A Tour of Emerging Cryptographic Technologies

What They Are and How They Could Matter

May 2021

Ben Garfinkel

Centre for the Governance of AI,
Future of Humanity Institute,
University of Oxford

Abstract

Historically, progress in the field of cryptography has been enormously consequential. Over the past century, for instance, cryptographic discoveries have played a key role in a world war and made it possible to use the internet for business and private communication. In the interest of exploring the impact the field may have in the future, I consider a suite of more recent developments. My primary focus is on blockchain-based technologies (such as cryptocurrencies) and on techniques for computing on confidential data (such as secure multiparty computation). I provide an introduction to these technologies that assumes no mathematical background or previous knowledge of cryptography. Then, I consider several speculative predictions that some researchers and engineers have made about the technologies' long-term political significance. This includes predictions that more "privacy-preserving" forms of surveillance will become possible, that the roles of centralized institutions ranging from banks to voting authorities will shrink, and that new transnational institutions known as "decentralized autonomous organizations" will emerge. Finally, I close by discussing some challenges that are likely to limit the significance of emerging cryptographic technologies. On the basis of these challenges, it is premature to predict that any of them will approach the transformativeness of previous technologies. However, this remains a rapidly developing area well worth following.

Report background

This report was written as a follow-up to a 2016 workshop at the Future of Humanity Institute, exploring the implications of blockchain technology. The workshop participants were Stuart Armstrong, Shahar Avin, Nick Bostrom, Miles Brundage, Vitalik Buterin, Jeff Coleman, Owen Cotton-Barratt, Wei Dai, Owain Evans, Virgil Griffith, Georgios Piliouras, Anders Sandberg, and Vlad Zamfir. I later expanded the report by drawing on sources ranging from academic journal articles to blog posts to technical whitepapers to informal conversations with researchers and engineers in this space. The goal was to write something that would allow complete non-experts (like myself when I began the project) to quickly understand what these new technologies are and why so many people are excited or worried about them.

Most of this report's text dates back to 2017. Fortunately, because the report's focus is on fundamental principles and limitations, I have only felt the need to make modest revisions in the intervening years. At least as of Spring 2021, I believe it can still be read as an accurate description of the state of field.[1]

[1]Thank you to Jeff Coleman, Rhys Lindmark, Jaan Tallinn, Morten Dahl, Andrew Trask, Allan Dafoe, Jan Leike, Anish Mohammed, Luke Muehlhauser, Carrick Flynn, Pablo Stafforini, Nick Brown, and Oge Nnadi for detailed comments on previous drafts of this report. Thank you as well to Anne le Roux, Laura Pomarius, and Justis Mills for help with the preparation of the report.

Contents

1 Introduction

The past century saw the emergence of a number of transformative technologies, ranging from nuclear weapons to computers.

For the sake of anticipating future challenges and opportunities, it is worth considering which areas of technology, if any, could play similarly consequential roles this century. One area that has recently provoked a great deal of excitement is *cryptography*, or the study of techniques for encoding, protecting, and authenticating data.

Arguably, progress in cryptography (and its complementary field, *cryptanalysis*, which focuses on decoding) should already be included on any list of the most consequential developments of the past century [13]. The most famous case of cryptography and cryptanalysis in the 20th century may be Germany's use of then-advanced encryption during the Second World War and Britain's corresponding cryptanalysis effort, which at least one historian has estimated sped up Allied victory by more than a year [94]. More recently, cryptographic technologies developed in the last fifty years have allowed the internet (and other long-distance communication channels) to be used for otherwise impractical purposes, such as making financial transactions and sending private messages. In addition, the successful use of encryption has posed a continuing challenge to government surveillance and intelligence programs.

Over the past decade, a number of new cryptographic technologies have begun to emerge (see Tables 1 and 2). These technologies include blockchains, which have enabled the creation of highly reliable records and computer applications that no single party controls; cryptocurrencies, such as bitcoin, which have accrued hundreds of billions of dollars of value and allow users to make digital transactions without relying on traditional financial institutions or traditional fiat currencies; smart contracts, which allow users to enter into agreements that are enforced largely by algorithms; and fully homomorphic encryption, which allows users to process data without having access to it in unencrypted form.[2] Some existing technologies, such as secure multiparty computation, have also become much more practical to implement or found novel applications.

It remains to be seen whether these recent developments in cryptography will be as significant as those that came before. However, a number of fairly radical claims have been made about their importance. I list just a few examples: The United Kingdom's Government Office for Science has described blockchains as the first significant innovation in record-keeping since ancient times [179]. Ralph Merkle, one of the founding figures of modern cryptography, has written a paper arguing that blockchains will enable novel forms of democracy [113]. Jaan Tallinn, co-founder of Skype and the Future of Life Institute, has advocated for the use of smart contract technology to solve global collective ac-

[2] Some of these technologies, particularly cryptocurrencies and smart contracts, differ from more traditional cryptographic technologies in a pair of important ways. First, they depend in part on systems of economic incentives to function, and second, their applications primarily concern the transfer of property. For these reasons, it may be more appropriate to refer to them as "cryptoeconomic" technologies, as some engineers working to develop them currently do [154]. However, since no standard terminology has yet been adopted, I will continue to use the term "cryptography" as a catch-all.

tion problems [35]. Elsewhere, researchers have argued that cryptocurrencies could make it much more difficult for governments to influence or trace private transactions, while others have written that homomorphic encryption could enable novel forms of surveillance that require less infringement of individuals' privacy [83, 166].

Unfortunately, discussions of such claims have often played out in scattered blog posts and papers that are difficult for a non-specialist reader to find or understand. This report is intended to be a contribution to the project of gathering and clarifying these discussions.

In particular, this report is divided into three sections: First, I introduce some recent developments in cryptography, aiming to include close to the minimum level of detail needed to discuss the relevant technologies clearly. Second, I describe several potential long-term, politically significant consequences. Finally, I explore some of the technical limitations and political constraints that could prevent these consequences from arising.

In addition, an appendix discusses the relationship between the future of cryptography and the future of artificial intelligence.

For readers not already familiar with cryptography, the first section is best read in order. However, later subsections are not so dependent on one another.

2 Cryptographic technologies: definitions, explanations, and examples

The set of cryptographic technologies that we will be considering is highly diverse. Of interest in this report will be:

- Public-key cryptography
- Digital signatures
- Cryptographic hash functions
- Trusted timestamping
- Tamper-evident logs
- Blockchains
- Cryptocurrencies
- Zero-knowledge proofs
- Smart property
- Smart contracts
- Homomorphic encryption
- Functional encryption
- Secure multiparty computation and secret sharing

Some of these technologies are new, having been developed primarily in just the last fifteen years. Some are older, but either serve as core components of these newer technologies or continue to find additional applications of their own. In this section, I aim to provide descriptions of each technology that are sufficient to enable informed discussions of their potential applications and limitations.

For an overview, Tables 1 and 2 provide a highly abridged summary. In addition, for a deeper look, I will now also recommend some sources for further reading.

Readers interested in more thorough or technical descriptions of well-established technologies, like public-key encryption, digital signatures, and cryptographic hash functions, can find them in any of a number of widely used introductory textbooks, such as *Introduction to Modern Cryptography* by Jonathan Katz and Yehuda Lindell [99].

Readers interested in blockchains, cryptocurrencies, smart property, and smart contracts can find discussions of them in the textbook *Bitcoin and Cryptocurrency Technologies*, by Narayanan et al. [119]. Vitalik Buterin's whitepaper for the Ethereum blockchain and Ethan Buchman's thesis outlining the Tendermint blockchain also continue to serve as good introductions to blockchain technology itself [41, 34]. Many of Vitalik Buterin's blog posts also provide unusually clearheaded descriptions of different aspects of blockchain ecosystem [38, 39].

Zero-knowledge proofs and secure multiparty computation receive coverage in many intermediate cryptography textbooks. There are a handful of book-length treatments. For an introduction to zero-knowledge proofs that focuses on recent developments, I recommend the "Explaining SNARKs" series on the Zcash blog [186]. For an accessible introduction to

secure multiparty computation, I recommend the paper "Secure Multiparty Computation for Privacy-Preserving Data Mining" by Lindell and Pinkas [108]. In addition, Vaikuntanathan's "Computing Blindfolded: New Developments in Fully Homomorphic Encryption" [171] and Boneh et al.'s "Functional Encryption: Definitions and Challenges" [27] are reasonable introductions to homomorphic encryption and functional encryption respectively.

Finally, it is important to note that the list of technologies I have chosen to investigate is not exhaustive. Among areas I have excluded, the most important may be the subfield of *quantum cryptography*, which applies quantum phenomena to cryptographic tasks. This exclusion is primarily a matter of limited space, although it is also worth noting that many of the most interesting technologies associated with quantum cryptography (such as *quantum money* and *quantum copy-protection*) stand out as particularly far from seeing practical applications [1]. Other potentially significant technologies, not discussed in this report, include *program obfuscation* and *verifiable computing*. Program obfuscation allows users to share computer programs with others while leaving their inner workings opaque, and verifiable computing allows users to outsource computations to others and receive short proofs that the computations have been executed as promised [139, 178]. Both of these technologies are also associated with a number of breakthroughs in the past decade.

2.1 Public-key encryption

Public-key encryption is a technology that allows users to communicate through code without sharing secret information ahead of time [57, 99].

Say that one party, Alice, wants to send a private message to some other party, Bob, using a channel that may have eavesdroppers. For example, Alice might want to share a secret with Bob over e-mail without anyone else—such as a government intelligence agency—being able to learn the secret too. The way to do this is to *encrypt* the message, meaning to encode it in a way that no one else can understand.

The oldest class of encryption schemes, known as *symmetric key* schemes, has been used for thousands of years. These schemes rely on a single shared piece of information, known as a *secret key*, and a mutually understood rule for translating unencrypted messages (or *plaintext*) and encrypted messages (or *ciphertext*) into one another using that key. For example, in the simple "Caesar cipher" the key was a short number, X, and the rule for translating plaintext to ciphertext was to move each individual letter forward by X places in the alphabet.[3]

The trouble with symmetric key schemes is that, to be used, both parties must somehow settle on a secret key without any third parties learning it too. However, the difficulty of communicating secret information such as this is exactly the difficulty that encryption is meant to solve in the first place. Private key cryptography schemes therefore suffer from a "chicken and egg" problem. The problem is further exacerbated by the fact that secret keys cannot be reused without a very significant loss of security.

[3] As an example, the key "1" would turn the message "HELLO" into "IFMMP."

Technology	Origin	Functional description
Public-key encryption	1973	Allows users to communicate through code without sharing secret information ahead of time
Digital signatures	1979	Allows users to identify messages' senders
Cryptographic hash functions	1979	Allow users to associate data with unique "digital fingerprints"
Trusted timestamping	1991	Allows users to timestamp pieces of data
Tamper-evident logs	1979 (ambiguous)	Contain chronological records that cannot be inconspicuously edited
Replicated state machines	1984	Replicate the provision of a service across multiple computers
Blockchains	2008	Replicated state machines that maintain tamper-evident logs
Permissionless blockchains	2008	Blockchains that allow any computer to participate in service provision
Decentralized applications	1980 (ambiguous)	Applications associated with an open-ended set of service providers (as in a permissionless blockchain)
Consortium blockchains	2012 (ambiguous)	Blockchains that allow computers owned by multiple parties (but not any computer) to participate in service provision
Consortium-backed applications	Ambiguous	Applications associated with multiple privileged service providers (as in a consortium blockchain)
Cryptocurrencies	2008	Digital currencies whose ownership is managed through a decentralized or consortium-backed application; are also essential to the operation of permissionless blockchains

Table 1: Summary of cryptographic technologies, part 1

Technology	Origin	Functional description
Zero-knowledge proofs	1985	Allow users to prove mathematical statements to others without conveying additional information
zk-SNARKs	2010	Allow users to do the above succinctly and without back-and-forth interactions
Physical zero-knowledge proofs	2012 (ambiguous)	Allow users to prove statements about physical objects to others without conveying additional information
Smart property	1994	Devices with electronic components that facilitate that transfer of their ownership
Smart contracts	1994	Contracts whose execution is automated to a significant extent
Homomorphic encryption	1973	Allows users to perform certain computations on encrypted data; outputs are also encrypted
Fully homomorphic encryption	2009	Allows users to perform *any* computations on encrypted data; outputs are also encrypted
Functional encryption	2010	Allows users to perform certain computations on encrypted data, such that the outputs are *not* encrypted
Secure multiparty computation	1982	Allows users to run computations with inputs from multiple parties, while allowing these parties to keep their own inputs secret
Secret sharing	1979	Allows users to split private data into shares, which can be recombined to retrieve the data

Table 2: Summary of cryptographic technologies, part 2

Public-key cryptography, first developed in the 1970s, solves this problem. In a public-key scheme, there is not a single key. Instead, each person in a network has a unique pair of keys: one known as their *private key* and one known as their *public key*. Although technical details need not concern us, these are the defining traits of a public-key system:

- There is a *setup algorithm* that each party in the system can use to generate an (almost certainly) unique key pair.
- Each party in the system can announce their public key without revealing their private key. Public keys may also be used as digital pseudonyms.
- There is an *encryption algorithm* that can take a plaintext message and the recipient's public key as inputs and then produce an encoded message as an output. There is also a *decryption algorithm* that that can take an encoded message and the recipient's private key and then produce the original message as an output. By applying these two algorithms in sequence, the sender and recipient can communicate through code.[4]
- There is no practical algorithm that would allow anyone without the recipient's private key to decode the sender's message.

As stated above, public-key cryptography is not a new technology. After its initial development (or, more precisely, rediscovery by academics outside of the classified research community), the possibility of its widespread adoption was for many years considered a threat by government agencies such as, within the United States, the NSA and the FBI [11]. These agencies argued that, without the ability to read intercepted messages, they would be much less able to counter criminal activity and other threats to security. They pursued several strategies to either slow the technology's adoption or outlaw variants that did not grant the government a "backdoor" to decrypt messages using its own special private keys.

Ultimately, it became clear by the late 1990s that these agencies had lost the fight, and, at least within the United States and European Union, all forms of public-key cryptography are now perfectly legal to use [140]. Until recently, though, the vast majority of messaging services still applied encryption in a way that allowed the service provider, and therefore government agencies, to access their users' unencrypted messages. Partially as a reaction to the 2013 Snowden leaks, this state of affairs has begun to change. It has become increasingly common for services to offer *end-to-end encryption*, which, in practice, refers to implementations of encrypted messaging that do not grant service providers viewing privileges [59]. Over the course of just 2016, the number of end-to-end encryption users across all services may have increased by over one billion, due largely to WhatsApp's decision to begin enabling the feature by default [12]. In addition, a number of groups are working to develop practical end-to-end messaging services that can also reliably obscure each message's metadata, such as its recipient [172]. If such projects are ultimately successful, then

[4] To express this mathematically, let *Enc* be the encryption algorithm, *Dec* be the decryption algorithm, *u* be a user's public key, *r* be the same user's private key, and *m* be a message. Then, *Enc(m,u)* is illegible, and *Dec(Enc(m,u),r) = m*. Alternatively, to express this by analogy, we can think of a user's "public key" as actually being a particular lock design, which others use to protect the messages sent to them, and the user's "private key" as the key that opens the lock.

user privacy may increase even further.

While even the use of perfectly implemented end-to-end encryption does not guarantee that one's messages will not be read by anyone other than the intended recipient, it does decrease the odds of successful eavesdropping.[5] Agencies in a number of countries now allege, controversially, that the information channels they rely on are increasingly "going dark" [70].

2.2 Digital signatures

A *digital signature* can be used to demonstrate that a given piece of data was sent by the owner of a particular key pair (see section 2.1) and that it has not been modified since its sending [136, 99].

Digital signatures work in the following way:

- Each party in the system agrees on a *signing algorithm* and a *signature-verifying algorithm*.
- The signing algorithm takes a party's private key and a piece of data and outputs a code known as a *signature*.
- The signature-verifying algorithm takes a party's public key, a piece of data, and a signature, and outputs "Yes" if and only if the signature was generated from the data and the corresponding private key.[6]

The use of digital signatures is currently ubiquitous online and, among many other applications, enables online commerce. For example, when you shop online, your computer verifies that you are in fact connected to Amazon.com (and not a scammer after your credit card details) by checking a signature it sends against a public key known to be associated with the website.

In recent years, some countries have also moved toward assigning their citizens public keys as a form of identification, so that individuals can prove their identities, access personally relevant government records, vote, and even sign legally binding contracts using digital signatures (ordinarily stored on highly protected ID cards) [110]. Estonia is the most notable case, with its citizens having issued hundreds of millions of signatures since the program's inception.

