide the work when an iota is redeemed if the iota has already expired or has already been redeemed. Otherwise, user C should now receive quantity $q$ of work from A. If the iota has already been redeemed then this implies a double spending attack.

If A refuses to redeem an iota for any other reason, then this is a “step-omission attack” by A. We discuss attacks in Section 23.3.2.

Example 23.3. See Figure 23.3. First, users A and B transact. A receives 500 units of work from B, and creates an iota in the amount of 500 and transfers the iota to B. Second, users B and C transact. B receives 500 units of work from C and transfers the iota received from A to C. Third, users C and A transact. C receives 500 units of work from A and redeems the iota. At the end of this cycle of transactions, there are no iotas in circulation and every user has both received and contributed 500 units of work.
23. Digital Currencies

Figure 23.3.: An example of using the iOwe digital currency. $B$ works for $A$, then $C$ works for $B$, then $A$ works for $C$ (redeeming the iota earlier issued by $A$.)

23.3.2. Potential iOwe Vulnerabilities

What is to stop a double spending attack where $B$ spends an iota with $C$ but also redeems the iota with $A$? We explore various attacks in this section, and describe an approach to mitigate the power of the attack. We later review whether or not these responses are incentive aligned for users.

Whitewashing attacks. In a whitewashing attack on the iOwe currency, a user joins the system, issues some iotas, and then re-joins the system under a new identity. We have seen this kind of attack in the context of reputation systems (Chapter 21). In order to defend against this attack, the iOwe protocol suggests the following policy in regard to new entrants:

$P1$: No trust for new-entrants: A new user must build up trust through a series of successful barter transactions with an existing user. Only then will the existing user trust iotas issued by the new entrant.

In Example 23.3, user $A$ must have already established trust with both $B$ and $C$. This is because both $B$ and $C$ need to trust user $A$ to deliver on the promise. We might expect the amount of “$A$ iotas” that another user is willing to accept to increase as more trust is gained about $A$.

Double-spending Attacks. In a double-spending attack on the iOwe currency a user spends the same iota more than once.

Example 23.4. User $B$ executes a double-spending attack by spending the iota it received from $A$ through transfer $I_{A\rightarrow B}$ at both $C$ and $D$. This “forks” the signature chain, resulting in two signature chains ($A \rightarrow B \rightarrow C$) and ($A \rightarrow B \rightarrow D$).

In Example 23.4, the signature chains were both signed by $B$ and thus constitute a proof-of-misbehavior about user $B$. Alternatively, $B$ could simply try to spend the same iota twice with user $C$ (or redeem it twice with user $A$). In either case, the proof-of-misbehavior would
be established by exhibiting two signature chains, say \( I_{A\rightarrow B\rightarrow C} \) that are identical except for their time stamp when transferred from \( B \) to \( C \).

Any user who observes a forked signature chain or a duplicate signature chain can prove that \( B \) has committed a double-spending attack. In the worst case, the attack may go undetected except when both iotas are redeemed at user \( A \). For this purpose, user \( A \) stores all issued iotas until their expiry time. The expiry time of an iota helps to reduce the amount of overhead required in tracking iotas. In order to defend against a double-spending attack, the iOwe protocol suggests the following policy:

\( P2: \) Grim-trigger on Double-spenders: When user \( A \) detects a double-spending attack by user \( B \), then \( A \) never trusts \( B \) again and will not accept iotas transferred from \( B \) or created by \( B \). User \( A \) also forwards the proof-of-misbehavior to all of her trusted peers, who in turn forward it to their trusted peers, and so on.

This is the kind of sanctioning behavior we saw in the context of the repeated Prisoners’ Dilemma models of reputation systems, where defection is penalized through future defection by others (Chapter 21).

**Double-spending via a Sybil Attack.** A more sophisticated double-spending attack involves the creation of sybils, which are fake user identities under the control of a user. A sybil attack can work as follows:

1. User \( A \) earns the trust of other users in the network through barter exchanges.

2. User \( A \) creates a sybil \( S \), and creates and transfers an iota \( I \) to the sybil. The sybil now spends the same iota at two users \( B \) and \( C \) (who trust \( A \)), bringing in twice the quantity of work as would be the case were \( A \) to spend the iota only with \( B \).

This attack may later be caught, but it would be traced to sybil \( S \) and not \( A \). Later, \( A \) can create a new sybil, and repeat the process. To defend against this, the iOwe protocol suggests the following policy.

\( P3: \) Chain-of-trust: A user only accepts iotas if she trusts both the issuer of the iota and every user who has received the iota on the signature chain.

This means that user \( C \) in Example 23.3 should only accept the iota if she trusts both intermediary \( B \) and the issuing user \( A \). This limits the power of the attack because a sybil needs to bootstrap trust before it is able to spend an iota.

**Step-omission Attacks.** In a step-omission attack, the issuer of an iota refuses to redeem the iota even though the iota has not yet expired and has not been previously redeemed. To defend against this attack, the iOwe protocol suggests the following policy.

\( P4: \) Grim trigger on step-omitters: If user \( C \) detects a step-omission attack by user \( A \), then \( C \) no longer accepts any iotas created by user \( A \). User \( C \) also becomes a witness to \( A \)'s step omission and informs her trusted peers, who in turn inform their trusted peers, and so on.
23. Digital Currencies

For this, $C$ transfers the iota to a special public key $PK_0$, which is reserved to signal a step omission:

$$X = spend_C(I_{A\rightarrow\ldots\rightarrow C}, PK_0),$$

(23.7)

By sending $X$ to her trusted peers, $C$ also informs them that the original iota has been canceled and can not be redeemed at $A$. This does not constitute a proof of step omission, but rather a claim of step omission. This opens up the possibility of threats, where a user may threaten to unfairly block a participant in the network. In response, the iOwe protocol suggests the following policy in regard to how some other user, say $D$, should decide when to no longer trust $A$.

$P5$: Grim trigger on alleged step-omitters: User $D$ will no longer trust some other user $A$ after receiving at least a threshold $t_D > 1$ of claims about a step-omission attack by $A$.

Although this adds robustness to the iOwe protocol (and also against non malicious reasons for step omission such as a network outage or a user being on vacation), a group of users could still coordinate in order to block a particular user. A typical choice of the value of the threshold that has been used in experiments with iOwe is 10.