[5] A third party might still read the messages if they gain access to the intended recipient's private key and intercept the message, if they trick the sender into associating the intended recipient with their own private key, if they manage to install malware on either the sender's or the recipient's device, and so on. In addition, there remains a risk that the application developer has misrepresented the security or method of encryption used in their application, as has sometimes occurred.

[6] For example, say that my key pair consists of private key X and public key Y. If I would like to tell you "HELLO," and demonstrate that I am the one telling you this, then I will first input "HELLO," and X into the signing algorithm to produce a signature. I will then send you a message consisting of the word "HELLO" followed by the signature. Finally, if you know my public key, you can apply the verifying algorithm to "HELLO," Y, and the signature, and thereby see that I am the one who signed the message.

2.3 Cryptographic hash functions

A *cryptographic hash function* encodes a piece of data as a code of some fixed length, known as a *hash* (or *digest*) [114, 99]. The function has the following properties:

- It is easy to check that a given piece of data produces a given hash.
- Pieces of data that are only slightly different will produce very different hashes.
- It is impractically difficult to find a piece of data that will produce a given hash, or to find two pieces of data that produce identical hashes.

In a sense, hashes act like "digital fingerprints" for pieces of data. In the same way that each human is associated with an almost certainly unique set of fingerprints, without these fingerprints providing any other information about the person, each piece of data can be associated with an almost certainly unique hash, without this hash providing any other information about the data.

Arguably, hash functions are primarily important as a building block for other cryptographic technologies. We will now discuss a trio of such technologies: *trusted timestamping*, *tamper-evident logs*, and *blockchains*.

2.4 Trusted timestamping

One interesting application area for cryptographic hash functions is *trusted timestamping*, or techniques for demonstrating that a given piece of data existed at a given time [84, 170]. In many cases, the task is significantly complicated by the user's desire to keep the data private at the time of its timestamping.

For instance, suppose that you have some research result that you are not ready to publish, but which you would like to be able to claim priority for. One simple solution is to take the hash of your data and then publish that hash to a newspaper or to a website that can be trusted to reliably log publication times. Later on, you can publish the actual research results, and, by comparing its hash against the published hash, people will be able to verify for themselves that you had the results at the time of the hash's publication.

As an example of this technique, the political organization WikiLeaks will sometimes post hashes of sensitive documents that they obtain to Twitter [158]. If the hashes of eventually released documents do not match—as has happened in at least one case—then it will be clear that someone has modified the documents in the time since WikiLeaks advertised their existence.[7]

[7] Here, Twitter is unintentionally filling the role of *Time Stamping Authority* (*TSA*). It is being trusted, in particular, both to produce authentic timestamps (by publishing accurately dated tweets) and to ensure that these timestamps will remain available into the indefinite future. More sophisticated timestamping protocols can also remove this second responsibility. For example, rather than storing the hashes it receives from users, the TSA can send back signed messages that contain both the hash and the time of its receipt. If the user would later like to convince others that the data existed at this time, then all they need to do is share the signed message along with the data. Further security can be provided by yet more sophisticated proposals, which replace a TSA with multiple trusted parties.

Other use cases for trusted timestamping include preventing the forgery of documents and digital media. Often, the fact that a piece of data is known to predate a certain point in time can provide evidence that it is genuine. Investigators can trust that a timestamped government document, for example, was not concocted after the fact to cover a corrupt official's tracks. Similarly, if a photograph claims to depict a secret meeting between two officials on a particular day, it will be much more credible if it is actually timestamped for that day. As discussed in section 3.1.4, this latter use case may become much more important as artificial intelligence becomes increasingly efficient and effective at forging photographs and videos [4].

As a final technical point, it can sometimes be useful to associate a large collection of data with a single timestamp. Naively, one simple way to accomplish this task is to apply a hash function to the full collection together. However, this method has a significant downside. In particular, using this method, it is impossible to demonstrate that a single piece of data in the collection was used to produce the timestamp without also sharing the rest of the collection, which may be undesirable from the standpoint of efficiency or privacy.

A better method of timestamping a large collection of data at once is to use *Merkle trees* (see Figure 1). Merkle trees are produced by hashing each piece of data individually, then repeatedly hashing pairs of hashes to form a tree structure. The hash that stands at the top of the resulting tree is known as the *Merkle root*. The Merkle root, if published, serves just as well as a "digital fingerprint" for the whole collection. If one would like to prove that an individual piece of data belongs to the timestamped collection, though, then it is only necessary to share that individual piece and a relatively small portion of the Merkle tree, rather than it being necessary to share the other pieces of data as well.

Merkle trees were first proposed in the late 1970s, by Ralph Merkle, and have been used perhaps most often in the context of file hosting and sharing services that wish to assure users that files have not been altered from their most recent versions.

2.5 Tamper-evident logs

A *tamper-evident log* is a chronological collection of records that is designed so that any alterations to records, once they are added, will be easily detectable [55].[8]

Tamper-evident logs are a highly valuable technology, insofar as trustworthy record-keeping plays a crucial role in political and economic life. One would hope, for example, that one's banking records, medical records, tax records, property records, and criminal records continue to be maintained properly.

In many cases, the records in a log are objects of significant interest in their own right. Logs are often instrumentally useful, though, in allowing users to verify the integrity of more sophisticated processes or services. For example, a trusted log of deposits, withdrawals, and transfers can be used to determine the correct balance for a bank account or to restore it to an earlier value if an error should occur. In general, many computer applications are

[8]The terminology here is not entirely standard. However, as a technical note, tamper-evident logs can be classified as a particular variety of what are known as *authenticated data structures*.

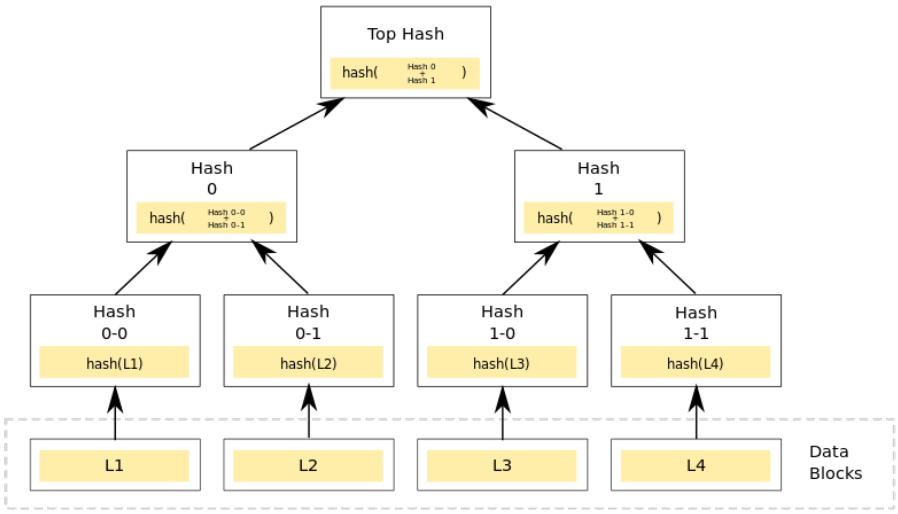

Figure 1: A Merkle tree. The top hash, or *Merkle root*, serves as a "digital fingerprint" for the blocks of data at the bottom. Here, proving that Data Block L1 is consistent with the Merkle root would require sharing just the block itself and the two hashes Hash0-1 and Hash1. There is no need to share the other data blocks. (Image by David Gäthberg.)

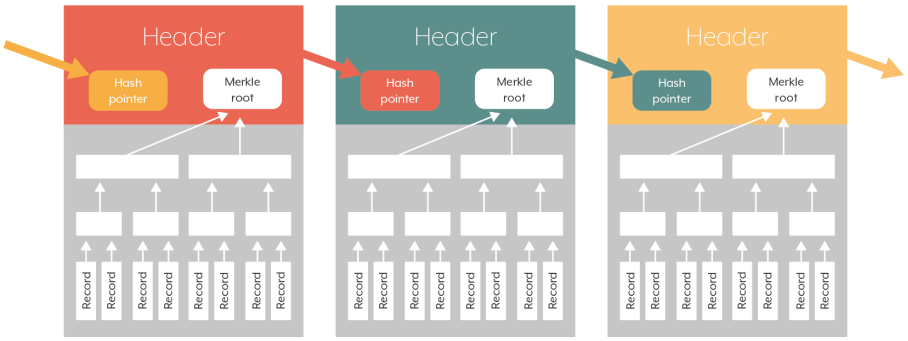

Figure 2: A hash-linked log. Any edit to an individual record will produce an inconsistency with the Merkle root in the block's header. Furthermore, any edit to a block's header will produce an inconsistency with the hash pointer in the following block.

associated with logs, typically referred to as *transaction logs*, that track how they respond to user inputs over time.

Granting the parties maintaining such logs the ability to more easily detect attempts to tamper with them, and to more easily demonstrate to third parties that their logs have not been tampered with, can be a significant boon to security and trust.

One design stands out as particularly effective (see Figure 2). Although the proper terminology is contested, I will refer to it as a *hash-linked log*. Here, the log is subdivided into sequential blocks. Every time sufficiently many new records are collected, they are gathered into a new block and added to the sequence. Each block is divided into two sections, one that contains records and one that is known as a *header*. In turn, each header contains the date, the Merkle root of the records the block contains, and the hash of the previous header.

Suppose that Alice is maintaining such a log—say, for an online payment service that Bob uses—and that she publishes the block headers as they are created. By including Merkle roots in the headers, Alice timestamps each block's records when she publishes its header. By including hash pointers in the headers, though, she also goes further and establishes a canonical sequence of records. It is now impractical, for instance, for Alice to remove a block from the log without producing an inconsistency with the headers that follow it; she also cannot simultaneously create two contradictory versions of a block, then inconspicuously swap one out for the other later on.[9]

Generally, we can understand hash-linked logs as achieving their security through the use of *linked timestamping*, a class of timestamping techniques in which each timestamp references

[9] As a further benefit of including hash pointers, Bob only really needs to know the most recent header to detect tampering. Any attempt by Alice to pass off an alternative sequence of headers as legitimate would be foiled by the sequence's obvious inconsistency with the one that Bob knows.

the one before it.

Although the linked timestamping scheme described here dates back to a series of papers by Haber and Stornetta in the early 1990s, a number of governments and companies have just recently taken an interest in its applications. The aim is both to protect the integrity of their records and to provide greater assurances to their citizens or clients. The Estonian government, for instance, began in 2008 to use a linked timestamping service for logging alterations to some varieties of government documents. The research company Google DeepMind also initiated a project, in 2017, to develop a tamper-evident logging system for use by the British National Health Service [91].

2.6 Blockchains and distributed computing

2.6.1 Background: Concepts in distributed computing

In this subsection, we will take a brief break from discussing cryptography directly. Instead, we will discuss some concepts in the field of distributed computing that will inform the discussion that follows.

A *distributed application*, first, is an application whose operation requires interaction between multiple computers. One familiar example of a distributed application is Google Search. When you use Google, information is constantly being passed back and forth between your computer and computers managed by the company. Your computer sends search requests, and it receives search results back in return.

Google Search is also an example of a distributed application that is well-described by the *client-server model*. In the client-server model, a set of users, known as *clients*, request services from a service provider, known as a *server*. Here, your computer would be considered a client and Google's would be considered servers.

Other familiar examples of distributed applications that fit the client-server model include e-mail, Facebook, and online banking. As one can see, such applications are ubiquitous.

Servers are not, in general, perfectly reliable. They may experience errors, lose the ability to receive requests, or otherwise stop functioning properly. One might worry, in particular, about the *integrity* of the application a server supports or about its consistent *availability* to clients.

One approach to lessening these concerns is known as *state machine replication*. In state machine replication, the operations of a single server are reproduced across multiple copies. These copies, known as *nodes*, communicate with one another to stay in sync and come to consensus about the proper responses to requests from clients. Collectively, the nodes can be conceptualized as constituting a single *replicated state machine* (*RSM*).

The consensus protocols for replicated state machines are designed such that, if some small portion of nodes are faulty, then the application will continue to respond to clients and respond appropriately. The most secure of these protocols, which were largely developed

in the 80s and 90s, can practically guarantee the availability and integrity of an RSM so long as at least two-thirds of its nodes follow the protocol and remain in contact.[10]

One feature of such consensus protocols that will be worth noting is that they often depend on each node storing a log of the requests it has received from clients. Such logs serve as an important tool for preventing and resolving inconsistencies that arise between the nodes. So long as they agree on the ordering of this log and follow the same procedure for responding to sequences of requests, their behavior will be consistent.

The nodes that make up an RSM are typically controlled by a single party, like a technology company. It is also possible, though, to create RSMs whose nodes are distributed among multiple parties. I will call such RSMs *consortium replicated state machines*.[11] At least in theory, this strategy can be useful in contexts where any single actor entrusted with providing a service would suffer from a worrisome conflict of interest. The banking industry, for example, has recently begun to investigate using RSMs maintained between multiple firms to run applications that reconcile financial records. Applications supported by consortium RSMs can be described as *consortium-backed applications*.

A further step in the direction of decentralization is taken by *permissionless replicated state machines*, which place no restrictions at all on which computers can act as nodes. Simply put, anyone anywhere in the world can participate in maintaining a permissionless RSM. Applications run using permissionless RSMs can be described as *decentralized applications*, a class of applications in which the set of service providers is open-ended and not centrally organized.[12]

Intuitively, it is deeply surprising that any permissionless RSM could ever work. Given that standard RSM consensus protocols fail if even a third of the nodes behave maliciously, letting in any node that asks to join has the sound of a disastrous idea. Nevertheless, reliable permissionless RSMs are in fact possible.

Bitcoin, proposed in 2008 by the pseudonymous engineer Satoshi Nakamoto, provided the first-ever example of a permissionless RSM. Designed to implement a decentralized payment application, allowing users to make and receive payments in a digital currency of the same name, Bitcoin has proven itself remarkably successful (see section 2.7). It has developed millions of users and never itself experienced a significant security breach, despite now tracking the ownership of billions of dollars' worth of currency. It has inspired both

[10] In cases where the only concern is that some modes might lose contact, then having even half the nodes function properly is sufficient.

[11] This term, along with the terms "permissionless replicated state machine," "consortium-backed application," and "decentralized application" below, is not standard. However, to my knowledge, there does not yet exist any strongly established terminology here.

[12] There are also some simple applications in this class that do not require service providers to act as an RSM because it is unnecessary for service providers to synchronize their responses to client requests. For example, in the decentralized file sharing application BitTorrent, which launched in 2001, a wide variety of service providers store pieces of files and individually send these pieces to clients upon request. However, most reasonably complex applications do require the synchronization associated with RSMs. A payment application, for example, would open itself up to the possibility of users "double-spending" their money if nodes did not agree about what payments have already been made.

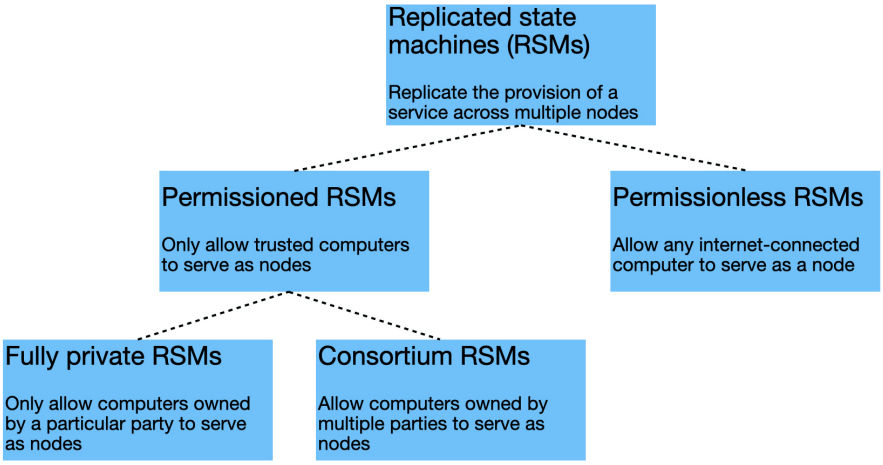

Figure 3: A taxonomy of replicated state machines.

further decentralized payment applications, many of which have been similarly successful, and a number of decentralized applications, such as those associated with the Ethereum blockchain (see section 2.10), that aim to provide more sophisticated services.

Bitcoin was created with a seeming lack of awareness of previous work on RSMs.[13] Despite this fact, Bitcoin's design constitutes an enormous contribution to the field. Most obviously, it provided the first successful consensus protocol for permissionless RSMs (see section 2.6.3). However, as the next subsection will discuss, it also played an important role in integrating the use of RSMs and the use of tamper-evident logs.

2.6.2 Blockchains

The term "blockchain" lacks a standard definition and has been used in several distinct ways.

By some definitions, blockchains are simply hash-linked logs of the sort depicted in Figure 2. By other definitions, blockchains appear to be roughly synonymous with replicated state machines. Perhaps most commonly, the term is also used to refer to a rough constellation of technologies that defies easy summary.