23.3.3. Challenges facing iOwe

iOwe requires neither trusted hardware or costly proof-of-work (which we will see are essential to the functioning of Bitcoin), and enables the transfer of IOUs and thus provides a primitive currency. A clever aspect of its design is that there is no guarantee that an iota is only “owned” by one user at any time. Rather, a user can both redeem an iota and spend it with another user, or spend an iota with two other users. iOwe is designed to overcome this apparent vulnerability through proof-of-misbehavior coupled with sanctioning mechanisms. However, the iOwe protocol does suffer from some important limitations.

Suggested policies (P1) and (P3) require pre-existing trust exist between users who trade iotas and between a receiving user and the original issuer of an iota. This limits the ability of iOwe to function as a currency, since iOwe does not fully overcome the problem of the double coincidence of wants. Rather, iOwe suggests users establish trust through barter. Moreover, iotas may vary in intrinsic value since they are “backed” by work from the issuing user and users are likely to be somewhat heterogeneous. Another concern is that iotas do not act as a store of value because they have an expiry time.

The need to bootstrap through reciprocal, barter trades also reduces iOwe’s ability to provide privacy of user transactions. Although identities can be pseudonyms and need not be tied with real world identities, a user cannot readily create multiple identities because of the need to bootstrap trust. This may lead to a compromise of privacy as data accumulates about transactions (see Chapter 30 for a detailed discussion about privacy attacks.)

There are also problems of incentive alignment. Although incentives are well aligned for catching and punishing a double-spending attack (P2), it is not clear why a user who detects a step-omission attack would choose to follow policy (P4) and become a witness to the attack. In our example, upon detecting a step omission attack by $A$, user $C$ is supposed to inform her
trusted peers about $A$'s step-omission. But this comes at a direct cost to $C$, since she could instead try to spend the iota with some other user who still trusts $A$.

### 23.4. The Bitcoin Digital Currency

Bitcoin is the most successful digital currency to date. We can see from Figure 23.4 that one Bitcoin (BTC) can be exchanged for hundreds of US dollars. Bitcoin is a fiat currency.

In Bitcoin, the problem of money printing is prevented by requiring a *proof-of-work* for the creation of new coins, which limits the rate at which new coins can be created. Anyone can participate as a “miner,” but mining coins is costly. The idea of a proof-of-work was developed in cryptography, and allows one user to convince another that she has completed some costly computation. A typical proof-of-work is to provide the answer to a hard problem that can only be solved, under standard complexity assumptions, by brute force search, but for which the correct answer is easy to check. In fact, the total amount of Bitcoins that can ever exist is also limited by the design of the Bitcoin protocol.

Similar to iOwe, Bitcoin uses public-key infrastructure to authorize the transfer of money. But unlike iOwe, it is not necessary to trust a user before accepting coins from the user. Instead, double spending is solved through the use of a *blockchain* that achieves consensus around a single history of all Bitcoin transactions. Nodes in the Bitcoin network can verify that new transactions don’t conflict with the ledger (for example, no coin is spent twice by the same user.) Transactions that do not conflict are included in new blocks, extending the blockchain. In this sense, coins are owned in Bitcoin in that a coin can only belong to one user at any point in time.

Creating a new block also causes new money to be printed. A valid block requires a proof-of-work, but in return the block includes a transaction that pays a reward of new coins. Because of this, the process of extending the ledger is referred to as “bitcoin mining” and the nodes in the network who do this work are “miners.”

The *Bitcoin network* is the peer-to-peer network that is used to broadcast new transactions and new blocks. Broadcast occurs through a *gossip protocol* where peers broadcast messages they receive with other peers with whom they have a connection.

There is no need for a central institution to issue Bitcoins or verify transactions. Even though Bitcoins are just strings of bits, the use of the shared ledger makes Bitcoin trustworthy. In particular, proof-of-work makes it costly to create new money or attack the blockchain data structure.

The protocol is not fixed but can be modified over time by agreement of a community of developers, and as such can be considered to be a “consensus protocol.” This protocol suggests a kind of equilibrium of a larger game, where nodes in the Bitcoin network follow the protocol because this is a good strategy given that the majority of others also follow the protocol.
23.4.1. Bitcoin Transactions

The unit of currency in Bitcoin is called a coin or Bitcoin, abbreviated BTC. Coins can be split and combined in arbitrary ways to form the quantity of coins that a user wants to transfer to another user (the smallest possible unit is called one Satoshi = $10^{-8}$ BTC, named after the pseudonym of its creator.)

A Bitcoin transaction involves the transfer of one coin from some user $A$ to some user $B$. More generally, a single Bitcoin transaction can involve the transfer of fractional coins to each of multiple different users. All transactions are recorded on the ledger so that the entire ownership history of a (fraction of a) coin can be checked, tracing all the way from the miner who created the coin to the current owner.

Bitcoin makes use of cryptographic hash functions. A hash function takes as input a string of bits, and produces as output a hash value of some fixed size, for example 1024 bits. Although it is possible that multiple inputs can map to the same hash value, with well-designed hash functions this happens very rarely. Hash functions are one-way functions, designed to make it computationally infeasible to find an input that hashes to a particular value. Thus, a hash value can be computed easily, but the function cannot be reversed.

One place in which hash functions are used by Bitcoin is to link together a sequence of transactions that involve (some fraction of) the same coin.

**Definition 23.1** (Bitcoin transaction). A Bitcoin transaction is the transfer of (a fraction of a) coin from one user to another, say $A$ to $B$. A transaction specifies:

- User $B$’s public key $PK_B$.
- The hash value of this public key and the transaction that transferred the coin to user $A$. 

---

Copyright © 2017 Parkes & Seuken. Draft: Do not distribute without permission.
23.4. The Bitcoin Digital Currency

Figure 23.5.: The signature chain corresponding to the transfer of a coin from $A$ to $B$ and then from $B$ to $C$.

- *The signature of this hash value, signed with the secret key $SK_A$ of user $A$.*

Figure 23.5 illustrates a sequence of transactions, where a new coin is transferred from $A$ to $B$ and then from $B$ to $C$. Each user may control multiple public keys, also referred to as Bitcoin addresses.

For user $A$ to transfer the coin to user $B$, the following steps take place:

1. User $A$ computes a hash value of user $B$’s public key and the transaction representing the new coin.

2. User $A$ uses her own secret key to sign this hash value. The transaction is defined by the public key of user $B$, the hash value, and the signed hash value. $A$ broadcasts this transaction over the Bitcoin network.

3. User $B$ receives the transaction over the network, and can verify that user $A$ has computed the hash value correctly, correctly included user $B$’s public key, and signed with the secret key corresponding to the public key that owns this coin (checking the ledger for this).

Crucially, if user $B$ can trust the correctness of the ledger entries about this coin, then she now believes that this transaction is being signed by the user who rightfully owns the coin.

*Copyright © 2017 Parkes & Seuken. Draft: Do not distribute without permission.*
23.4.2. The Blockchain

What is to stop a user from doublespending, where the same Bitcoin is used to pay two or more other users? This is the role of the blockchain, which ensures that all users come to adopt a single, consensus view of the history of all transactions. All transactions are written to the blockchain, and it is the blockchain that is used to establish who owns which coins, including the resolution of conflicts where someone tries to spend the same coin twice.

The Bitcoin protocol is designed so that on average a new block is added to the blockchain every ten minutes. The creation of a new block involves the computation of a new hash value. The input to the hash function is the block itself and the hash value of the previous block. See Figure 23.6.

The blockchain technology provides three important properties:

- Miners are incentivized to create new blocks. To achieve this, a new block includes a special generation transaction, which is used to create new Bitcoins.
- It makes it unlikely that multiple miners will create a new block at around the same time.
- A proof-of-work is required to create a new block, which makes it costly for a single entity to control the blockchain. This is achieved by requiring that the hash value of a new block must take on a special form.

The creation of a new block also prints a certain number of new coins as a side effect (as specified by the protocol, with a block only valid if the correct number are credited) to the miner. Figure 23.7 illustrates the items contained in a block. Only valid blocks can be used to extend the blockchain.

**Definition 23.2** (Valid block). A block is valid if it contains the following items:

- The hash value of a previous block.
- A generation transaction, that pays out some number of new Bitcoins.
23.4. The Bitcoin Digital Currency

Figure 23.7.: Each block contains a generation transaction, may contain new transactions that are consistent with the blockchain and self-consistent, the hash of the previous block, and a nonce value. Each block also has its own hash value. (Nakamoto, 2008)

- (optionally) One or more new transactions. These transactions must not conflict with themselves nor conflict with transactions in preceding blocks on the blockchain.
- A nonce value.
- A proof-of-work.

In addition, a valid block should not be larger than the size limit in the current consensus protocol.

The first transaction in a valid block is the generation transaction, transferring new Bitcoins to one or more users. As with iOwe, a nonce value is just a random string of bits. It plays a role in Bitcoin in proof-of-work.

The idea of non-conflicting transactions is illustrated by example.

Example 23.5. Consider a block with transactions $A \rightarrow B : 1.1$, $A \rightarrow C : 0.4$ and $D \rightarrow A : 0.2$. These transactions are valid as long as both $A$ and $D$ own enough BTCs based on the ledger. Let $x_{i,j,k} \geq 0$ denote the quantity of BTCs transferred from $i$ to $j$ in transaction $k$, and let $K$ denote the set of transactions on the blockchain before the current block. As a special case, $x_{0,j,k}$ is a generation transaction. For user $A$’s transactions to be non conflicting, we need

$$\sum_{k \in K} \sum_{i \neq A} x_{i,A,k} = \sum_{k \in K} \sum_{j \neq A} x_{A,j,k} \geq 1.3,$$

(23.8)
considering that $A$ pays $1.1 + 0.4$ but receives $0.2$ from $D$. For user $D$’s transactions to be non-conflicting, we need

$$\sum_{k \in K, i \neq D} x_{i,D,k} - \sum_{k \in K, j \neq D} x_{A,D,k} \geq 0.2.$$  \hspace{1cm} (23.9)

The order of transactions within a block does not matter. Rather, it is the order on transactions in different blocks that allows anyone to verify the rightful ownership of a coin. According to the semantics of Bitcoin, any transaction in block $A$ that is ahead of block $B$ in the chain has occurred before any transaction contained in block $B$.

It is incentive aligned for a Bitcoin miner to check that transactions are non-conflicting because the block will otherwise be rejected by other miners. At any time, each miner may know about a different set of transactions because of network latencies. This is fine, the only requirement is that the transactions in a valid block must be non-conflicting. Any transactions that are not included can be written into a subsequent block.

The hash value of a block provides the required proof-of-work. To be valid, the hash value must start with at least $\ell$ leading zeros. This can be achieved through a suitable nonce. $\ell$ is the difficulty level.

**Definition 23.3 (Bitcoin Proof-of-Work).** The Bitcoin proof-of-work requires that the hash value of a block must start with a sequence of at least $\ell$ leading zeros, where $\ell$ is the current difficulty level.

In particular, given a set of transactions $Txs$, a previous hash value $h$, and a nonce value $n$, let $B(Txs, h, n)$ denote the corresponding block and $H(B(Txs, h, n))$ denote its hash value. For difficulty level of $\ell = 17$, a valid block may have the following hash value:

$$00000000000000000134a957c59d8f4ab9b83f5850b2e1c34474d0e0ab4932ad$$

This hash value starts with 17 zeros. Hash functions provide a proof-of-work because they are one-way functions. This means that the computational puzzle of finding a suitable nonce can only be solved through “generate-and-test,” i.e., by guessing random nonce values and computing a hash value over and over again. Because generate-and-test is the best procedure, a miner suffers no set-back in computation by adding new transactions received over the network to a candidate block, and even though this effectively changes the puzzle.

With $\ell = 17$, fewer than one in $2^{17}$ attempted hashes will be successful, making this an effective proof-of-work. The Bitcoin protocol adjusts the difficulty level $\ell$ every 2016 blocks, such that on average a new block is generated by the mining community every ten minutes. In practice, a new block may be found in a few seconds or after tens of minutes, although the variance is relatively small due to the law of large numbers and because many miners are simultaneously working on finding a valid block. Increasing the difficulty level $\ell$ by a single bit leads to an exponential increase in time required to find a valid hash value.

In summary, the steps by which a new block is added to the blockchain are as follows:

1. New transactions are continuously shared over the Bitcoin network and if consistent with the current block chain can be incorporated into candidate blocks by miners.
23.4. The Bitcoin Digital Currency

2. Each miner works on finding a nonce that provides a hash of the block with at least \( \ell \) leading zeros, where \( \ell \) is the current difficulty level.