In order to draw a clean line between blockchains and technologies that substantially predate the term "blockchain," however, I will use the following definition:

A *blockchain* is a replicated state machine that maintains a hash-linked log.

Blockchains combine the security properties of these two already discussed technologies.

[13] The whitepaper proposing Bitcoin does not reference any papers in the literature, and, in general, the term "replicated state machine" has only recently begun to be used by developers in the associated community.

The use of an RSM helps to ensure the availability of the log and integrity of the process by which new records are added to it. The use of a hash-chained log then helps to ensure the integrity of records that have already been added, by ensuring that any attempt to tamper with them will be easy to detect.

As discussed in sections 2.5, logs are often objects of interest in their own right. However, in the context of RSMs, logs of clients' requests are typically a tool for implementing and auditing more sophisticated services. In Bitcoin, for example, the complete log of transactions in the currency is used to determine the correct amount in a user's account.

The Bitcoin blockchain, which came online in 2009, is perhaps less notable for being the first blockchain than it is for being the first permissionless RSM. The consensus protocol it implements and the variants it has inspired are general enough—at least in principle—to allow any application to be run as a decentralized application. The fact these early protocols relied on tamper-evident logs was, in some sense, of secondary interest.

However, since 2009, the idea of associating RSMs with hash-linked logs has also received broader interest. In the past few years, companies like IBM have published designs for permissioned blockchains that lightly adapt existing RSM algorithms to provide gains in both integrity and efficiency. A number of companies and government bodies who have not previously used RSMs or tamper-evident logs have begun to explore the possibility of shifting some of their existing services to blockchains. In addition, perhaps even more actors have moved to adopt or expand their use of tamper-evident logs in general. A linked timestamping service provided by the company Guardtime, for example, has begun to be applied to a large portion of Estonian governmental records. In media accounts and promotional materials, such services are typically described as implementing "blockchains." However, in the context of this report's less expansive terminology, this labeling would be incorrect.

Starting in approximately 2011, private sector interest in blockchains spilled over into proposals for implementing consortium blockchains. As stated above, consortium blockchains have attracted particular interest in the financial sector, and several major banks have now publicly explored the possibility of using consortium blockchains to store shared financial records. One significant motivation is the billions of dollars they spend each year in resolving discrepancies between records that they maintain independently [96]. Another smaller early adopter has been the diamond industry, in which several companies have turned to blockchains to better track the movements of individual diamonds as they change hands [175]. Facebook has also recently designed a digital currency system (see section 2.7) that will be run by a consortium of technology companies, financial services companies, and nonprofits. This digital currency system, Diem, is arguably the most high-profile application of consortium blockchains yet.

For something of an overview of proposed applications for permissioned blockchains, of both the consortium and fully private varieties, one good resource is the United Kingdom Government Office for Science's report, "Distributed Ledger Technology: beyond block chain" [179]. Overall, the report expresses a high level of enthusiasm about the economic

Advantages of blockchains	Disadvantages of blockchains
• May increase the availability and integrity of applications • May reduce reliance on trusted third parties (less applicable to private blockchains) • May enable applications that, due to a lack of trust in individual services providers, it would otherwise be unreasonable to use (less applicable to private blockchains)	• Require greater computing power, storage, and communication • May reduce the confidentiality of records, unless sophisticated cryptographic technologies are applied

Table 3: Some advantages and disadvantages of blockchains, in comparison with typical centralized servers

implications of permissioned blockchains, describing them as the first significant innovation in ledger technology since ancient times.

Since its creation, Bitcoin has inspired the creation of many more permissionless RSMs. Most of these implement fairly similar decentralized payment applications, but others implement at least early versions of more sophisticated decentralized applications. There are a number of applications, for instance, that aim to create decentralized markets that allow users to buy storage space on one another's computers. Perhaps most sophisticated blockchain to follow Bitcoin so far, though, is Ethereum, which allows users to design and implement arbitrary decentralized applications (see section 2.10).

Before moving on to describe the consensus protocols that allow permissionless blockchains to work, let us attempt to briefly summarize the core features of blockchains. Compared to the use of centralized servers to provide services, blockchains offer a number of advantages and disadvantages (see Table 3).

As a first advantage, blockchains can sometimes provide greater assurances of integrity and availability. Whereas a single server might go offline, fall victim to a cyberattack, or otherwise exhibit faulty behavior, it will typically be less likely that a large portion of the nodes making up a permissioned blockchain will be faulty in the same way. Although the degree of security associated with permissionless blockchains is still ambiguous (see section 4.2.3), in some cases they may also provide greater protection. In addition, if the integrity of a blockchain's log is compromised, then this will be much easier to detect than a compromise to the integrity of a log that is not tamper-evident.

As a second advantage, blockchains can sometimes be used to reduce reliance on trusted third parties to provide services. The users of consortium and permissionless blockchains

rely on large networks of actors, rather than one particular actor. For instance, whereas a credit card user must rely on company such as Visa to process their transactions, Bitcoin users do not rely on any single actor. In cases where trusted service providers hold an undesirable degree of power, at least from certain users' perspectives, this advantage can be significant. A number of early adopters of Bitcoin, for example, saw it as a way to get around restrictions credit card companies had placed on payments to WikiLeaks.

As a third advantage, blockchains can sometimes be used to provide services that it would otherwise be unreasonable to trust a single actor to provide. Banks, for example, can use consortium blockchains to store and automatically resolve discrepancies in their records without needing to grant control over the records to any individual bank or third party.

As an important disadvantage, however, blockchains are much less efficient than traditional centralized servers. Permissionless blockchains, in particular, are typically able to process no more than a couple of dozen requests per second, and they typically consume very large quantities of computing power, storage space, and bandwidth [54, 60]. These limitations are due in large part to the fact that each individual node must store a complete copy of the log and verify each new request. In general, the whole network can process requests no faster than the least powerful device capable of serving as a node. One upshot, then, is that any reasonably sophisticated or heavily used application cannot in practice be run as a decentralized application. While decentralized payment applications are feasible, even something so simple as a decentralized two-player chess program sits about at the limits of what is currently possible. Decentralized versions of applications like Uber or Facebook of course stand much further away still.

Developers are currently researching a number of potential ways to reduce these limitations. Some permissionless blockchain projects that are still very new, at the time of writing, also claim to have achieved substantial speedups without sacrificing security. In general, though, it is an open question whether permissionless blockchain technology can "scale" to accommodate large numbers of active users or applications that require very frequently updated records (see section 4.2.1).

As a second disadvantage, blockchains are naturally ill-suited for maintaining confidential information. In the case of a permissionless blockchain, everyone in the world is granted complete access to all of the records. Even in the case of a consortium blockchain, the number of parties with complete access may still be greater than desired. Encrypting sensitive records is only a limited solution, since the parties maintaining the blockchain must still have access to enough information to determine whether a given record is valid. Ultimately, those hoping to maintain confidential information on a blockchain may be forced to rely on more sophisticated cryptographic technologies, such as "zero-knowledge proofs" (see section 2.8). The use of these technologies can further add to the cost and complexity of record-keeping.

The next subsection will explore the consensus that enables permissionless blockchains in greater detail. This next subsection is less essential to read than preceding sections, since it focuses on how blockchains *work* rather than focusing on what blockchains *do*. These

details are still quite helpful, however, for understanding efforts to overcome the current limitations of blockchain technology (see section 4.2).

2.6.3 Consensus protocols

Although consensus protocols for permissioned blockchains remain an active research area, practical protocols in this category have been known for decades. As mentioned in subsection 2.6.1, these protocols can assure that permissioned blockchains will continue to function as expected so long as at least two-thirds of their nodes behave properly.

The case of permissionless blockchains is more difficult. If one naively attempts to adapt these older protocols, then two major problems will stand out.

> **Sybil attacks:** A dishonest party, if sufficiently motivated, might carry out what is known as a *Sybil attack* by setting up many different "sock puppet" nodes in order to artificially increase their influence on block creation.
>
> **Lack of incentives for honesty:** Since the parties involved in running nodes are not pre-selected, there may be no basis for expecting them to vote honestly.

In response, the consensus protocols for permissionless blockchains provide two corresponding solutions.

> **Tying voting power to scarce resources:** Voting power is made proportional to demonstrated ownership of scarce resources. For example, it is possible to make voting power proportional to computing power by requiring voters to provide solutions to computationally intensive puzzles. Then, no one can inflate their influence beyond the amount of computing power they possess.
>
> **Rewarding honesty with digital currency:** Voters are incentivized to vote honestly by rewarding them with a quantity of digital currency, to be recorded in the blockchain, if they vote for blocks that the network ultimately converges on (or costing them digital currency if they do not). By this mechanism, a state of affairs in which each node expects the network to converge on honest blocks is likely to be stable.

If this description is still overly mysterious, then the following bullet points provide more detail, describing the working of a somewhat typical permissionless blockchain, along the lines of Bitcoin [119]. Note, though, that this description is not meant to describe how all permissionless blockchains work and glosses over a handful of subtleties.

- There is some network that anyone is free to join, along with some software associated with the blockchain that anyone is free to run. The software implements a protocol that describes under what conditions a request sent through the network is valid. Users who maintain a full copy of the log and run the software are said to be running *full nodes*.

- The blockchain supports a payment application, which tracks the ownership of a digital currency known as a "cryptocurrency." Among the requests that the blockchain can process are requests to send currency from one account to another.
- When users submit requests, they must pay cryptocurrency fees. These fees are extracted from their accounts when requests are processed.
- Each person running a full node maintains a backlog of the requests they have received since they last added a block to their copy of the log. Someone running a full node is said to be *honest* if they are indeed using a copy of the correct software and are, therefore, only processing requests if they are valid. Honest users also do not send contradictory requests to other members of the network.
- Some subset of users running full nodes also compete to solve brute force computational puzzles generated on the basis of the previous block's contents. These users are known as *miners*. The odds that a given miner will solve a puzzle first is proportional to the quantity of computing power they direct at the puzzle. When a miner does solve a puzzle first, they turn their backlog into a new block, B, which they add to their current version of the blockchain. They are also permitted to write and process a special request to deposit a cryptocurrency reward into their own account. Some or all of this reward is derived from the fees paid by the users whose requests are included in the new block. These fees incentive miners not just to solve puzzles, but also to process all of the valid requests they receive. Part of the miner's reward may also include some new quantity of cryptocurrency that was not previously owned by anyone.
- The miner who solves a puzzle first advertises their new block, B, to the others in the network. The other full nodes either add B on to their individual copies of the blockchain, if it is consistent with the earlier blocks in their copies, or ignores it, if it is not consistent.
- If, after at least n intervals, B is part of the longest consistent version of the blockchain being proposed, then B and the blocks preceding it are generally acknowledged to be *confirmed*. This is partly a social phenomenon, in the sense that it concerns the manner in which people assign the contents of a blockchain practical significance.
- If it is the case that, a sufficiently large portion of the time, the person who gains the ability to create a new block is honest, then if n is sufficiently large, it follows that it is overwhelmingly likely that only blocks containing valid records will be confirmed and that they will remain confirmed into the future.[14] Due to the fact that the rewards that successful miners grant themselves will only be confirmed if the blocks they propose are also confirmed, it also follows that miners have strong financial incentives to be honest.

It is worth noting that, in protocols of this form, there is not really any decisive moment in which a "vote" occurs. Rather, the nodes simply take turns proposing blocks, and, over

[14] Intuitively, it seems as though the "sufficiently high proportion of the time" ought to include anything above 50%. In fact, for somewhat technical reasons having to do with the potential to cause confusion by reporting conflicting records to different members of the network, the bar may be as high as 75% [64].

time, it becomes increasingly clear whether the other nodes are predominantly building on top of a given block or ignoring it. Eventually, if a block is buried deep enough in the longest proposed version of the blockchain, then its contents are taken to represent part of the "confirmed" history of records.[15]

The particular protocol sketched above ties the frequency with which a node can propose blocks to the computing power it applies to puzzles. But this is only one of two main approaches to basing influence on scarce resource ownership [20]:

- The selection of block-creating nodes on the basis of computing power, as described in the above protocol, is known as *proof of work (PoW)*. Users known as "miners" demonstrate their ownership of computing power by competing to solve computationally difficult puzzles that are generated on the basis of the previous block's hash. Whoever solves the problem first gains the ability to create a new block, and the probability that a user solves the problem first will be proportional to the amount of computing power they direct at it.
- The selection of block-creating users on the basis of digital currency ownership is known as *proof of stake (PoS)*. In a PoS system, the probability that a user is selected depends on the amount of cryptocurrency they have deposited (relative to the amount of cryptocurrency deposited by other users).

Proof of work is currently widely used, including in Bitcoin. Proof of stake is less commonly used, but has attracted growing interest, in part because it avoids the need to devote huge amounts of electricity to solving mining puzzles. Since some estimates have placed the total electricity consumption of Bitcoin miners as roughly on par with that of the entire country of the Netherlands, and since the costs borne by miners can lead to very high transaction fees for users submitting records, decreasing electricity consumption is a high priority [138].[16]

One particularly valuable feature of both PoW and PoS is that, in practice, they create additional financial incentives for the most influential users to be honest. This is because, to obtain a significant level of influence, these users must have invested heavily in specialized hardware or in the relevant digital currency. If they were to undermine trust in the relevant blockchain through dishonest block proposals, then the value of their investments could evaporate.

In addition, PoW schemes have the added benefit of associating concrete physical costs with the generation of blocks. The further back a record is in the longest version of the blockchain, the more electricity would need to be expended to go back and make an equally long version where that record is absent or replaced.

[15] In fairly rare cases, more than one proposed version of a blockchain can take on social significance. Such an event is known as a *fork* of the blockchain. Forks normally occur when some subset of users decide they would like to adopt a new set of standards for adding new records or forming blocks, while retaining the previous records. In the long run, forks also imply the emergence of two independent cryptocurrencies.

[16] Another benefit of proof-of-stake systems is that they may allow transactions to be confirmed more quickly. There is no need to wait for a sequence of mining puzzles to be solved.

2.7 Cryptocurrency

A *digital currency*, generally, is a form of currency that consists of balances recorded in electronic databases. PayPal credit and video game money serve as two now-mundane examples.

Although definitions vary, we will define *cryptocurrencies* as digital currencies whose ownership is tracked and transferred using decentralized or consortium-backed applications. Bitcoin is the most prominent example. In most cases, including the case of bitcoin, the relevant application is decentralized and relies on the use of a permissionless blockchain.

Cryptocurrencies are objects of interest in their own right. However, as discussed above, cryptocurrencies also play important roles in consensus protocols for permissionless blockchains. They both enable permissionless blockchains and, in many cases, are themselves enabled by permissionless blockchains.

A simplified cryptocurrency system, similar to the one associated with Bitcoin, might work in the following way [118, 119].

- A cryptocurrency is associated with a permissionless blockchain.
- At any given time, the currency is divided up into discrete units known as *coins*, which are understood to be owned by individual users of the currency. Users possess pairs of cryptographic keys, with their public keys serving as pseudonyms.[17]
- A coin of value V is minted (or "mined") if a record of form "A new coin of value V is granted to X" is added to the blockchain's log, where X is the public key of whichever user will own the coin. Users can only mint coins if they have just created a new block, as part of the reward for their work.
- Say that a user with public key X would like to give a coin of value V to a user with public key Y. To do this, they can simply submit a record of form "X gives Y a coin of value V" to the blockchain, along with their digital signature. The record will be classified as valid, and therefore be added to the log, so long as these conditions obtain: there is a previous record granting X the coin, there is no subsequent record showing that X gave the coin away already, and X's signature is correct.[18]

Variations on this system might associate a certain public key with a "central bank" that can mint new coins, or might allow new coins to be created on the basis of a complex voting process. In principle, there is a great deal of flexibility in how a cryptocurrency system might be designed. However, in practice, most systems are not very different than the simple idealized system just sketched.

[17] As a technical note, it is more common in practice to produce pseudonyms by taking the hashes of public keys (known as *addresses*). This is because public keys can be quite long.

[18] The step of verifying that a coin has not already been spent is what implies the need for a complete log of previous transactions. While a traditional digital currency system solves the *double-spending problem* by relying on a trusted third party to maintain the log, a cryptocurrency system achieves a greater degree of decentralization by using a permissionless blockchain instead.