3. Upon finding a valid block, a miner will typically broadcast the block over the Bitcoin network. The first transaction in the block is a payment of Bitcoins to the miner.

4. If the block is valid, the Bitcoin protocol says that other miners should add the block to their local view of the blockchain and start working on extending this chain.

We defer discussion of the procedure for handling conflicting blocks to Section 23.4.4.

The complete blockchain is a large data structure that includes all transactions ever completed. The block chain was 80 GB in size in August 2016, and with an average block size of 0.8 MB (capped at 1MB) can be expected to grow by 40-50 GB per year. However, as earlier parts of the chain become trusted they can be archived and replaced with a snapshot of just the state of the ledger. As long as the archived ledger has been checked for correctness by enough miners this does not introduce any new vulnerability.

**23.4.3. Incentives for Bitcoin Mining**

A miner receives payment for finding a new block that becomes part of the blockchain. This payment consists of two parts:

1) Payment to the miner, through the generation transaction, of some number of Bitcoins that are created “out of thin air”, and

2) Transaction fees that may be specified as part of the transactions broadcast over the network, authorizing the miner to take a small fraction of the money in the transaction.

Figure 23.8.: The head of the Bitcoin blockchain as of 8.25am EDT on August 24, 2016 (blockexplorer.com)
Even though it has no sender, the generation transaction is accepted as a valid transaction by other miners because this creation of a Bitcoin is what the protocol specifies should happen when a new block is created. An originator of a transaction may include a transaction fee as an incentive to a miner to include a transaction in a new block.

Example 23.6. Figure 23.8 lists the blocks at the head of the current chain. If we look inside Block #426649 we see the following:

- Block hash: 00000000000000000134a957c59d8f4ab9b83f5850b2e1c34474d0e0ab4932ad
- Nonce value: 2928431880
- Mined by: Discus Fish
- First transaction: pay 12.76129312 BTC to 1KFHE7w8BhaENAswwryaoocDb6qcT6DbYY
- Second transaction: transfer 0.024211 BTC from address 1HtSEJrH2wBAufUXiZjAqrGWbybTuXj5 and 0.01018801 BTC from address 14pvojZewBZvc8DULmF1U3SVY4rfGisST5u to 0.01382878 BTC to address 1ZBwFLuHn3LEnhbtwG2Bw5D2q1CT1 and 0.01957023 BTC to address 1M8hxwYNuNhm8XUjF8Y829n49NXx3gA, with fee 0.001 BTC.

There are 1202 transactions in the block, and the block has size 0.71 MB. “Discus Fish” is the nickname used by F2Pool, which is a large Chinese mining pool. The pay out of 12.76129312 BTC in the generation transaction reflects 12.5 BTC for mining the block and a total of 0.26129312 BTC in fees. The second transaction is an example of a complex transaction that, in this case, takes payments from two addresses and divides up the money and sends some to one address and some to another.

The amount of Bitcoins awarded for the creation of a new block is halved every 210,000 blocks, which is approximately every 4 years. The second ever “halving day” occurred on July 9, 2016, with the mining of the 420,000th block, when the pay out was reduced from 25 BTCs to 12.5 BTCs (worth about $7,300 as of August 2016). The amount of currency in circulation will approach 21 million Bitcoins around 2140. Once that number is reached, no new Bitcoins will be created, and the total amount of currency in circulation will remain fixed.

Transaction fees will become more important as the payment in new coins decreases, and it may become necessary to include a fee as incentive for a miner to include a transaction in a block. Including a transaction in a block incurs a cost to the miner because it increases the size and thus latency of broadcasting the block over the network. It is possible for this additional latency to allow another miner’s block to be received by the network ahead of this miner’s block.

23.4.4. Maintaining a Consensus on a History of Transactions

The way in which Bitcoin achieves synchronization, so that everyone who agrees to agree on a single history of transactions, is at the core of the design of the protocol.
A conflict can arise when two miners find different, valid blocks in quick succession. When this happens, some miners will hear about one block first and some miners will hear about the other block first. This causes the blockchain to fork, with different miners working to extend both the chains. In general, these chains can contain different transactions. For example, if there are two conflicting transactions broadcast over the network such as “transfer A’s Bitcoin to B” and “transfer A’s Bitcoin to C” then each miner can decide whether to include the first transaction, the second transaction, or neither in a block. The Bitcoin protocol achieves consensus through two rules that govern block creation and block adoption:

1. **Block creation is difficult (by design)**, which means that new blocks are only created on average once every 10 minutes. Because of this, a new block will tend to be propagated over the network before the next one is produced and conflicts will be rare. Thus, a subsequent block will reference this block and not include conflicting transactions.

2. **Adopt the longest chain.** The protocol specifies that miners should accept the longer of multiple blockchains when presented with a choice. This means that the next time a valid block is created, say for the first chain, then this chain will become the longest and all miners will switch to work on this chain. On a very rare occasion there may be another tie, but this will soon be resolved.

Transactions on the dropped chain may no longer be part of the history, and if any are in conflict with those in the surviving chain then they cannot go into an extension of the surviving chain.

The upshot of this is that in the event of a forked blockchain, one chain will win the race and be adopted by all. Thus, nodes in the Bitcoin network maintain consensus on a single blockchain, and maintain agreement about a single order on transactions.

### 23.4.5. Potential Bitcoin Vulnerabilities

Bitcoin has become the most successful digital currency of all time since its launch in 2009. Still it has some potential vulnerabilities.

**Double-Spending Attack.** Bitcoin is designed to make a double-spending attack require a lot of computational power. We can think of each block that contains the transaction or is added after this block as providing a confirmation of the transaction, supporting its inclusion on the ledger. Assuming that merchants wait for 1-confirmation, it is unlikely for an attacker to succeed with a double-spending attack without controlling more than 50% of the total computational power available in the network.

Suppose that user A buys some goods from a merchant (user B), and that B waits for a 1-confirmation before shipping the goods. In a double-spending attack, user A also spends the same coins with user C, and tries to fork the blockchain so that the second transaction and not the first is ultimately in the consensus blockchain. See Figure 23.9. Following the creation of block t, user A needs to generate two blocks (containing C’s but not B’s transaction) before the rest of the network creates a block containing the B transaction and a second block. If
A double-spending attack on Bitcoin. User A needs to create two blocks containing the transaction with C before the rest of the network creates two blocks containing the transaction with B.

If A succeeds, then she can broadcast these two blocks (at some point after the B block has been created) and the chain including the C transaction will become the consensus chain. But even with 49% of the total computational power of the mining network, this would occur with probability at most $(0.49)(0.49) \approx 0.24$.

Also, a merchant can gain additional protection by insisting on more than one confirmation. With every additional confirmation, it becomes exponentially harder for the attacker to overtake the blockchain. In practice, some merchants choose to accept transactions with 0-confirmation as long as they have been broadcast through the network, trusting they will later be included in the blockchain. Thus far very few of these 0-confirmation transactions have become compromised.

A sybil attack is not useful in Bitcoin because sybils do not increase the computational power of the attacker, and as long as honest nodes can still communicate with each other (so that transactions are reliably received and added to new blocks by honest miners.) Conceptually, the community of honest miners is continuously voting on which is the correct chronological order of transactions, not with “one vote per user,” but with “one vote per CPU.” Any computational resources available to a sybil could just as usefully be used by the attacker directly.

**Concealed Mining Strategy.** With the value of new blocks worth US $10,000 or more, a considerable amount of computational power is being used to mine blocks. Measured as the hash rate of the network, this can be estimated from the difficulty level and from the rate at which new blocks are mined. In turn, mining has become concentrated around a small number of Bitcoin mining collectives or pools. Mining pools invest in specialized hardware to compute hash functions (see the chapter notes.) Today, an individual with a standard computer has almost no chance of creating a new block without joining a mining collective (where costs and
23.4. The Bitcoin Digital Currency

benefits are pooled.)

A large mining collective can gain an advantage over individual miners through the *concealed mining strategy*. A particular form of this attack is the “51% attack”, and assumes the collective has more than 50% of network computational power. But it is not necessary to have this much power. The concealed mining strategy works as follows. Upon any one of the collective finding a new block $B_1$ before other miners, the collective conceals this block and starts working (privately) to also find a successor block $B_2$. When a competing block $B'_1$ is announced by other miners, the collective will then quickly broadcast $B_1$. As long as they are well connected and a large enough group, this block $B_1$ can reach the majority of the network before block $B'_1$. By forming the consensus blockchain, the collective can still receive a payment for the mined bitcoins. In addition, by mining for $B_2$ while $B_1$ was concealed the collective had a monopoly on finding $B_2$ and thus an advantage over the rest of the system. It is possible, for example, that they have found $B_2$ already, in which case they can broadcast $B_1$ and $B_2$ in quick succession.

That collectives can gain more value than is suggested by their share of computational resources tends to lead to further centralization of Bitcoin mining. This is a potential concern because a large collective could threaten to destabilize the currency by creating forks or controlling the transactions that can be added to the ledger.

**Hiding Transactions.** The peer-to-peer Bitcoin network relies on simple communication protocols for spreading information about new transactions and blocks on the network (called *gossiping protocols*.) A potential concern is that it might be in a miner’s interest to deviate from the protocol and not share information about transactions, instead retaining a monopoly on transactions and gaining the associated fees in her next successful block. If this pattern should develop, then a possible response is to modify the protocol to also reward nodes who communicate information about a transaction.

**Price Instability and political threats.** The value of Bitcoin has fluctuated enormously, even by as much as 50% in one day. Events that have triggered this include the hacking of a currency exchange, or the change in a law in regard to the legal status of Bitcoin. This has made Bitcoin less useful as a medium for storing value, even if it is effective as a mechanism for the transfer of value. For example, a user might convert USD to BTC, transfer BTCs to another user, who then converts BTCs to GBP.

Governments are increasingly taking a stance on the legal status of Bitcoin as it gains traction as a digital currency. It is conceivable that some governments may simply declare Bitcoin to be illegal, for example because it is perceived as competing with the government’s own currency or because of concerns about international transfers or money laundering. As of 2016, this worst-case scenario has not happened. In the U.S., Bitcoin has been declared to have the legal status of property rather than currency, which has allowed it to circumvent further regulation. The Chinese government did declare Bitcoin not to be a currency in December 2013, and banned financial institutions from handling Bitcoin transactions or trading in Bitcoins, which may have contributed to the rapid devaluation of Bitcoin in early 2014 that followed.
a speculative run-up in 2013. However private parties can still trade and hold Bitcoins in China, and 80% of trading volume on Bitcoin exchanges is reported to be on the Yuan-Bitcoin currency pair.

Beyond concerns about the effect of security vulnerabilities or new legal frameworks, there are some concerns that the design of the protocol may affect price stability. The total amount of Bitcoins that will ever be printed is fixed at 21 million coins; although Bitcoin is a fiat currency, it shares with gold-backed currencies the properties that it is costly to print, and has a finite amount that can be printed. Even at a price of $1,000 per Bitcoin, the value of all Bitcoins combined would only be 21 billion USD. In comparison, the GDP of the U.S. was almost USD $18 trillion in 2015. Thus, for a significant amount of economic trade to happen through Bitcoin, the currency will need to significantly appreciate in value. This currency deflation is a concern because it can make people reluctant to spend a currency, preferring instead to hoard money. This in turn can drive up the price of Bitcoins, until they are so valuable that there is a crash with many people selling Bitcoins quickly.

Disagreement in the Bitcoin Community. The design of digital currency protocols such as Bitcoin is inherently flexible. Because there is no centralized institution running the protocol, a working group of developers makes ongoing decisions about changes to the consensus protocol. An active debate in recent years, for example, has been about the cap on the size of a block (1 MB as of August 2016). This cap helps to keep out very small transactions and restrict the rate at which the size of the blockchain can grow. But at the same time, it limits the number of transactions that can be processed. An undesirable outcome of disagreement would be that the blockchain permanently forks into two or more chains. This could hurt the viability of Bitcoin because each chain would in effect become its own currency, with different values.

Security Vulnerabilities. While Bitcoin’s core protocol and its network has not been successfully compromised to date, the ecosystem of services that support the storage of bitcoins (“wallets”), the exchange of bitcoins into cash (“exchanges”), and provide markets for goods and services (“marketplaces”) have been less secure and suffered numerous theft to hackers or insiders. See Table 23.1. The most notorious attack to date was on the Mt Gox exchange in February 2014. These attacks threaten trust in Bitcoin.