A core appeal of cryptocurrencies is that they can function as a sort of borderless, digital cash. Assuming the system is not set up with some more complicated set of rules restricting valid transactions, any user can send cryptocurrency to any other user anywhere else in the world, without needing to go through institutions like banks or credit card companies. This makes cryptocurrencies perhaps most obviously appealing to people who lack access to such institutions (often referred to as the "unbanked") or who are distrustful of them [173, 97]. The absence of financial intermediaries in cryptocurrency transactions also, in many cases, allows for unusually low transaction fees. For this reason, cryptocurrencies are sometimes used as a low-cost alternative to overseas transfers and local currency exchanges.[19] In addition, cryptocurrencies may allow users to get around political restrictions on the use of traditional payment services. Cryptocurrencies have been used, for example, to make payments in online black markets like the "Silk Road" and to sustain WikiLeaks after major credit card companies blocked payments to it [111].

As stated above, the first successful cryptocurrency was bitcoin, which was launched 2009 and is maintained through a proof-of-work protocol. In the bitcoin system, as in the simplified system described above, additional quantities of the currency come into existence whenever users create blocks, and anyone is free to send or receive currency using an indefinite number of public-key pseudonyms. The initial user base for bitcoin had a strongly libertarian or self-identified "crypto-anarchist" bent, drawn largely by an interest in undercutting financial institutions [131]. However, over time, the currency has slowly crept further into mainstream use, although mostly as an investment rather than as a means of exchange. At the time of this writing, several hundred billion dollars' worth of bitcoin exists.

Many other cryptocurrencies have also been created since 2009, although at the moment only a few have market caps or users bases of comparable size. Some of these cryptocurrencies are associated with blockchains that offer just slight variations on Bitcoin's design. Others, such as Ether (discussed in section 2.10), are associated with blockchains that aim to provide services beyond payment processing; in these cases, the cryptocurrency's role as a necessary design element is perhaps more important than its role as a store of value or medium of exchange.

One interesting question, which may seem in need of answering, is the question of how cryptocurrencies come to be accepted as having monetary value. For most cryptocurrencies, the simple answer is that, like traditional fiat currencies, they are accepted as having monetary value because some initial portion of people accept them as having monetary value. For example, bitcoin has value because some businesses are willing to accept it as payment and because some currency exchanges are willing to exchange it for more established currencies such as US dollars.

For other cryptocurrencies, however, the answer is slightly more complicated. The value of a *stablecoin* is linked to the value of some other asset, such as the US dollar or gold. The most straightforward way for the developers of a cryptocurrency to establish this link is

[19] However, anyone converting a cryptocurrency payment back into their local currency will still need to pay some non-negligible fee.

to set up an organization that pledges to exchange units of the cryptocurrency for units of the asset at some fixed rate. For instance, an organization that supports a gold-backed cryptocurrency might buy up a large quantity of gold, then pledge to accept any offers to exchange N coins for N pounds of gold. Stablecoins whose stability is achieved through this method are known as *asset-backed cryptocurrencies*.[20]

The key advantage of stablecoins, unsurprisingly, is that the purchasing power of individual coins will be stable so long as the value of the linked asset is stable. As a point of comparison, it is common for the purchasing power of an individual bitcoin to rise or fall by more than a factor of two within an individual year. These sorts of radical swings have led most people to treat bitcoin primarily as a risky investment, which has some chance of skyrocketing in value if held for long enough, rather than treating it as a regular currency that can be used to make casual purchases.[21]

The main downside of asset-backed stablecoins, at least from the perspective of more libertarian-minded cryptocurrency supporters, is that they miss much of the original "point" of cryptocurrencies. Asset-backed cryptocurrencies are not as decentralized as traditional cryptocurrencies, such as bitcoin, because they rely on a backing organization with substantial power over the coin's value. The behavior of this backing organization is likely to be highly regulated by traditional financial institutions, and the asset it uses to back the cryptocurrency is likely to be deeply intertwined with traditional financial institutions as well.

Nonetheless, due to the use of blockchain technology, even an asset-backed coin may come with an unusually strong guarantee that no one can easily block transactions or alter account balances. Diem, a high-profile asset-backed cryptocurrency project initiated by Facebook, illustrates the advantage of this feature [107]. Due to low public trust in many countries, Facebook would probably have a rather difficult time if it attempted to push for the widespread adoption of a digital currency that it managed using its own servers. The use of a blockchain, in this case a consortium blockchain, reduces the need for the users of Diem to place any trust in Facebook itself.

Although there is much more that can be said about cryptocurrency systems, I will close this section by summarizing some of their disadvantages when compared to more cen-

[20] There are also other, less popular methods of tying the value of a cryptocurrency to the value of another asset. For example, if a cryptocurrency system grants some particular central-bank-like user the authority to mint or destroy coins, then they can use their powers to target a particular exchange rate. The value of individual coins, relative to the external asset, will tend to grow when existing coins are destroyed and shrink when new coins are minted. This process of minting and destroying coins, in order to cancel out any temporary shifts in the exchange rate, can also be automated through the use of some algorithm that is implemented by the blockchain itself. However, even in this case, there will still be a need for some trusted user or network of users to be given special authorities: someone needs to be trusted to input accurate information about the current exchange rate. The Ampleforth project, for instance, uses an "oracle system made up of whitelisted independent data providers" to input exchange rate data into its blockchain [104].

[21] The vast majority of companies still do not accept cryptocurrency payments. Nonetheless, the number of companies that do accept cryptocurrency payments is growing. Tesla, for instance, has given customers the option of using bitcoin to buy cars. PayPal has also announced plans to support cryptocurrency use, albeit not "direct" use, by making it easy to convert cryptocurrency back into fiat currency at the point of purchase.

Advantages of cryptocurrencies	Disadvantages of cryptocurrencies
• Lower institutional barriers to use • Typically lower transaction fees • Reduced reliance on and need to trust traditional financial institutions • More confidentiality, if further cryptographic techniques used	• Typically slower transaction processing speeds (and maximum transaction volumes) • Encourage hoarding behavior, if not tied to the value of another asset • Place burden of security more strongly on end-user • Less confidentiality, if no further cryptographic techniques used

Table 4: Advantages and disadvantages of cryptocurrencies, in comparison with centralized payment applications

tralized payment services like those offered by credit card companies (see Table 4). These comments will focus on decentralized cryptocurrency systems, rather than cryptocurrency systems supported by permissioned blockchains, both because their disadvantages are especially large and because they represent an especially large break from familiar payment systems.

First, particularly for cryptocurrencies built on permissionless blockchains, commonly used distributed consensus protocols can create inconvenient delays between a user proposing a payment and the network confirming it (see section 2.6.3). Second, since most existing permissionless blockchains can process no more than a couple of dozen transactions per second (see sections 2.6.2 and 4.2.1), most decentralized cryptocurrencies currently face fairly hard ceilings on how much payment activity they can support. The result of these ceilings can be to drive up transaction fees if too many users would like to make transactions at once.[22] Third, cryptocurrencies which are not asset-backed generally have highly volatile exchange rates and are deflationary, thereby incentivizing hoarding behavior rather than spending behavior.[23] Fourth, cryptocurrency users must typically take

[22]In the case of Bitcoin, for example, transactions take about an hour on average to be confirmed. Greater transaction volume has also led the average transaction fee to grow to several dollars, or even dozens of dollars in boom periods, although a typical fee used to be only a few cents. Fortunately, a number of second-wave cryptocurrencies (such as Litecoin) process transactions several times faster and, due to their faster processing speeds and smaller userbases, currently require smaller transaction fees. At the time of writing, a number of newly developed permissionless blockchains, underpinned by novel consensus protocols, have begun to advertise processing speeds thousands of times greater than Bitcoin. These newer projects and the questions that still surround them will be discussed in section 4.2.1.

[23]One view on this point is that the tendency for non-asset-backed cryptocurrencies to induce hoarding behavior means that they should not actually be thought of as currencies. A key feature of a currency is that the people who hold it should often use it to make purchases. Arguably, then, cryptocurrencies like bitcoin are less analogous to traditional currencies than they are to rare paintings. Like rare paintings, these cryptocurrencies are exotic assets that investors sometimes use as alternatives to traditional financial assets; their valuations depend

greater responsibility for the security of their holdings than users of traditional currencies. A cryptocurrency owner who loses their own private key will have absolutely no way to use their coins, and a cryptocurrency owner whose private key is discovered by someone else will have no way to reverse any fraudulent transactions; the additional security features offered by credit card companies, for example, can offer a very large amount of value to users. Finally, just as blockchains are naturally less suited to confidentiality than centralized servers (see section 2.6.2), cryptocurrency transactions can often be less confidential than transactions made using traditional payment services.

This final point may be surprising, sometimes it is sometimes claimed that cryptocurrencies like bitcoin allow users to make anonymous payments. In fact, bitcoin is *pseudonymous*, since payments are still attached to unique public-key pseudonyms, and this distinction makes all the difference [119]. Every payment that a given user's pseudonym makes or receives is logged for anyone else to see, and law enforcement agencies (and other users) have found it easy to attach pseudonyms to real-world identities by analyzing patterns of payments [134]. In addition, if the user does not use an anonymizing service such as Tor, then it may even be possible to associate their pseudonym directly with their device's IP address. Finally, and perhaps most importantly, users who would like to exchange cryptocurrency for some good or service in the physical world will still normally need to expose some aspect of their identities to whoever they are interacting with. Bitcoin, in short, is not very private.

On the other hand, some recently developed techniques for obscuring transactions hint that the long-term trend may be toward much greater privacy for cryptocurrency users. These include "mixing services," which allow users to swap coins in order to frustrate network analysis, and "state/payment channels," which allow sets of users to conduct sequences of transactions and then only publish the final outcomes of these transactions.[24]

However, the most promising techniques, now to be discussed, are almost certainly those that rely on cryptographic tools known as zk-SNARKs.

2.8 Zero-knowledge proofs

Zero-knowledge proofs are proofs of mathematical statements that do not convey any information other than that the statements are true (and that, as a logical consequence, the prover has the knowledge necessary to prove them) [80].

As an illustration, consider the question of whether there is a path around the graph depicted in Figure 4 that touches each node exactly once. Typically, the way to prove to another party that such a path exists is to share one such path, as Figure 4 does, and allow them to check that it meets the criterion. In contrast, a zero-knowledge proof would

on opaque and sometimes baffling social dynamics, and it would be unusual to use them to buy some milk at the store.

[24] Currently, the most popular cryptocurrency designed to obscure transaction details is Monero. It applies "ring signatures," another cryptographic technology, to introduce uncertainty about the origins of payments. Some research suggests that the level of privacy offered by Monero, while greater than that offered by Bitcoin, is still relatively low [116].

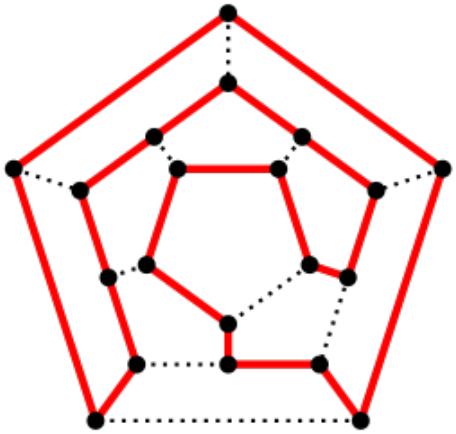

Figure 4: There is a path around this graph that touches each node exactly once. A typical proof of this statement requires sharing the details of at least one such path (as in the above figure). In contrast, a zero-knowledge proof does not require sharing any additional information of this sort. (Image by Christoph Sommer.)

not require any particular path to be shared. The only information that a zero-knowledge proof communicates is the simple fact that whatever statement it is meant to prove is true.[25]

Zero-knowledge proofs were first described in a 1985 paper. Since this time, cryptographers have discovered that all mathematical statements that it is practical to prove at all can also be proven in zero-knowledge [16].[26]

Zero-knowledge proofs can be divided into *interactive* and *non-interactive* varieties [24]. In the interactive variety, a *prover* and *verifier* engage in back-and-forth interactions that ultimately serve to convince the verifier that the relevant statement is true. In the non-interactive variety, which has been more recently developed, the prover simply publishes a proof that anyone is free to verify. While interactive proofs only serve to prove a given statement to one particular party at a time, non-interactive proofs can be used to prove statements to large masses of parties at once.

One recent event of note is the development of *zk-SNARKs* ("zero-knowledge Succinct Non-interactive ARguments of Knowledge") in 2010. These are a variety of non-interactive

[25] The definition of a zero-knowledge proof can also be given more formally. Suppose there is some function *V(X,w)* that determines whether some information *w*, known as a *witness*, suffices to prove some mathematical statement *X*. In a typical proof, the prover shares a witness with verifiers, who then compute *V(X,w)* to determine that *X* is true. In contrast, a zero-knowledge proof is a proof in which the prover demonstrates that they possess a witness of the relevant statement without actually sharing the witness. For above case, *X* would be the statement that the given kind of path around the graph exists, *w* would be a description of a path, and *V* would be a function that checks whether the path is valid and touches each node exactly once.

[26] Specifically, as another technical note, anything in the complexity class *PSPACE* can be proven in zero-knowledge.

zero-knowledge proofs that are particularly short, efficient to check, and therefore practical to use [18].

One application of zk-SNARKs, already alluded to, is in developing cryptocurrency systems that allow users to prove the validity of the payments they record on the blockchain, without also publicly revealing sensitive information that implies the payments' recipients, values, or origins [141]. The first example zk-SNARKs being used in this way is Zcash, a cryptocurrency developed by American and Israeli academics and launched in late 2016. Developers have also been gradually rolling out tools to facilitate the use of zk-SNARKs on Ethereum, one of the most widely used blockchains [7]. It could be only a matter of time until the typical cryptocurrency transaction truly is private [135].[27]

Another application of zk-SNARKs is in developing what Joshua Kroll refers to as *accountable algorithms* [103]. In a recent paper, Kroll describes a general protocol that can be used to provide accountability in any case where an institution is using an algorithm to make decisions that affect individuals. Examples of such cases include the use of algorithms, applied to personal data, to select individuals for search or for tax auditing. In Kroll's protocol, the use of zk-SNARKs achieves two ends. First, zk-SNARKs allow affected individuals to verify that the algorithm has received the approval of an auditing body and meets certain other criteria, without needing to learn the details of the algorithm. Second, zk-SNARKs allow an auditing body to verify that the algorithm they have approved is in fact being used to make decisions, without needing to learn what these decisions are or the personal data used to make them. The ultimate effect of schemes such as Kroll's could be to increase public trust in institutions, while decreasing corruption and opening up the procedures followed by these institutions to greater scrutiny.

It should be noted, however, that the use of zk-SNARKs does come with some downsides. First, zk-SNARKs will always be somewhat less efficient than traditional proofs. Second, the use of zk-SNARKs requires an initial *trusted setup* procedure in which a set of parties generate information that is a necessary input when constructing zk-SNARKs [19]. This procedure also generates some secondary information, informally known as "toxic waste," that could be used to construct fraudulent zk-SNARKs. The parties who participate in the setup procedure must take great pains to demonstrate that they have destroyed this "toxic waste," rather than saving it to exploit it later [164].[28]

Another recent development of note is a series of recent proposals for practical applications of what are known *physical zero-knowledge proofs* [68]. These are protocols that ap-

[27] On the other hand, there is also some evidence that user interest in strong privacy features is fairly low. Bitcoin remains more popular than Zcash even among people actively engaged in criminal activity [149]. Furthermore, only a minority of Zcash's current users choose to use its (optional) privacy features. It is possible that many potential users of Zcash's zk-SNARK technology either do not understand or do not trust its ability to protect their privacy.

[28] Of note, however, is some recent research into another variety of non-interactive zero-knowledge proofs, *zk-STARKs*, that would lack this vulnerability [17]. Currently, the main downside of zk-STARKs over zk-SNARKs is that zk-STARKs are substantially longer. As of 2018, according to the original zk-STARK paper, they require about one thousand times more storage space. This difference in storage requirements is especially important in the context of traditional blockchain-based applications, given the need for many different users to store complete transaction records.

ply the principles of interactive zero-knowledge proofs to the demonstration of physical (rather than mathematical) claims.

Consider the following illustration. Alice would like to prove to Bob that two cups contain the same number of balls without revealing what this number is. To do this she can fill up a bucket with another number of balls, known only to her, then make a *commitment* by stating how many balls will be in the bucket if she pours either cup in. She allows Bob to *challenge* her by picking, at random, one of the two cups to pour in. Then, as a *response*, she shows Bob how many balls are now inside the bucket. Since her initial statement would have had only a 50% chance of being correct if the cups were unequal, the result of this procedure should increase Bob's confidence in their equality. The procedure can be repeated until Bob's confidence reaches any given threshold. In this way, Alice can prove her proposition about the balls in the cups (probabilistically) while still keeping their quantity a secret.

While physical zero-knowledge proofs were first described primarily for pedagogical purposes, helping to illustrate how non-interactive zero-knowledge proofs might be possible, it is now clear that they can have significant practical applications in their own right.