Because Bitcoins are just bits, the theft of coins is a matter of hacking and issuing transfers of coins that can then be exchanged for cash, goods or services. One kind of vulnerability arises when a user’s private key is compromised. With access to a private key, which may be stored for example on a user’s computer drive, a third party can simply transfer all coins in the associated Bitcoin account to other addresses. One way to gain access to private keys is by compromising a site that stores keys on behalf of multiple users, as may be the case with a wallet site or exchange.

Moreover, unlike credit card transactions, Bitcoin transactions are irreversible because Bitcoin operates like cash. Even the ability to track transactions is not very useful because mixing services can be used to mix together clean and dirty coins (i.e., coins linked to an attack), combining them and then splitting them out into new addresses.
23.4. The Bitcoin Digital Currency

<table>
<thead>
<tr>
<th>Date</th>
<th>Site</th>
<th>Theft amount (BTCs)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2011</td>
<td>User: Allinvain</td>
<td>25,000</td>
<td>wallet account attack</td>
</tr>
<tr>
<td>March, May 2012</td>
<td>Bitcoinnica</td>
<td>61,500</td>
<td>exchange</td>
</tr>
<tr>
<td>Sep 2012</td>
<td>Bitfloor</td>
<td>24,000</td>
<td>exchange</td>
</tr>
<tr>
<td>Oct 2013 (twice)</td>
<td>inputs.io</td>
<td>4,100</td>
<td>wallet</td>
</tr>
<tr>
<td>Dec 2013</td>
<td>Sheep</td>
<td>5,400</td>
<td>marketplace</td>
</tr>
<tr>
<td>Feb 2014</td>
<td>Mt Gox</td>
<td>~ 850,000</td>
<td>exchange</td>
</tr>
<tr>
<td>Feb 2014</td>
<td>Silk Road 2</td>
<td>~ 4,300</td>
<td>marketplace</td>
</tr>
<tr>
<td>Jan 2015</td>
<td>Bitstamp</td>
<td>~ 19,000</td>
<td>exchange</td>
</tr>
<tr>
<td>Aug 2016</td>
<td>Bitfinex</td>
<td>~ 120,000</td>
<td>exchange</td>
</tr>
</tbody>
</table>

Table 23.1.: Example attacks on the Bitcoin infrastructure including wallets and exchanges.

The chapter notes provide further discussion and references for all of these.

23.4.6. Bitcoin and other Cryptocurrencies

Bitcoin has proved to be a successful digital currency. Some of the promised benefits of digital currencies are:

- Lower cost. As of August 2016, the number of new BTCs minted each day is $6 \times 24 \times 12.5$, and at USD $585 per coin. Assuming efficient mining and network operation, this implies total computational work of at most $1$ million per day. Compared against an average transaction rate of 225,000 per day, and an average daily trade volume of USD $125$ million, this is $5$ per transaction but only 0.8% of the transaction volume. This compares favorably to the 3% and upwards charged by banks and credit card companies for electronic transactions.

- Privacy. Even though each transaction is written to the ledger, transactions are only linked to pseudonyms and not linked with real-world identities. A user can create any number of public-private keys and thus Bitcoin addresses, and can also make use of mixing services to hide the flow of particular coins. This makes it hard to track Bitcoins.

- Resilience. No government or third-party attack has shut-down Bitcoin or been able to prevent payments being made across international borders using Bitcoins.

- Control of money supply. According to the Bitcoin protocol, the total number of coins that will ever be printed is 21 million coins.

Still, there are some real concerns about Bitcoin in addition to those discussed in the previous section. Beyond the cap of 1 MB on block size, these include the 10 minute latency for confirming a transaction, that the network latency is quite high, and that the proof-of-work is best solved with specialized hardware that promotes the concentration of mining power.
A number of alternative digital currencies ("altcurrencies") have been introduced in recent years, both as a response to concerns and as a way to make money. Table 23.2 lists four popular digital currencies as of August 2016. All four currencies use their own blockchains, with incentives aligned with proof-of-work through their own fiat currencies.

These currencies offer different tradeoffs in the design space. They vary, for example, in whether the total amount of the currency is capped or not. The three more recent currencies use proof-of-work that requires a large amount of memory in order to provide a higher barrier to the use of specialized hardware (which becomes more costly when more memory is required.) The designs adopt increasingly faster confirmation times. This does not mean that transactions can be completed more quickly however. Rather, the confidence in a transaction depends on the amount of computational work performed in confirmations.

Ethereum allows for flexible computation to be performed on its blockchain. Whereas the nodes in the Bitcoin network check that transactions are non-conflicting, nodes in the Ethereum network can be programmed to step through a scripting language. Ethereum has been called a “World computer” since it is fully decentralized and can run programs. For example, Ethereum allows the programming of smart contracts where Ether is transferred contingent on a signal being written to the blockchain ("pay 0.1 Ether to Mum when Dad says I used her car.") In order to compensate nodes for the computation and storage required to correctly execute the scripts, different instructions are associated with different amounts of payment according to a payment schedule. These payments are referred to as "gas" in an analogy to providing the fuel to run the computer.

A concern with proof-of-work for providing trust is that this consumes a large amount of power and negatively impacts the environment. In January 2016, for example, the Bitcoin network computed hashes at a rate of 800 petahashes per sec (1 peta = 10^15), requiring around 340 MW (1 mega = 10^6) of power. By 2020, the hash rate could be as high as 40,000 petahashes...
per sec (with estimates varying by orders of magnitude), which would require 1,600 MW of power consumption at an estimated average efficiency of 0.04 J per gigahash (1 giga = $10^9$). For context, this is the equivalent of the average power consumption of 1.3 million American households (using 2014 consumption numbers), but just a tiny fraction of total US power consumption (less than 0.0003% as of 2014).

See the chapter notes for further discussion if these alt currencies.

23.5. Chapter Notes

The iOwe digital currency protocol is due to Levin et al. (2011), and much of our treatment is based on the discussion in their paper. Dandekar et al. (2011) provide an economic model of a credit network, analyzing the probability that a network can sustain transactions between users and the effect of topology on the resilience of a network. Dandekar et al. (2015) complement this study with an analysis of the strategic formation of credit networks. A real-world implementation of the idea of credit networks is provided by Ripple, which has at its core an IOU-based network for moving liabilities through a chain of trusted intermediaries; see https:\\ripple.com. Ripple is a decentralized system with a non-proof-of-work approach to consensus, and is used as a settlement infrastructure by banks. Ripple also has its own XRP (“ripple”) digital currency for transfers, this currency having a market capitalization of US $214m as of August 2016 (rank #3). Seuken and Parkes (2014) study the trustworthiness of voluntary reports of contributions in an IOU-like setting.

The origins of the Bitcoin protocol are still a mystery, until today. Using a pseudonym, Nakamoto (2008) published the original white paper describing the design of Bitcoin. The open source community developed software following publication of this whitepaper, and the first block was mined on January 3, 2009, this “genesis block” printing 50 BTC. The block included (coded in hex) the following text:

“The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”

Beyond a proof that this block was created on or after this date, this may also be intended to provide a comment on concerns about governments’ ability to print currency. A good source of empirical data on Bitcoin is https:\\blockchain.info, including charts on hash rate, number of transactions, trade volume and the exchange rate. This is a rapidly developing area. Wikipedia and the Bitcoin Wiki https://bitcoin.it (maintained by the open source community) continue to provide good, up-to-date information about its evolving design along with information about topics such as mixing services, theft and security concerns, and specialized mining hardware.

Zohar (2015) provides an excellent, technical overview of Bitcoin, and is the source of some of our discussion, especially in regard to the way in which Bitcoin ensures consensus on a single history of transactions. As Zohar points out, the Bitcoin protocol is not actually to pick the longest chain to break a tie, but the chain with the highest aggregate difficulty of proof-of-work computations. But it is simpler to think about the length of the chain as this aggregate measure of effort. Zohar reports that while relatively few attacks on 0-confirmation transactions have
taken place, they still pose a risk. Böhme et al. (2015) provide a nice discussion of Bitcoin from an economic (including consumer) and regulatory perspective. Other economists have also provided their viewpoints on Bitcoin is (Krugman, 2011; Surowiecki, 2011).

The idea of proof-of-work was introduced in Dwork and Naor (1993) and the concept was formalized and coined “proof-of-work” by Jakobsson and Juels (1999). Digital currency that uses cryptography to secure transactions and control the creation of currency is often referred to as a cryptocurrency. Bitcoin proof-of-work is via the SHA-256 hash function (Schneier, 1996). Bitcoin mining leverages specialized mining hardware, first GPUs and today ASICs (Application Specific Integrated Circuits). An important performance metric is the energy efficiency, with the best hardware as of summer 2016 operating at 0.1 J per gigahash.

The concern about the incentive to hide transactions in Bitcoin, as well as a suggested solution to this problem, are described by Babaioff et al. (2012). A simplified, non-technical summary is also available (Babaioff et al., 2011). The idea is to reward not only the miner who succeeds with creating a new block, but also all miners on the path from the originator of the transaction to the miner who creates the block. Effectively, the transaction fee could be split up between all users along that path. The concealed mining strategy is elaborated on by Eyal and Sirer (2014), who further explain how it is possible for a collective with more than 1/3 of the total mining power to collect rewards that exceed its proportion of mining power. Kash et al. (2015) presents an elegant model to explain the way in which crashes can occur with fiat currency (“scrip” in the language of their paper). A well-known real-world example of a scrip system that crashed is the Capital Hill babysitting co-op, as described in Krugman (1998).

A June 2016 New York Times report “How China Took Center Stage in Bitcoin’s Civil War” by N. Popper states that “over 70% of the transactions on the Bitcoin network were going through just four Chinese companies” with most flowing through two of these companies. Concerns that the need for specialized hardware promotes concentration of mining power promoted the adoption of memory-hard proof-of-work such as scrypt (pronounced “ess crypt”) and Ethash in more recent blockchains. These hash functions are designed to require a large amount of memory to run quickly, so that ASICs become costly (requiring a large amount of cache memory per core.) Still, ASICs were available for scrypt since early 2014, but not for Ethash as of August 2016. The statistic on the volume of Yuan-BTC exchange is from a Goldman Sachs report published in March 2015. There are likely a number of contributing factors to the Bitcoin crash in early 2014, including the speculative build-up in value in 2013, new Chinese regulation, as well as the Mt Gox collapse. The data on market capitalization of various digital currencies is from coinmarketcap.com.

Ethereum was proposed by Vitalik Buterin in late 2013 and launched in July 2015, with a goal of building decentralized applications and described as a “next-generation cryptocurrency and decentralized application platform.” There is considerable interest in the possibility of building decentralized, autonomous economic entities on top of Ethereum’s blockchain, with applications for example to personal finance, crowdfunding, and securities trading. Ethereum uses gas to compensate nodes for work in implementing transactions, with the blockchain-based platform providing a decentralized virtual machine, the Ethereum virtual machine. As of August 2016, one unit of gas is 10^{-5} Ether (or 10 Szabo, which is 10^{-6} Ether.) The point of these payments is to make users pay for resources used while also stopping infinite loops.
and denial of service attacks. For example, a transfer of Ether is 1 gas, a SHA-3 computation is 20 gas, addition is 3 gas and multiplication is 5 gas. A good resource for learning more is https://www.ethereum.org.

Ethereum can be used to build distributed anonymous organization, essentially firms with rules of operation that are programmed via smart contracts. This requires some care to get right. A cautionary tale is the crowd-funded investment vehicle “The DAO,” that launched April 30, 2016. In the first stage, people could invest Ether in return for DAO tokens. 11.5m Ether were invested by 11,000 users by the end of May, representing 16% of the total supply of Ether and worth US $150 m at the time. The DAO’s program would give investors the right to vote on projects and a share of profits according to the number of tokens held. On June 17, 2016 there was an attack on The DAO, with 3.6m Ether transferred to a “child DAO” that was controlled by the attacker. The attack exploited a loophole in the DAO code that allowed DAO tokens to be copied over and over again, and followed earlier concerns (Mark et al., 2016).

By the rules of The DAO, the child DAO could not pay out Ether for a period of in excess of four weeks, giving the Ethereum community time to debate how to respond. Ultimately, a vote conducted in July found around 80% in favor of a hard fork, that would move (through instructions that would ordinarily be irregular and not approved) the 11.5m Ether into a new smart contract that would refund initial investors in DAO tokens. This hard fork was performed in block 1,920,000 on July 20, 2016. Interestingly, the community was split and some miners continued on the old chain and the currency forked into two. As of August 2016, Ethereum Classic is the original blockchain and Ethereum is the hard-forked chain, with market capitalizations as of August 2016 of US $136m and $929 m respectively (#6 and #2 ranked.)