A 2014 paper, published in *Nature*, showed that physical zero-knowledge proofs can be used to verify that a country is disposing of a genuine nuclear warhead, without needing to learn the design details of this warhead [76]. It has since been shown that it is also possible to verify that a given suspect's fingerprints or DNA do not match those found at a crime scene, without needing to collect their fingerprints or DNA [68].

Since very few academics have ever written on physical zero-knowledge proofs, it is plausible that many more applications remain to be found.

2.9 Smart property

Cryptocurrency is not the only form of property whose ownership can be tracked or determined with blockchains.

As a very closely related use case, a blockchain's users might also create *tokens*, whose ownership is recorded in just the same way cryptocurrency ownership is. Users can sell the tokens to others with the promise that these tokens will be exchangeable for a particular service in the future. It is reasonable to think of these tokens as being roughly analogous to the tokens sometimes sold at arcades (which can be exchanged for a certain number of rounds of play at specific games). Alternatively, users may sell tokens purely as collectables, which are valued for roughly the same reasons that rare stamps are valued. Unique tokens, known as *non-fungible tokens* (*NFTs*), have in some cases sold for millions of dollars [15, 63].[29]

[29] An NFT is typically associated with some piece of digital media. For example, an artist may assert that an NFT, in some sense, "represents" a particular animation that they have made. At first glance, the valuations of certain NFTs are rather confusing. However, one should keep in mind, they are at most slightly more confusing than the valuations of many other collectables.

As another well-known case, traditional property records, which might otherwise be stored in another form of database, can of course also be stored using a blockchain. For example, at least as of 2018, the Ghanaian national government has been considering a joint project with IBM to place local land deeds on blockchains in order to lower the risk of meddling by corrupt officials [58].

The link between blockchain-based records and property ownership can also be made much more concrete, however, through the creation of *smart property*: devices with electronic components that facilitate the transfer of their ownership [159, 119].

Consider the following system:

- The device is initially associated with some public key, which belongs to the device's owner. This ownership is logged as a record on the blockchain, which the device can read.
- To unlock or operate the device, its owner must send a message to the device and digitally sign it with their private key. They may send this message through a channel such as a Bluetooth connection or a card reader slot.
- Say that the owner has public key X, another person has public key Y, and the device is denoted by D. If the owner would like to give the device to the person with public key Y, then they can add a record of form "X gives D to Y" to the blockchain, along with their digital signature. Now the device will respond to messages signed by the new owner and will no longer respond to messages signed by the old one.

To make this description tangible, we can imagine that the device is a car, and that the car will only unlock or start if it receives a message signed with the correct key. More limited access rights (such as the right to use the car only on a certain day) could also be granted through a similar scheme.

With the rise of the *internet of things* (*IoT*), or the trend of more and more electronic devices having internet connections, it seems like the large-scale implementation of physical smart property could be feasible [48]. This implementation could be achieved using permissionless blockchains or consortium blockchains maintained by the devices' producers. The advantage of using such blockchains, here, is that it would be unreasonable in any other case to trust a single actor with running an application that directly controls a wide swath of physical property.[30]

However, it remains to be seen whether there will be any significant consumer interest in smart property. As with cryptocurrency, some users might be attracted by the idea that smart property can reduce reliance on centralized institutions that track property ownership or protect property from theft. Smart property might also offer value by reducing inefficiencies in transactions and consolidating records of ownership. On the other hand, one could argue that there is not much hardship involved in (e.g.) transferring the ownership of a car in the traditional way.

[30] More broadly, a number of companies are also investigating the advantages of using blockchains to provide greater security for their own IoT devices.

We should note that essentially the same privacy concerns that exist for cryptocurrencies exist for smart property too. Any smart-property transactions implemented on the most widely used permissionless blockchains today would be highly visible (although future systems using zk-SNARKs may be able to offer more privacy), and even specialized consortium blockchains might present greater privacy concerns than less comprehensive centralized databases. This would seem to be an important limitation.

Finally, it is worth noting that systems of property ownership and exchange that are more complex (and less decentralized) than the one described above could also be designed. For example, we can imagine a smart-property car for which, by design, a government-held private key is always capable of shutting down or transferring ownership, should an appropriate legal order be given.

2.10 Smart contracts

A *smart contract*, broadly defined, is a contract whose execution is automated to a significant extent.

Vending machines provide a simple illustration. When someone places money into a vending machine and receives a candy bar in return, the machine is, in effect, executing a contract between the buyer and seller.

The concept of a smart contract was first formulated by Nick Szabo in 1996, who argued that, beyond offering efficiency gains, smart contracts can be used to reduce the role of trust in contract execution [159]. If a contract can be enforced automatically, and the relevant mechanism can be made sufficiently transparent and resistant to tampering, then the parties to a contract will not need to trust the other to execute their end of it; they will also not need to trust any third party to act as an intermediary or enforcer.

By enabling applications whose execution does not rely on trusted third parties, consortium and permissionless blockchains may offer a particularly ideal platform for implementing smart contracts.[31]

Consider, for example, the simple case of a decentralized application for playing chess: Players take turns submitting moves, and a digital currency deposit is automatically transferred from the loser to the winner. This application is, in effect, executing a contract between the players. Here, though, there is no need to trust the loser to hand money over to the winner, an escrow service to handle the transfer, or a court to extract compensation for dishonesty.

The topic of smart contract design received a major boost in 2013, when Vitalik Buterin proposed an influential design for a blockchain known as Ethereum. Finally launched in 2015, Ethereum offers its users the ability draft and execute any smart contract that can be expressed in terms of the blockchain's state.

[31] Unless otherwise stated, the use of the term "smart contract" throughout the rest of the report can be assumed to refer to a smart contract implemented using a consortium or permissionless blockchain.

To gain some intuition of how this can work, consider the following (highly idealized) description of a smart contract's creation and execution:

- Suppose a user with public key *X* would like to sell a smart-property car, denoted by *D*, to a user with public key *Y*, at the cost of *V* units of cryptocurrency.
- To accomplish the transaction, the seller can submit a contract of this form: "If *X* and *Y* both submit signed messages approving the transaction, then transfer *D* to *Y* and a coin of value *V* to *X*." This contract is then logged. If both the buyer and the seller proceed to submit messages of approval, signed with their respective private keys, then it will be recorded that the ownership of the two items has switched.
- The seller gains the ability to spend the coin, and the buyer gains the ability to operate the car and demonstrate that it is rightfully theirs. The trade has been accomplished, without any need for a trusted intermediary and without any risk of either participant failing to follow through.

Naturally, the conditions for transferring an item through a blockchain-based smart contract can only be expressed in terms of information that is stored in the blockchain. However, this does not make it impossible to set conditions that depend on the external world.

Say that two users would like to enter into a bet about the high temperature for tomorrow. A simple way to carry this bet out would be to write a smart contract that will transfer cryptocurrency, automatically, when a certain trusted friend submits a signed report to the blockchain. Trusted services could also be set to print information about the weather onto the blockchain to facilitate weather-related contracts.

In addition, there are some new or emerging services that use a mixture of reputation systems and consensus protocols, associated with conditional payments, to incentivize users to provide reliable inputs to others' smart contracts. These services are known as *decentralized oracle systems*, and have so far been most extensively explored in the context of a betting market application known as Augur, which itself functions by executing smart contracts among bettors [130].[32]

Smart contracts may also be used as building blocks, of a sort, for designing higher-level applications. Augur, just mentioned, is one example. Other examples include proposed applications that would use smart contracts to facilitate the rental of cloud computing resources or storage space. In fact, Ethereum's smart contract system is flexible enough to build any possible decentralized application.[33]

[32] As a very simple example of how a consensus protocol like this could work, consider again the case of two users who want to bet about the high temperature in their city. They could create a smart contract that depends on the input of several other parties who do not know each other, such that the contract will pay whichever of these inputs does not diverge from the median input by more than a degree. If these parties cannot collude, then they will be incentivized to converge on the truth, since they should expect their self-interested counterparts to converge on it too. A more sophisticated version of this protocol, for repeated bets, might track "reputation points" for users who input information, such that users with higher reputations receive more weight in disagreements and earn higher premiums.

[33] In technical terms, Ethereum provides a "Turing-complete" language for writing applications.

Unfortunately, the same limitations faced by decentralized applications discussed in section 2.6.2 still apply. All established permissionless blockchains, including Ethereum, are too inefficient to run anything beyond fairly simple applications. Again, even something as complex as an application that checks who has won a game of chess is pushing up against Ethereum's current limitations.

While researchers are currently investigating a number of ways to "scale" blockchains to accommodate much greater numbers of users and much more complex applications, it is not yet clear how effective any such techniques will be (see section 4.2).

2.11 Homomorphic encryption

Homomorphic encryption is a form of encryption that allows computations to be run on encrypted data, such that the outputs of the computations are encrypted as well [171].

For instance, we say that an encryption scheme is "homomorphic under addition" if we can encrypt any two numbers, perform a certain operation on them, and then decrypt the result to return their sum. Efficient schemes that are homomorphic under only addition or only multiplication have been known for a number of years.[34]

A scheme is known as *fully homomorphic* if it allows any computable function to be computed on encrypted data. The first fully homomorphic scheme was invented in 2009, by Craig Gentry [72]. Since this time, a number of superior alternatives have also been proposed.

The basic appeal of fully homomorphic encryption is that it makes it possible to create applications in which service providers do not require access to clients' unencrypted data. Possible examples include cloud computing services that do not need to know the data they are being asked to compute on, online medical services that flag health concerns on the basis of individuals' DNA without needing to know their genetic data, and targeted advertising services that do not need to know the interests that users have indicated through their browsing behavior.

However, fully homomorphic encryption has not yet found significant use, as all known schemes are highly inefficient. The most efficient schemes discovered so far still result in many-orders-of-magnitude blowups in the size of the encrypted data as it is operated on, and in the time the operations take to complete. For an example, in 2014, it was the case that even a well-optimized implementation running on a moderately powerful computer might be incapable of performing more than 50 multiplications per second (with addition being much less costly) [8]. This is actually a vast improvement over Gentry's original scheme, but still enormously slow. To a very rough approximation, fully homomorphically encrypted data can be regarded as about a billion times less efficient to compute on than unencrypted data.

[34] Today, the most widely applicable homomorphic encryption scheme is likely the *Paillier encryption scheme*, which is homomorphic under addition while also allowing for multiplication between encrypted inputs and unencrypted ones. This is sufficient to compute any *linear function* of encrypted data.

Some help is provided by *somewhat homomorphic encryption* schemes, developed concurrently, which offer significant (but not decisive) speedups at the cost of allowing for only a finite number of multiplications. Over time, further progress in discovering efficient algorithms and in developing more powerful computers could make fully or somewhat homomorphic encryption practical in a wide range of cases. For the moment, however, their applications remain relatively narrow [117].

In the following two sections, we consider an additional two technologies that allow computations to be run on private data. These are *functional encryption* and *secure multiparty computation*.

2.12 Functional encryption

Functional encryption is a form of encryption that allows particular functions of encrypted data to be computed—such that, in contrast with homomorphic encryption, the output is not encrypted [27].

For example, using functional encryption, a hospital could encrypt its patients' medical records in a way that allows different categories of hospital workers, holding different keys, to compute only the information they need. A user of a cloud storage service could also encrypt their files in a way that allows the service, or government investigators, to compute only whether the files contain certain restricted material (like copyrighted media).

In general, a functional encryption system works in the following way:

- There exists a *setup algorithm* that can produce an almost certainly unique key pair, consisting of a *public key* and a *master private key*. As in a regular public-key encryption system, the public key, along with an *encryption algorithm*, can be used to encrypt data.
- There also exists a *key-generating algorithm* that can, given a master private key and a function, f, produce a *special private key*, k_f.
- Finally, there exists a *decryption algorithm* that can, given k_f and a piece of data encrypted using the master private key, produce the function $f(x)$.
- In a simple case, it is possible to encrypt one's data using a master private key, then grant others the ability to ascertain certain information from the data by distributing special private keys.

The concept of functional encryption is quite new, first appearing in academic papers in 2010 [124]. Researchers have since developed a number of general functional encryption systems—meaning, systems whose key-generating algorithms can be used to produce a special key for any desired function [79, 78].

Similar to homomorphic encryption, functional encryption can make it easier to provide privacy-preserving services. In the two examples given above, functional encryption allows the hospital workers and the government investigators to gain just exactly the information that is necessary to do their jobs.

In addition, while only a small amount of research has addressed practical implementations of functional encryption, it can, for many functions, be much more efficient than fully homomorphic encryption [150]. Nevertheless, applications of functional encryption have been little explored.

2.13 Secure multiparty computation and secret sharing

Finally, *secure multiparty computation* (*MPC*) refers to techniques that allow users to run computations with inputs from multiple parties, while allowing these parties to keep their own inputs secret [151].

As a popular illustration, consider the service provided by dating applications like Tinder, in which two people will receive a notification if and only if they both express interest in the other. We can think of these applications as computing a function *f(X,Y)*, where *X* and *Y* represent the users' inputs and *f* represents whether or not both inputs are expressions of interest. More importantly, these applications compute *f(X,Y)* for the two parties without requiring them to share their inputs with one another. Since "no match" is compatible with either one or two inputs of "not interested," a user who inputs "not interested" gains no information at all.

These applications solve the "dating problem" by acting as a trusted third party: they collect both parties' inputs and then promise to compute the function correctly and to not share their inputs with anyone else. We might ask, though, whether it is possible to achieve the same end without any trusted third party at all.

In fact, as Andrew Yao proved in a classic 1982 paper, MPC protocols can be used to solve the dating problem without trusted third parties [185].[35] By engaging in a fairly complex sequence of interactions and individual computations, the two potential daters can together compute *f(X,Y)* while maintaining their individual privacy. Since this time, cryptographers have also learned that MPC protocols can be designed to compute any joint function of private inputs, for any number of parties. In just the past decade, there has been substantial progress in developing MPC schemes that are practical to implement, with efficiency increasing by several orders of magnitude [74].

The first economically significant application came in 2008, when MPC was used to carry out a secure sugar beet auction in Denmark with over one thousand bidders [26]. In this case, the various parties determined the winning bid for each item without otherwise sharing their bids with one another. Without MPC, they would have needed to share their bids either with one another, or with a third party who was trusted to report the proper outcome.

One frequently discussed use case for MPC is in secure voting systems, which would allow voters to jointly determine the outcomes of elections without sharing their votes with one another or trusting any third party to tabulate them [53]. Another use case is training machine learning systems across multiple privately held datasets, such as individual medical

[35] Technically, Yao showed this for a more general problem he called the "millionaires' problem."

records or classified information held by different government agencies [108].

In addition, we can understand the use of homomorphic encryption to provide privacy-preserving services (see section 2.11) through the lens of MPC [47]. Specifically, client-server applications (see section 2.6.1) in which the client homomorphically encrypts the data they send to the server can be interpreted as implementing a particular variety of MPC, in which the client's input is the data to be encrypted, the server's input is the algorithm to be applied to that data, and only the client receives the computation's output. While processing data in this way is associated with enormous computational overhead, as stated above, there exist other MPC protocols that accomplish the same end much more efficiently. This reduction comes at the cost of requiring extensive communication between the server and client and, in general, requiring the client to be a much more active participant. Nevertheless, these more communication-heavy MPC protocols are comparatively practical.

Another way to use more efficient MPC protocols to offer privacy-preserving services, in this case without placing such a large burden on the client, involves a cryptographic technique known as *secret sharing* [148]. In secret sharing, a private piece of data, or *secret*, is divided into a number of *shares*. These shares individually provide no information about the secret, but if sufficiently many of the shares are combined, then the secret can be retrieved.

Consider the following protocol:

- Alice has some data, X, and she would like some function of it, $f(X)$, to be computed. To make the case concrete, we might imagine that X is personal medical data and $f(X)$ is a list of health risks suggested by the data.
- Alice splits her data into a number of shares, using secret sharing, and distributes the shares among an equal number of other parties.
- Now, so long as each party keeps its own share private, the problem of computing $f(X)$ is transformed into an MPC problem with the shares as inputs.
- Running the MPC protocol until just before its end results in each party holding, in effect, a share of $f(X)$. At this point, they send their output shares to Alice alone, and she recombines them to learn $f(X)$. She has received a service from these parties without sacrificing her privacy.

In general, the use of secret-sharing protocols makes it possible to separate the parties that provide a computation's data inputs, the parties that receive its outputs, and the parties that perform the bulk of the computational work. We can divide the relevant parties up into the *input parties*, *computing parties*, and *results parties* (see Figure 5) [160]. The input parties divide shares of their data among the computing parties. The computing parties compute shares of a joint function of this data. The results parties receive these shares and join them together to learn the final output. In a classic MPC protocol, these three sets of parties would all be the same. In the above description, though, Alice is the only input party and the only results party, and she is not a computing party.