Litecoin was released in October 2011 by Charles Lee, and is inspired by and nearly identical to the Bitcoin protocol (changes including a smaller block generation time, an increased maximum number of coins, and the scrypt rather than SHA-256 hashing algorithm.) Dogecoin was released in December 2013 by Billy Markus, based on Luckycoin that was in turn based on Litecoin. Dogecoin features a likeness of the Shiba Inu dog from the “Doge” Internet meme, popular in 2013, as its logo.

The estimates of the energy usage of the Bitcoin mining network are adapted from the analysis published by Deetman (2016). The projected network hashrate of 40,000 petahashes per sec by January 2020 extrapolates growth from 810 to 1550 petahashes per sec from January 2016 to September 2016, and is larger than but close to Deetman’s lower estimate (and orders of magnitude smaller than his pessimistic estimate.) The average efficiency of Bitcoin mining as of January 2016 was estimated by Deetman to be 0.42 J per gigahash. The anticipated efficiency of 0.04 J per gigahash is more optimistic than Deetman’s estimate, but reflects the surprising efficiency of new ASICs (0.1 J per gigahash, relative to Deetman’s best estimates of 0.2 J per gigahash for September 2016.)

Motivated in part by concerns about energy cost of proof-of-work, some crypto-currencies require proof-of-stake for a valid block. In proof-of-stake, users are asked to prove ownership of a certain amount of currency (the “stake”) in order to mine a new block. This is done without allowing the user with the largest balance to gain control. The idea of proof-of-stake seems to date back to a 2011 post to Bitcoin forum https://bitcoin.org by QuantumMechanic; see https://en.bitcoin.it/wiki/Proof_of_Stake. Peercoin, launched in 2012, was the first
23. Digital Currencies

cryptocurrency to launch using proof-of-stake. Nxt, launched in 2013, is the most successful, proof-of-work based currency as of summer 2016, with a market cap of US $24 m (rank #14). As of summer 2016 there is discussion but no agreement as to whether Ethereum should adopt proof-of-stake.

23.6. Comprehension Questions and Exercises

23.6.1. Comprehension Questions

c22.1 State two differences between a gold-backed currency and a fiat currency.

c22.2 State two problems with the iOwe protocol.

c22.3 What is the most important property provided by the use of the blockchain in Bitcoin?

c22.4 What do you think is Bitcoin’s biggest weakness?

c22.5 Would you like to start using Bitcoins to do your online shopping? Explain!

c22.6 Why does a digital currency such as Dogecoin have value?

23.6.2. Exercises

22.1 Credit networks

(a) Give an example of a setting where you would expect weak trust and a setting where you would expect strong trust.

(b) What application can you think of for a credit network (provide a different one than those suggested in the chapter)? Justify why this is an plausible application.

(c) How might you imagine solving the problem that different IOUs may have different units of value?

22.2 Bitcoin: Empirical

[This assignment assumes Bitcoin is still in operation.] If so, you will need to visit http://blockexplorer.com/ and https://blockchain.info/, or similar sites.

(a) What is the current BTC to USD exchange rate?

(b) What was the average number of blocks generated per hour in the list of latest blocks at http://blockexplorer.com/? How does this relate to the target of the Bitcoin protocol?

(c) Plot a histogram of the distribution of the total transaction value in USD of the new blocks generated in the view at https://blockchain.info/. Give a brief discussion on what you find.
23.6. Comprehension Questions and Exercises

(d) Click on a block ID and look at the transactions. How many transactions are there in total in the block? What is the average USD value per transaction in the block? Provide a brief, qualitative description of the first three transactions.

(e) Visit \url{http://blockchain.info/charts/hash-rate}. What is meant by the “hash rate” of the bitcoin network, and how can you explain the trend in hash rate over the past year? How does the hashrate compare with the total power of worlds top 500 supercomputers?

(f) Look here \url{https://en.bitcoin.it/wiki/Mining_hardware_comparison#ASIC} (or similar) to find the best Mhash/s/\$ rate amongst ASICs currently shipping. How much would you need to spend on hardware to achieve the current network-wide hash rate?

(g) If your university has a compute cluster, and by reference to similar machines listed at \url{https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison} (or similar), determine the total gigahashes per sec achievable on your school’s cluster. How does this compare to the current hash rate of Bitcoin? Do you think you could make money by harnessing all of these machines? [Please don’t try!]

22.3 Cryptocurrency protocols

(a) Explain briefly what causes a new block to be formed in Bitcoin, why there are “bitcoin miners,” and how double spending is prevented in Bitcoin?

(b) Investigate the details of the Bitcoin protocol rule by which the difficult level is adjusted, and explain them in your own words. Be sure to cite any sources.

(c) Give two concrete but simple examples of what can be implemented by writing code that runs on Ethereum. Be as specific as you can, and cite any sources.

(d) Proof-of-stake is suggested as an alternate approach to proof-of-work. Conduct some research and: (i) provide a paragraph to explain, as concretely as possible, one specific way to define a proof-of-stake approach to block mining, (ii) explain a vulnerability that people are concerned about in regard to proof-of-stake. Be sure to cite any sources.

(e) What is a real-world application that has been deployed using Ethereum or a distributed application platform in the spirit of Ethereum? Give a brief description, citing any sources.

22.4 Keeping up-to-date

Give your sources throughout.

(a) Provide a plot of the historical rate for BTC vs USD since late 2011 to the present day.

(b) If you can find any, list three attacks on Bitcoin infrastructure that have occurred since the August 2016 Bitfinex attack. Give the amount of BTCs involved and the
kind of service attacked. If you don’t find any on Bitcoin then list three attacks (if any) on other digital currencies.

(c) Provide a plot of the historical exchange rate for Ether vs USD from launch up to the present day. If Ethereum is still operating, what is the current price of gas?

(d) List any proof-of-stake currencies that are in the top 10 digital currencies by market capitalization. List the current market capitalization for the four currencies in Table 23.2 where they are still operational.

(e) If still operational, then bring the energy calculations for Bitcoin up-to-date. What is the current hash rate, what is the current estimated average energy efficiency, and what do we expect to see in terms of total power consumption in January 2020 (use the actual data if Jan 2020 has already passed). Make your own projection of energy use by Bitcoin miners four years from now. State any assumptions that you make and be sure to cite any sources that you use to guide your analysis.