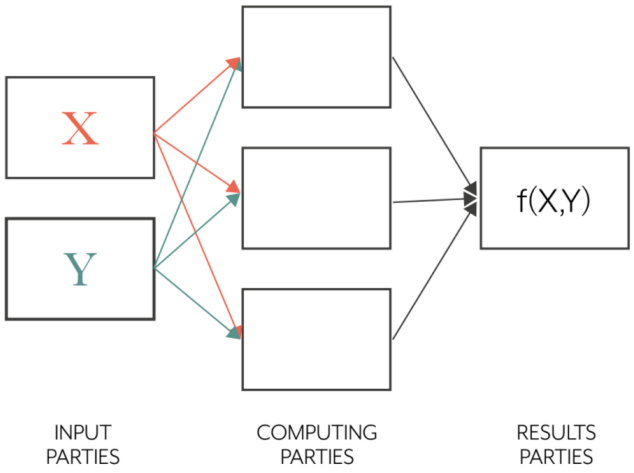

Figure 5: In this MPC protocol, the input parties use secret sharing to distribute shares of their private data, *X* and *Y*, among the computing parties. The computing parties compute shares of a function of this data, *f(X,Y)*, and then send these shares to a results party that combines them to learn *f(X,Y)*. In general, a single party might play multiple roles. In classic MPC protocols, the input parties, computing parties, and results parties are all the same.

The most obvious security challenge with secret-sharing MPC protocols is the need to ensure that the computing parties do not collude to combine their shares, thereby learning the relevant secrets. Another worry is that computing parties might not carry out their portions of the computations correctly.

While stricter security standards imply stronger requirements, there exist protocols in which even a single honest party is sufficient to ensure privacy of the relevant data. Furthermore, conditional on the assumption that the computing parties have access to a trustworthy public log, there exist protocols that also allow even uninvolved parties to verify that the computation was performed correctly [14].

It is not unreasonable to think that these requirements—a trustworthy public log and a single honest party—can often be met. Blockchains, or even just simple tamper-evident logs, are a good way of meeting the first requirement. Economic incentives may provide an ideal way of meeting the second. One solution might be to rely on companies whose business models depend on their reputations as reliable computing parties for MPC protocols. Another solution is being explored by an MIT-based project known as Enigma [187]. Enigma is a decentralized application (see section 2.6.1) meant to implement a market for MPC computations. Parties contract others to serve as computing parties, and these parties lose substantial cryptocurrency deposits if they are found to engage in malicious behaviors.

The use of MPC is still associated with substantial overheads. For example, at the time

of writing, the most efficient MPC protocols increase the time required to train a deep learning system by about a factor of one thousand [177]. However, these overheads are far smaller than the overheads associated with fully homomorphic encryption. They are already quite manageable for short computations. Further innovations may decrease these overheads even further.[36,37]

[36] At the time of writing, OpenMined is perhaps the most active project attempting to reduce barriers to the widespread adoption of MPC [126]. Their primary focus is on developing software libraries that make it easier for data scientists and machine learning engineers to implement efficient MPC protocols, alongside other privacy techniques. They also publish educational materials and advocate for the adoption of MPC and related techniques.

[37] One unifying perspective on zero-knowledge proofs, homomorphic encryption, functional encryption, and secure multiparty computation is that they support what might be called *structured transparency*: precise controls on who can know what, when they can know it, and what they can do with this knowledge (see section 3.1.2) [167]. For instance, the computing parties for a computation performed using MPC, unlike the computing parties for a typical computation, do not need to know what its inputs or outputs are. The party reading a zero-knowledge proof, unlike the reader of a typical proof, does not need to know anything besides the simple fact that the proof is valid. The social value of structured transparency techniques is that, in many cases, they can enable desired uses of information without also opening the door so widely to undesired misuse.

3 Speculative consequences

In this section, I gather and discuss a number of possible consequences of the developments described above.

Since the full range of possibilities is of course very broad, I have limited myself to consequences that I believe would have large-scale political significance. For example, although a number of banks have investigated the possibility that consortium blockchains will allow them to reconcile their records more efficiently, it is not sufficiently clear to me that the effects of these developments would seep far outside the financial industry.

I will be considering the following seven possibilities:

1. Information channels used to conduct surveillance could "go dark"
2. Privacy-preserving surveillance could become feasible
3. Non-intrusive agreement verification could become feasible
4. It could become easier to combat forgery
5. The roles of banks, technology companies, voting authorities, and other traditional institutions could shrink
6. It could become feasible to solve collective action problems that existing institutions cannot
7. A new variety of institutions, known as "decentralized autonomous organizations," could emerge

There are of course reasons, sometimes quite strong reasons, to be skeptical of each of these possibilities. I discuss some reasons for skepticism within this section, but section 4, "Limitations and skeptical views," describes several in significantly greater detail. This section of the report is less focused on the question "What is likely to happen?" than it is on the questions "What are the most interesting predictions that other researchers or engineers have made?" and "What predictions would be truly consequential if they came true?"

In addition, it should be noted that a period of several decades can often separate the initial development of a technology and its eventual refinement and adoption [85]. Even predictions that are plausible in the long run may still be implausible, say, within the next ten years. In fact, a number of these possibilities explicitly require very significant technological progress.

3.1 Consequences from technologies other than blockchain

3.1.1 Information channels used to conduct surveillance could "go dark"

Three trends could conceivably make surveillance significantly more difficult: first, the growing use of end-to-end public-key encryption; second, the emergence of digital currency systems that apply zk-SNARKs and other techniques meant to obscure economic transactions; and, third, the potential for methods of computing on confidential data to alter the economics of private data collection.

As discussed in section 2.1, it has only recently become common for major providers of messaging services to offer "end-to-end encryption" for all messages that users send. "End-to-end encryption" refers to a process of encryption and decryption that occurs solely on the sender's and receiver's devices, meaning, in practice, that the service provider cannot access the messages even if they would like to [59]. This, further, prevents government agencies from gaining access via the service provider.[38] Just over the course of 2016, the number of end-to-end encryption users likely increased by more than a billion [12]. This change is due to a number of major applications, most notably WhatsApp, beginning to offer the service by default. In addition, there is the less widely recognized prospect of increasingly practical end-to-end encrypted messaging systems that obscure not just the content of messages but also the associated "metadata"—particularly, data about who is messaging whom [59, 172]. A number of prominent officials, such as former FBI director James Comey, have responded to the growing use of end-to-end encryption by sounding an alarm that information channels they rely on are "going dark" [51, 70]. While the Chinese government has moved to crack down on the use of this technology, civil liberty concerns provide significant barriers to regulation in many other countries [30].[39]

The development and adoption of privacy-preserving cryptocurrencies may lead a further information channel to "go dark." While it is sometimes perceived that today's widely used cryptocurrencies already offer significant privacy, this perception is mostly mistaken [134, 119]. As discussed in section 2.7, for most cryptocurrencies the full details of every transaction are recorded by the relevant blockchain for everyone to see, and it is in practice fairly easy to tie users' pseudonyms to their real-world identities. If cryptocurrencies do ultimately make surveillance much more difficult, then, it may be the result of cryptocurrencies that apply zk-SNARKs (or other varieties of non-interactive zero-knowledge proofs). At least one new cryptocurrency, Zcash, claims to offer users the ability to make transactions whose contents and participants are obscured from other users. Zcash is not yet widely used, but the developers have been gradually expanding zk-SNARK support on the Ethereum blockchain (see section 2.10). Despite some lingering security, efficiency, and user demand concerns, in the long run it seems plausible that cryptocurrency transactions will become significantly less transparent. In addition, even if zk-SNARKs or other forms of zero-knowledge proofs do not become widely used, a handful of other methods of obscuring transactions, including "mixing services," "ring signatures," and "state/payment channels," could achieve similar effects.

Finally, methods of running computations on encrypted data might lead further information channels to go dark. "Don't Panic," a 2016 report associated with Harvard's Berkman Center for Internet and Privacy, argues that the "going dark" problem is overstated due, in large part, to private companies' economic incentives to collect user data [70]. While this point is likely robust in the short run, and indeed the medium run, it may not be in

[38] Although, as mentioned above, even the use of perfectly implemented end-to-end encryption does not guarantee perfect security, for instance, if the user's personal device is insecure or if they are tricked or coerced into revealing their password. In addition, an agency might still convince service providers to weaken protections on some users through software updates.

[39] The European parliament has even recently considered a proposal, motivated by civil liberty concerns, to prevent member states from banning the use of end-to-end encryption [75].

the long run. As discussed in sections 2.11–2.13, technologies such as secure multiparty computation, functional encryption, and homomorphic encryption can reduce economic incentives to collect unencrypted user data. Just as end-to-end encryption makes it possible for a service to send messages for a user without gaining access to their contents, these technologies make it possible to send users targeted ads without gaining access to their interests, to train machine learning algorithms on users' activity logs without gaining access to these logs, and so on. In addition, there are a number of less powerful, but already practical techniques for learning from user data in a privacy-preserving manner [2]. For instance, *differential privacy* techniques work by making random alterations to datasets in order to prevent the individuals included in the datasets from being identified. In 2016, Apple began to apply these techniques to data associated with the use of its devices [161].

On the other hand, it is not impossible to imagine a future in which cryptographic developments could support the expansion of surveillance in certain ways. For example, as will be discussed in the next section, methods of computing on confidential data could make it easier to gather security-relevant information about individuals without also gaining access to irrelevant private information. This capability could make it easier for government agencies to engage in robust surveillance without running up against privacy concerns.

In addition, it is very plausible that other technological developments, mostly unrelated to cryptography, will ultimately play a larger role in increasing the ease of surveillance. Video surveillance augmented by artificial intelligence, for example, may become much more pervasive. Some authors have also raised the possibility that, in the coming decades, small and difficult-to-detect drones could become highly effective tools for surveillance [181]. More prosaically, the authors of the Berkman Center report highlight the growing number of sensor-bearing devices that make up the "internet of things" as one long-run trend that is making surveillance easier. Ultimately, when it comes to the future feasibility of surveillance, new developments in cryptography might be of fairly secondary importance.

3.1.2 Privacy-preserving surveillance could become feasible

In discussions of surveillance, the supposed trade-off between privacy and security is frequently discussed. While this "trade-off" narrative is heavily simplified in a number of ways—excluding, among other points, the fact that privacy can itself provide security against exploitation—it does still capture something of the truth [181, 152].

A trade-off will exist to whatever extent the ability to access security-relevant information also entails the ability to access information that would ideally remain private. For example, for a police officer manually searching bags, learning whether someone is carrying a weapon also requires learning everything else they are carrying. The choice is either to forgo security-relevant information (whether there is a weapon in the bag) or to obtain extraneous private information (what else is in the bag).

Fortunately, there is no fundamental reason why granting a party the ability to learn security-relevant information *must* entail granting them the ability to learn anything else.

Bomb-sniffing dogs, which report only whether there are explosives within a given bag, can be seen as an example of a technology that minimizes the privacy-security trade-off in the case just described (at least, in the idealized case that the dogs have perfect accuracy) [31]. Technological progress may make it become possible to minimize the trade-off in a wider and wider variety of domains. Or, to re-express this point in the terminology of one recent paper, it may become increasingly possible to achieve *structured transparency* [167].

For the surveillance policy, the basic principles are these:

> **Surveillance tasks can be automated:** If progress in artificial intelligence continues, then it will become possible to automate an increasingly large portion of the tasks currently performed by human analysts (as well as tasks that human analysts cannot currently perform). For example, AI systems are likely to become more capable of identifying faces in videos, noticing suspicious patterns of transactions, and judging from private messages whether someone is engaged in illegal activity.
>
> **If a surveillance task can be automated, then it can be completed without access to private data:** In any case where an AI system is extracting security-relevant information from private data, it is, in principle, possible to design the system in a way that does not require the party running it to collect the data in unencrypted form. The relevant technologies, which enable computing on confidential data, include secure multiparty computation, functional encryption, and homomorphic encryption (see sections 2.11–2.13).[40]

Generally, work on the applications of cryptographic technologies to privacy-preserving surveillance has received fairly little attention [66, 145, 22, 69]. However, in recent years there have been some useful proofs of concept. The most noteworthy case is probably the development of an MPC-based system to detect cases of value-added tax fraud in Estonia: although the system was not ultimately adopted, it was designed to identify discrepancies between declared sales and purchases without violating the confidentiality of individual companies' financial records [25].

There have also been proposed protocols for privacy-preserving "set intersection" searches, which identify individuals who reappear across several datasets of interest [146]. For example, set-intersection searches are sometimes used to identify criminals from multiple sets of cell tower records associated with locations where they are known to have committed crimes. A trivial form of secure multiparty computation over the datasets could prevent the investigators from needing to access the full datasets in cases like these. Similarly, widely used "contact-chaining" techniques, which generate lists of suspicious individuals on the basis of social network analysis, could easily be implemented without extraneous

[40] An intermediate privacy solution, which agencies such as the National Security Agency already employ to varying degrees, is to collect and store unencrypted data, but limit the access of individual analysts. An analyst, for example, may be allowed to make only a limited set of queries to the database and only view the portion of records that are classified as matching these queries. However, systems like this provide weaker assurances of privacy, in that they depend on the continued self-restraint of the agencies holding the data. The act of collecting the data is also subject, in many cases, to significant legal constraints.

data collection.

To illustrate the potential breadth of long-run possibilities, I will now describe a general protocol for privacy-preserving mass surveillance with secure multiparty computation. This protocol is inspired by Andrew Trask, a machine learning researcher, who recently described a similar protocol that employs homomorphic encryption instead [166]. The protocol would infeasible to implement today, but can interpreted as a kind of "limiting case" of what might be possible in the future.

- Technology companies and service providers each maintain their users' data independently. No government agency collects this data.
- A number of these companies maintain subsidized data centers, which are designed to engage in secure multiparty computations with a certain government agency.
- The agency develops a classifier system intended to identify criminal activity on the basis of personal data. An example of a simple classifier would be one that just performs set-intersection searches on cell records, identifying individuals who were present at multiple linked crime scenes. A more sophisticated classifier might use a variety of more sensitive data, including the contents of private messages, to identify individuals who appear to be involved in particular criminal activities. The classifier is tested to ensure that it does not exceed a mandated maximum false positive rate.[41]
- This procedure is overseen by an independent auditing body. The body confirms that the classifier has the properties claimed. Once the classifier is put into use, the auditing body will also be tasked with confirming that its false positive rate remains below the maximum allowed value.
- Now, the agency engages in a secure multiparty computation with the relevant companies. The inputs to the computation are each company's private data and the parameters of the agency's classifier. (Data from other sources might also be included through secret sharing.) The output, which is received only by the agency, is a list of individuals that the classifier deems sufficiently suspicious. The agency may ultimately request further data on these individuals from the companies, subject to judicial approval.
- Different classifiers could be deployed for different crimes. The explicitly coded threshold for reasonable suspicion or probable cause could also vary with the severity of the crime and could be decided through public debate.[42,43]

[41] The classifiers could be trained and tested by using datasets that are known to include examples of both criminal and law-abiding behavior.

[42] Historically, at least within the United States, it has been held that the standards for probable cause resist quantification. However, with the rise of sniffing dogs, partial fingerprint matches, facial recognition technology, and other "mechanistic" methods of establishing probable cause, it has become relatively common to take statistical evidence into account, at least in an ad hoc manner [77]. For example, if probable cause is established solely on the basis of a match made by facial recognition software, then the evidence is essentially nothing but statistical; it consists of the fact that software is making a certain classification and that the software is right a sufficiently large portion of the time. Some legal scholars argue that, in mechanistic cases, the most natural way to define probable cause is simply to define a necessary accuracy rate.

[43] Expanding the role of statistical evidence in establishing reasonable suspicion and probable cause, currently

- Finally, at least within the United States, the legal justification for using such a classifier could be similar to the justification for using sniffing dogs. In relevant Supreme Court cases, it has traditionally been held that, insofar as the dog is only capable of reporting unlawful activity, with sufficiently high accuracy, and insofar as the dog is not trespassing on its target's property, then its use does not constitute a "search" [31, 50].

It may be instructive to examine this illustrative protocol further, in order to highlight several important barriers to its implementation (or the implementation of similar privacy-preserving surveillance protocols). First, for most instances of criminal activity, it is not yet possible to develop classifiers that are sufficiently accurate. This is particularly true for crimes with very small rates of occurrence, such as terrorism. The viability of the protocol, outside of narrow contexts, then requires extremely significant progress in artificial intelligence to raise the achievable accuracy level. Taking the long view, though, it is worth noting that most experts in the field of artificial intelligence expect computer systems to become capable of completing all tasks that humans can within a century [82]. The wait for AI systems to be able to complete many individual classification tasks may be substantially shorter. In addition, a number of tasks, such as set-intersection searches, are either already automated or trivial for existing systems.

Second, as already discussed, secure multiparty computation is very expensive. The computational overhead and the need for interaction between the multiple parties could be expected, with current techniques, to increase the total cost of running any classifier by multiple orders of magnitude. This increase is not necessarily unbearable, especially given that available computing power and bandwidth have historically increased by about a factor of a hundred every decade [109, 121]. Still, every additional dollar of cost should be assumed to decrease the technical and political viability of a privacy-preserving scheme.

Third, in this scheme, there is still a need for the companies and the public to trust the auditing body to ensure the agency's honesty. Fortunately, it may be possible to loosen this constraint. As discussed in section 2.8, zk-SNARKs can be used to publicly demonstrate that a government agency is in fact applying an algorithm that has received an auditing body's approval, without revealing the details of the algorithm to the public [103]. In theory, it might even be possible to use zk-SNARKs to publicly demonstrate that the classifier achieves the desired accuracy rate without requiring trust to be placed in any auditing body at all.

Fourth, this version of the scheme requires the relevant service providers to have access to their users' data. It presumes that they do not offer end-to-end encryption of messages, or homomorphic processing of data. One more difficult alternative would be to require indi-

quite minor, might also be seen as its own end [115]. For example, a judge determining whether a police officer had probable cause to conduct a search, on the basis of their claim that a certain driver appeared suspicious, would not typically take into account the officer's track record. Intuitively, though, it is of great relevance whether the officer's judgements about suspicious drivers are correct 100% of the time or 0% of the time. An increase in the use of statistical evidence, which would be quite abundant for any classifier applied to large quantities of data, could enable more effective oversight, more precise discussions of standards, and more uniformly high-quality decisions.

viduals to share their data with a number of independent entities, through secret sharing, for the sole purpose of allowing these entities to use it for surveillance purposes. Public surveillance footage and other data not directly associated with technology companies and service providers could also be incorporated in this way.

Finally, even if all of these problems were resolved, there would still remain all of the legal, political, and social challenges that would surely be associated with radically revising the nature of a country's surveillance policies and infrastructure. There would also be the unavoidable fact that privacy-preserving surveillance can aid the enforcement of unjust laws in exactly the same way that it aids the enforcement of just laws.

In short, the feasibility of privacy-preserving mass surveillance would require large technical advances and significantly more thought than has been devoted to it so far. There may be reasons to be bullish about privacy-preserving surveillance in the long run, though. The intersection between emerging cryptographic technologies and surveillance is still a little-examined research area, and if fundamental barriers to privacy-preserving surveillance exist, then they are not obvious.

I should also emphasize that broad privacy-preserving surveillance protocols, if they become feasible, should not be assumed to be *socially beneficial*. Again, if some law is unjust, then removing privacy concerns as a barrier to its enforcement would be harmful. There are also sensible reasons to be wary of any large expansions of surveillance powers, even if the laws in question are just and even if privacy-preserving techniques will be used. Nonetheless, if highly privacy-preserving surveillance ever becomes feasible, then it is easy to imagine surveillance powers eventually expanding in many countries.

3.1.3 Non-intrusive agreement verification could become feasible

Just as methods of computing on confidential data might enable less invasive government surveillance, in the long run they could also enable less intrusive methods of verifying agreements.

The classic problem here is that nearly any agreement—such as an arms control agreement between countries—will require information to be collected to provide assurance of each party's compliance. However, the act of collecting this information can often be highly intrusive and will generally result in the verifier gaining access to information beyond the mere fact of the other party's compliance or non-compliance [6]. For example, agreements concerning chemical materials often entail inspections of private companies, which put these companies' valuable "confidential business information" at risk [100]. Similarly, the party verifying a disarmament agreement might take it as an opportunity to learn more about the weapons system being destroyed [52]. If the cost of intrusion is great enough, then an otherwise mutually beneficial agreement might become politically infeasible [125]. Even in cases where the cost of intrusion is ultimately deemed to be acceptable, like the JCPOA agreement concerning Iran's nuclear program, this cost can be a source of difficulty in negotiations, and attempts to decrease it can also decrease assurance [42].

As above, we can construct minimally intrusive tools for verification, if we assume there

will eventually be extremely significant advancements in artificial intelligence and in the efficiency of computing on confidential data. Ultimately, international agreement monitoring is just another form of surveillance. The only difference is that the subject of the surveillance is party to an international agreement.

Beyond the secure computation techniques discussed in the previous section, zero-knowledge proofs may also have their own applications to verification. For any future agreements concerning algorithms—such as algorithms applied in autonomous weapons systems, voting systems, or systems used to make decisions with relevance to human rights—zero-knowledge proofs could be a tool for demonstrating that only algorithms with approved features are being applied [103].

In addition, as mentioned in section 2.8, some applications of "physical zero-knowledge proofs" to agreement verification are already beginning to be found. Most significantly, recent papers have proposed a method of demonstrating that a country is disposing of a genuine nuclear warhead without revealing its design details [76].

It is interesting to consider what new applications of zero-knowledge proofs and physical zero-knowledge proofs might be discovered in the coming years. One important limitation, of course, is that we should not expect them to be any more powerful than traditional methods of demonstrating claims. For example, zero-knowledge proofs of *non-existence*—such as a proof that a certain prohibited material does not exist anywhere within a country's borders—should be taken as much more difficult than zero-knowledge proofs of existence.

3.1.4 It could become easier to combat forgery

As discussed in section 2.4, trusted timestamping can be used to combat forgery in a number of domains.

Timestamps put an upper bound on when a piece of data was created, providing assurance that, at the very least, it was not forged or tampered with after the fact. For example, an organization that has a strict policy of timestamping important documents can reduce a number of risks, from corrupt officials altering incriminating documents to external actors presenting forgeries as authentic. In large part, such concerns motivate the interest actors such as the Estonian and British governments have recently taken in timestamping services.

In the future, timestamping may also play an increasingly large role in helping people to distinguish between authentic photographs and videos and artificially generated ones. Progress in machine learning has recently made it possible to generate images that are almost indistinguishable from ones recorded by cameras. Similarly, although at great expense and using rather different techniques, Hollywood studios have recently gained the ability to produce nearly photorealistic renderings of long-dead actors. Taking note of these trends, some commentators have suggested that we may lose our ability to trust in the authenticity of digital media, with significant negative implications for journalists, intelligence services, and other groups relying on photo or video evidence [4, 32, 143].

While timestamps alone do not solve this problem, they can still help substantially. Consider someone recording a video and timestamping batches of frames as they do. If they include in the video a depiction of certain information they could not have possessed before a certain time—such as a newspaper held aloft or, better yet, a projection of incoming Bitcoin transactions—then this also serves to put a lower bound on when, if the video were a forgery, they could have begun the final stage of editing. If the gap between this lower bound and the upper bound provided by the timestamp is small enough, then it can be regarded as implausible that the video is a forgery.

While such techniques are fairly cumbersome, their feasibility suggests that it should remain possible to demonstrate the authenticity of at least the most important digital media.[44]

3.2 Consequences from blockchain-based technologies

3.2.1 Background: Concepts in institutional economics

Before turning to the political consequences of blockchain, it will again be useful to lay out some background concepts. In this section, I give a quick overview of certain concepts from *new institutional economics*, which explore the nature and purpose of institutions from an economic perspective.

It sometimes happens that parties would find it mutually beneficial to enter into a certain agreement with each other, but in practice fail to do so. These cases constitute *collective action problems*. A classic illustration of a collective action problem is the *tragedy of the commons* [87]. In this case, the members of a village all have access to the same grazing field. To avoid depleting the grass, the villagers will need to limit their consumption. However, for any given individual, the cost of limiting one's own consumption is not offset by the slight decrease in the rate of depletion. It follows that if everyone is left to their own devices, then the grass will be collectively depleted. Everyone would be made better off by a village-wide agreement on limited consumption, but the strong individual incentives to cheat might make such an agreement impractical.

Importantly, many of the world's most salient problems are at least in part problems of collective action. For example, the failure of the international community to properly address global problems like climate change and pandemic risk can be considered highly analogous to the tragedy of the commons [174]. Mutually harmful wars and arms races are another large-scale example. Some economists, like Bryan Caplan, have also characterized political apathy and irrationality as a sort of collective action problem; everyone would be better off if everyone else became more well-informed and epistemically rational, but the individual benefits of doing so are extremely low [43].

One way to explain the source of collective action problems is to apply the framework of

[44] Another proposed technique, in a somewhat different vein, is the use of physically tamper-proof cameras that digitally sign the images they record. The upside of this technique is that it is much easier for users to implement, while the downside is that it requires trust to be placed in the camera manufacturers and in physical methods for preventing tampering (which are relatively more ad hoc than cryptographic ones).

transaction costs [162]. The realization of each possible agreement is associated with a set of costs, which can in turn be divided into search costs, bargaining costs, and commitment costs. First, the *search costs* are associated with finding parties to enter the agreement with and learning whatever information would be necessary to identify the agreement as mutually desirable. Second, the *bargaining costs* are associated with converging on the agreement, given the available information, and perhaps formalizing it. Finally, the *commitment costs* are associated with increasing the probability that the parties will comply with the agreement and risking the possibility that they will not. These commitment costs might include the costs of monitoring the relevant parties, engaging in a potential arbitration process, or establishing credible threats, as well as the expected costs implied by possible non-compliance. In addition, in keeping with the above discussion of verification (see section 3.1.3), there are often transaction costs associated with the need to share extraneous sensitive information in order to share information that is relevant. These may be classed as either search costs or commitment costs, depending on whether they precede or follow the relevant agreements.

In the case of the tragedy of the commons, the work involved in gathering the villagers together and discussing an agreement not to overgraze would constitute search and bargaining costs. The work involved in keeping watch over the field, the work involved in punishing violators, and the value of any grass lost to undetected overgrazing would be examples of commitment costs.

If any party's share of the transaction costs associated with an otherwise mutually desirable agreement outweighs their share of its benefits, then, given that the party is rational, there will be a collective action problem.

Social institutions are critical for resolving collective action problems. They can be defined, rather abstractly, as stable patterns of behavior and expectation within a community. For example, within a country, the common tendency to accept certain pieces of paper (e.g., U.S. dollars) in exchange for goods, and the common expectation that other people will also accept these pieces of paper in exchange for goods, can be considered a social institution. Formal organizations like companies and governments can also be considered social institutions. For example, some of the "patterns" that ultimately give shape to a police department include: whose orders and which kinds of orders are treated as having force, who is let into the building and given a uniform and who is not, how people on the street respond to people in police uniforms, and so on.

The right social institutions can do a great deal to lower transaction costs [144]. Traffic rules save drivers from the cost of negotiating for the right of way, money saves businesses from arranging complex trades, regulation and reputation saves buyers from the cost of investigating products, and effective police departments save citizens from the myriad costs of ensuring mutual non-aggression and respect for property. In addition, anthropologists have found that cooperative norms tend to help small communities avoid most of the costs entailed by literal tragedies of the commons [128].[45]

[45]Typically, as communities grow in size, it becomes increasingly common to rely on institutions that are formal, hierarchical, and centralized. However, more informal, socially flat, and decentralized institutions always

One perspective on blockchain technology is that it supports the creation of new institutions, which might may either compete with existing institutions (to solve the same sorts of problems) or solve problems not solved by any existing institution.

3.2.2 The roles of banks, technology companies, voting authorities, and other traditional institutions could shrink

Systems that rely on new cryptographic technologies could begin to supplant existing social institutions, if they can offer more appealing versions of the services these institutions provide.

First, one key feature of cryptocurrencies is that they allow users to make virtual payments without relying on banks or credit card companies to act as intermediaries or on traditional central banks to manage the underlying currency. If a significant number of people find it appealing from an efficiency, liberty, or privacy standpoint to avoid relying on these institutions when possible, then their roles could shrink.

One early example of a decline in influence, already discussed, is banks' and credit companies' inability to prevent payments to WikiLeaks in 2012 when all major companies decided to refuse payments to the activist group. WikiLeaks supporters decided to simply cut banks and credit card companies out of the process and donated tens of thousands of dollars' worth of bitcoin instead [111]. In the same vein, a sign of the potential decline of central banks' influence is given by Venezuela, where a non-negligible number of citizens have begun to store some of their wealth in cryptocurrencies, rather than the hyperinflated national currency [120]. The existence of cryptocurrencies like Zcash that use zk-SNARKs to obscure how much currency each user owns may also pose problems for taxation, beyond those already posed by the ambiguous status of cryptocurrencies as taxable assets.

The past few years have also seen growing interest in *decentralized finance* (or *DeFi*) applications [61] that build on top of cryptocurrency systems. Examples include decentralized cryptocurrency exchanges (which allow users to trade cryptocurrencies with one another) and applications that support collateralized cryptocurrency loans (where the collateral is another digital asset). These applications might reduce the roles of a number of additional financial institutions.

Decentralized applications could also be used to reduce reliance on traditional technology companies to provide services [38]. For example, there are a number of proposed or early-stage decentralized applications to allow users to rent out storage space for encrypted files, using smart contracts, as an alternative to services like Google Drive [176]. If scalability issues can be resolved (see section 4.2.1), then, some writers have suggested, decentralized services provided by companies like Uber might eventually become practical [137].

Smart contracts, especially when used in conjunction with smart property, could conceivably allow people to reduce their reliance on courts and arbitrators when entering into agreements. As the legal scholar Lawrence Lessig has noted, it is sometimes possible to

retain their importance.

achieve a particular end through either legal code or computer code [105]. In this sense, smart contracts would appear to increase the space of possibilities for what can be achieved through computer code. If people can more easily and frequently enter into agreements with one another that they trust will be enforced, without needing to trust that courts will enforce them, then this would seem to diminish the importance of traditional legal systems [182].[46]

At the same time, as section 4.2.4 will discuss, smart contracts—or, at least, anything resembling existing smart contracts—appear capable of filling only a sliver of the role currently filled by traditional contracts. Generally, the potential of decentralized services might also be undermined by regulation (see section 4.2.2), security issues with the consensus protocols used in permissionless blockchains (see section 4.2.3), or the possibility that many existing institutions are "good enough" to remove most of the incentive to seek alternatives (see section 4.2.5).

In addition, there is some case to be made that emerging cryptographic technologies might also help to bolster traditional centralized institutions. In particular, as discussed in section 2.8, applications of zk-SNARKs could help to increase the accountability of these institutions. The use of technologies for computing on confidential data might also alleviate concerns about these institutions violating individuals' privacy or exploiting their personal information. The net effect could be to increase the palatability of reliance on centralized institutions and thereby decrease incentives to seek more decentralized or novel alternatives.

3.2.3 It could become possible to solve collection action problems that existing institutions cannot

Permissionless blockchains and smart contracts may be able to lower some of the "commitment costs" involved in solving collective action problems.

In certain cases, using a smart contract may remove the need to rely on a trusted third party, who might be inclined to engage in opportunism, or to take other pains to incentivize honesty. The parties engaged in a chess smart contract, to repeat an earlier example, do not need to trust the loser to hand money over to the winner, an escrow service to handle the transfer, or a court to extract compensation for dishonesty. Therefore, giving these competitors the option of using a smart contract might decrease the overall cost of ensuring that their agreement will be honored.[47]

[46] As a related framing, contract enforcement mechanisms may be either public (for example, a formal legal system) or private (for example, Mafia enforcers or reputational harm) [81]. The use of smart contracts may increase the relative prominence of private mechanisms.

[47] A secondary effect of smart contracts, in some contexts, may also be to lower bargaining costs. It may be particularly cheap to reuse sections of code from successful smart contracts, to write smart contracts that depend on the outputs of other smart contracts, and to create open-ended smart contracts that any number of parties can easily choose to engage with. As one example, an experimental smart contract-based venture capital organization, described in section 3.2.4, left room for an indefinite number of users to join it. Although it quickly failed due to a programming mistake, the computer code used to define the relevant smart contract can be reused and iterated upon by future organizations of this type. On the other hand, the opportunity to reuse pieces of previous contracts is certainly far from unique to smart contracts.

In other cases, the need for a trusted third party remains. To return to the case of a bet about the weather, a third party might be employed to input relevant weather information to the blockchain. In essence, the use of a smart contract here ensures that a certain transfer of money will occur if a certain report is made, but it does not ensure that the report will accurately reflect the world. A number of groups, like the Augur team, are working on consensus and reputation schemes designed to incentivize accurate inputs to smart contracts [130]. It is still too early, though, to judge how reliable or efficient these systems will be. Existing systems still have very few users and have not been subjected to much stress-testing. Section 4.2.4 discusses the apparent limitations of smart contracts more thoroughly, with a focus on the ways in which they can introduce new transaction costs as well.

Here, though, I will walk through a number of domains in which existing institutions struggle to solve collective action problems. In each case, I will discuss the possibility that blockchain-associated institutions will be substantially more effective.

Citizens of countries with weak institutions Weak government institutions lead many countries, especially developing countries, to suffer from particularly high transaction costs. The relevant issues typically include corruption, incompetent government officials (partly a result of corruption), inconsistent access to services, lack of robust honesty and reciprocity norms, and so on. High transaction costs associated with such issues have in turn been linked to low economic growth [122]. Notably, even if smart contracts do not offer advantages over developed world institutions, they could still offer significant improvements to available institutions outside the developed world. For example, even if cryptocurrencies are not superior to well-managed fiat currencies, citizens of countries with badly mismanaged currencies (e.g., erratic changes in the rate of inflation) may prefer to rely on cryptocurrencies in some circumstances. Here, the key point is that they do not need to place their faith in a central bank that might behave irresponsibly or incompetently. In fact, citizens of certain countries with unreliable financial institutions, especially citizens of hyperinflation-afflicted Venezuela, have taken an unusually large interest in cryptocurrencies [120].

Criminal groups Criminal groups often face barriers to interacting with institutions like courts, banks, and highly visible reputation systems. It is not possible, for example, to bring a drug dealer to court for stealing product, or to look up online reviews for hitmen. The net effect is that groups with an interest in participating in illegal activities are not nearly so organized or effective as they could be. However, smart contracts may reduce these coordination problems. One paper by Ari Juels, Ahmed Kosba, and Elaine Shi considers the possibility of "criminal smart contracts," and describes a range of schemes for using smart contracts to credibly commit to payments for data leaks, assassinations, and other crimes [98]. Some of these schemes are not yet practical with current smart contract systems. For example, the scheme for committing to payment for assassinations requires a service that inputs news to the blockchain as well as a smart contract that is capable of determining whether a timestamped plan for the assassination, revealed after the event, matches the news reports. However, some of the other schemes already appear to be relatively practical, and none appear to face fundamental barriers. If these considerations are combined

with the speculative possibility of government surveillance "going dark" (see section 3.1.1), then it is possible that some forms of illegal activity will become more prevalent in the future.

Disorganized shared-interest groups It is generally the case, within any given political system, that only a very small portion of groups with common interests successfully organize to pursue these interests. For example, private industries often organize to lobby the government, but the consumers of these industries' products do not. Influential grassroots movements have arisen around some issues, like abortion and police violence, but not others of comparable concern, like economic policy. The collection of people who would be happy to become vegetarian if it also meant everyone else did, to pay for news if it also meant everyone else did, and so on, do not form robust organizations or enter into pacts. Similarly, even though voters as a whole would have clear common interests in coordinating to become more well-informed, the possibility of such an agreement arising seems wholly implausible.

These disparities hinge on the fact that, for a sufficiently large interest group, any individual member's contribution to the group's shared aim will be insignificant. A single union member's dues are unlikely to make a difference to the union's bargaining and lobbying efforts, and a single well-informed voter is unlikely to make a difference to the quality of elected officials. If the act of participating in collective action does not bring additional benefits, and if agreements to participate are difficult to enforce, then a rational member of a large interest group will refrain from participating. In his classic book *The Logic of Collective Action*, the economist Mancur Olson summarizes this point with the thesis, "Unless the number of individuals in a group is quite small, or unless there is coercion or some other special device to make individuals act in their common interest, *rational, self-interested individuals will not act to achieve their common or group interests*" [123].

According to Olson, many industries are able to organize because they contain only a relatively small number of decision-making parties, who can direct the employees below them. Some grassroots movements are able to organize because, due to the nature of their central issue, membership offers significant social benefits, a significant personal sense of meaning, or opportunities to participate in enjoyable activities. Unions in some domains are able to maintain members due to a mixture of social benefits, control of professional resources, and opportunities for outright coercion. And so on. The result is an extremely uneven distribution of power between shared-interest groups, and extremely suboptimal outcomes for most shared-interest groups—including the group constituted by citizens of a given country as a whole.

Smart contracts, allowing individuals to credibly commit to collective action if others also do so, could prove to be a useful tool for many interest groups. Groups interested in funding lobbying efforts could use smart contracts that result in their making donations only if enough other people do, too, and perhaps scaling down the donation size as the number of people who commit grows. The most challenging problem seems to be verifying agreements to take action in the physical world. It is not immediately obvious, for example, how to credibly commit to attend a protest, become a well-informed voter, or become a vege-

tarian. Peer verification networks, where members of groups examine evidence produced by other members, seem possible. These systems could also be engineered to include reputation systems and incentives for honesty. However, each additional unit of human labor, unit of honesty-inducing incentives, and unit of complexity adds to the total transaction cost. Plausibly, the total transaction cost could often remain too high to enable collective action.

It is also not clear that blockchain would resolve the central barrier, whatever it is, that is preventing these kinds of systems from arising. It is certainly possible to imagine a centralized Kickstarter-like company that, for example, commits people to make donations to lobbying groups only if enough other people do too. If the company's bottom-line depended on the trust of its users, and if it had legal obligations to fulfill its commitments, then a reasonable person should expect the company to follow through and transfer the funds in an appropriate way. Whatever is preventing these kinds of services from arising, therefore, might be something other than the need for trusted third parties.

States A central concern in international relations is the relative inefficacy of international institutions [93]. Theorists often describe the interactions between countries as taking place under "anarchy," since there is no supreme authority capable of filling the same institutional role that states fill at the national level. Many of the greatest problems in international relations, and therefore many of the largest-scale problems in general, appear to follow from excessive commitment costs. For example, the possibility of "free-riding" makes it difficult for states to collectively agree to take costly measures to alleviate environmental issues and global risks, such as those associated with climate change. The fact that a rising power cannot credibly commit to non-aggression, and the fact that two mutually hostile nations cannot commit to halting military investment, open the door for preventative war and arms races [133]. In addition, the need to establish the credibility of one's commitments, in the absence of other mechanisms, has often been cited as at least a reason for participating in prolonged wars; the Vietnam War is a notable case. The number of lives lost, risked, and harmed because of an absence of good tools for making credible international commitments would be difficult to overestimate.[48]

Therefore, given the stakes, the possibility that smart contracts could have applications in the international sphere is very much worth investigating. We can construct at least hypothetical commitment mechanisms that take advantage of smart contracts.

For example, smart contracts might enable countries to more credibly commit to international agreements that include penalties for non-compliance. Traditionally, agreements that include financial penalties, such as the Kyoto Protocol, rely on "self-punishment" [95]. Countries that fail to comply with core aspects of the agreement must still, nevertheless, be trusted to comply with the portion that obliges them to pay out large sums of money. A number of scholars have suggested that the obvious pitfalls associated with "self-punishment" can be avoided through the use of escrow, with parties to agreements making initial deposits that are returned only if they demonstrate compliance [73, 112]. Smart

[48] At the same time, the ability to make credible commitments is far from being entirely benign [142]. Among other downsides, the ability to credibly commit to threats can increase the ease of extortion.

contracts could offer a method for implementing escrow for international agreements, without the need for a trusted third party to hold and disburse the deposits. In particular, countries might sign on to a smart contract consenting to the loss of large deposits if a quorum of other countries, a distributed oracle system, or a set of internet-connected sensors judge that they have violated the agreement; the relevant blockchain could be either a consortium blockchain held between several countries or a particularly well-established permissionless blockchain.

As another example, we might also imagine a pair of countries signing onto a smart contract that will deactivate their smart property weapons systems at the same time. The additional difficulty here, though, is the need to first verify that the weapons systems respond to the state of the blockchain in the appropriate way.

Unfortunately, to be used in such cases the relevant smart contracts would need to become exceptionally reliable. As section 4.2.1 will discuss, this may not be reasonable to expect. These potential applications of smart contracts to the international sphere are ultimately, in my view, unlikely. However, if they ever do emerge, then they would probably be the absolute most consequential applications of smart contracts.

As a final note, this discussion has a close connection to the earlier discussion of the value of non-intrusive agreement verification in the international sphere (see section 3.1.3). Non-intrusive agreement verification can be seen as another tool for lowering commitment costs, specifically the costs associated with submitting to monitoring. In the most idealized case, a state might be forced to reveal no information beyond the simple fact of its compliance. Likewise, non-intrusive inspection, more generally, can be seen as a tool for lowering search costs, in that it may reduce any disadvantages states accrue when they share information that is relevant to bargaining. It is difficult, for example, to impress upon another party the power of one's cyberweaponry without revealing details that will make these weapons less effective [106, 101]. In general, as James Fearon writes in his classic paper "Rationalist Explanations for War," "there is a trade-off between revealing information about resolve or capabilities to influence bargaining and reducing the advantages of a first strike" [65]. The existence of information asymmetries, due to incentives to maintain private information, along with commitment problems, are the two most powerful explanations for how mutually harmful wars (and other collective action problems) can arise even between rational actors.

As with smart contracts, it may not be especially likely, even in the long run, that non-intrusive inspection will significantly lessen international collective action problems. Some of the most relevant private information—such as the "resolve" of a given state in the lead-up to a potential war—does not seem to lend itself naturally to tricks with zero-knowledge proofs and secure multiparty computation. Again, though, the existing problems are of great enough magnitude that any tool with potential value deserves consideration.

3.2.4 A new variety of institutions, known as "decentralized autonomous organizations," could emerge

Some writers have speculated that smart contract technology could allow a new variety of institutions to emerge. In particular, especially within the Ethereum developer community, a large number of essays and blog posts have been written on the possibility of decentralized autonomous organizations. Although definitions vary, we will define a *decentralized autonomous organization (DAO)* as an organization, associated with its own pool of assets, in which decisions about how to use these assets are determined by smart contracts among the organization's members [36].[49]

The first and still most famous example of a DAO has been a $150 million venture capital fund, known as "the DAO," which launched on Ethereum in 2016 [132]. Investors participated by buying shares in the fund, then using their shares to vote on how to disburse the fund's common pool of cryptocurrency to proposed projects; smart contracts ensured that this money would be disbursed in accordance with the votes and that profits from the projects would be distributed appropriately among the shareholders. Unfortunately, the DAO came to an untimely end. Within a few days, a user discovered a programming mistake that allowed them to siphon the majority of funds out of the organization, and, given that smart contracts are by nature impossible to modify, the project was abruptly abandoned.

A number of later projects have also described themselves as DAOs [183]. For instance, similar to "the DAO," Moloch DAO uses a smart-contract-based system to automatically disburse cryptocurrency investments if members express enough support for an investment proposal [169]. Decisions to admit new members are also determined through voting and processed automatically. Unlike "the DAO," Moloch DAO does not attempt to make a profit. It simply disburses member-provided funds to other blockchain projects that the community believes are socially valuable and has no mechanism to collect anything in return. Smart contracts are primarily used to ensure that votes concerning membership expansion and currency disbursement are binding. A failsafe mechanism, designed in response to the collapse of "the DAO," also allows members to extract their share of the common funding pool if they strongly oppose a just-concluded decision concerning the disbursement of funds.

The few DAOs that exist today are still quite new and, in some ways, fairly rudimentary. Therefore, given this limited set of case studies, it is difficult to draw meaningful generalizations. Still, DAOs do seem to differ from most existing organizations in at least four interesting ways.

First, DAOs can possess a foundational set of features that is immutable. No party—for instance, no "leader" of a DAO—can prevent the relevant smart contracts from executing in the specified way. The case of "the DAO" demonstrates the downsides of this property, but, in the right cases, it can also be considered desirable. DAOs are especially capable

[49] Some alternative definitions are loose to the extent that permissionless blockchains themselves constitute DAOs. Under this conception, the nodes maintaining them act as their "members" and, through cryptocurrency rewards, share in the "profits" earned in the process of providing users with a "service."

of survival and resistance to change, even if there is a complete turnover of members or if future members are unanimously interested in altering some of their fundamental features. This steadfastness, even in the face of an opposed membership, is how DAOs earn the descriptor "autonomous."

Second, DAOs can be truly transnational. Since they are maintained on blockchains, with the relevant nodes most likely spread across dozens of countries, they will not have any particular physical instantiations. In addition, no state can disrupt a DAO—or, at least, its fundamental features—without doing the costly and potentially unpopular work of disrupting the blockchain itself. (See section 4.2.3 for a discussion of the possibility of such a disruption.)

Third, depending on the nature of the DAO and the blockchain it is maintained on, individuals may be able to participate under pseudonyms (typically, the hashes of their public keys). This form of pseudonymity, while not unheard of, is still rather unusual. The DAO, for example, was almost certainly the first-ever pseudonymous eight-figure venture capital fund.

Fourth, DAOs place an unusually heavy emphasis on *complete contracts*, or agreements which precisely specify the obligations of each party for every possible case [56]. Smart contracts, because they need to be expressed in code, are by nature complete. In traditional organizations, however, contracts tend to be highly incomplete. In fact, a widely held theory, associated with the field of new institutional economics, is that individuals form organizations in large part because of the impracticality of specifying complete contracts for the services they provide one another (see section 3.2.1 for a discussion of transaction costs) [180]. Particularly within organizations, interactions tend to be dominated by *relational contracts*, which are unwritten cooperative norms that resist formalization [10]. This body of theory suggests that DAOs are unlikely to be able to efficiently replicate the functionality of most organizations, unless they also rely heavily on more informal relational contracts in addition to smart contracts. Norms clearly play an important role in the functioning of Moloch DAO, for instance, since everything other than the enforcement of votes is managed without smart contracts. One plausible guess we can make about future DAOs, at least successful ones, is that they will still *mostly* be built on top of informal relational contracts. Nonetheless, even a modestly greater reliance on complete contracts might lead to interesting structural differences.

It is interesting to speculate about what future DAOs could look like, conditioning on the optimistic view that they are in fact plausible outside of a narrow range of cases. What could a DAO tech company look like? How about a DAO political party, a DAO criminal organization, or, more fancifully, a DAO state?

Some writers have made quite radical claims about the feasibility of replacing traditional political systems with DAOs. However, these claims are often ambiguous. As an example, I will briefly consider the ideas of Ralph Merkle, the inventor of cryptographic hashing, who has written a paper describing a system of government he calls "DAO democracy" [113].

Merkle's basic scheme, as described in his paper, is to implement a variant of "futarchy," a system of government first described by the economist Robin Hanson [86]. In Merkle's system, there are annual citizen satisfaction polls, and all citizens are allowed to place bets on the impact proposed sets of legislation would have on total satisfaction; the sets of legislation that this betting market indicates are most likely to succeed are implemented. To help secure the integrity of the betting markets, the bets are to be recorded on a consortium blockchain, with the voting power granted to each device being used to maintain the blockchain determined by further bets about the device's reliability. Putting aside questions of this scheme's practicality, however, it is unclear to what extent it would constitute a DAO, or whether it would truly require novel cryptographic technology. Merkle makes no suggestions about the use of smart contracts in the actual implementation of legislation, and while the use of a consortium blockchains to maintain the betting records seems like a useful way of increasing their integrity, it also seems somewhat tangential to the overall vision.

Other writers have explored the argument that blockchains could be a useful tool for experimenting with novel forms of democracy [37, 157]. Futarchy is often discussed, as is "liquid democracy," a system of representative democracy in which individuals can choose to grant anyone else the power to vote on their behalf for particular sets of decisions; this representative might in turn pass their accumulated voting power on to a representative they deem to be even more well-equipped to vote well [23]. Both of these forms of democracy must almost certainly be implemented electronically and require unified databases that are extraordinarily secure. The databases must be trusted to track, for example, active updates concerning who has the power to vote for whom. Blockchain technology, perhaps in conjunction with secure multiparty computation, is seen as significantly lowering the trust threshold for implementing these political systems and making it relatively easy for small (and potentially geographically dispersed) political organizations to trial them. If the outcomes of votes are more directly linked to tangible outcomes through smart contracts, then these organizations could, less ambiguously, constitute DAOs.

A number of writers within the blockchain community have suggested that there may arise DAOs that serve citizens' needs so sufficiently that traditional states simply wither away, or that DAO-based governance will lead people's political relationships to become almost entirely decoupled from geography. These are more extreme versions of the visions discussed in sections 3.2.2 and 3.2.3. Marcella Atzori's paper, "Blockchain Technology and Decentralized Governance: Is the State Still Necessary?" sympathetically surveys some such claims, although she ultimately (I believe correctly) finds them implausible [9].

Finally, we should note that although most current speculation about the political implications of DAOs seems to be associated with positive visions, it is also possible to imagine negative outcomes. Intuitively, criminal organizations and terror groups, which may be at risk of being "beheaded" through the arrest or killing of their leaders, would have incentive to transform into DAOs if this was feasible. As an example, we can imagine a DAO that is set up to create "criminal smart contracts" for acts of politically motivated sabotage or terrorism, using a pool of money contributed by anonymous supporters. (See section 3.2.3 for a brief discussion of criminal smart contracts.)

It must be reiterated, however, that the basic practicality of DAOs, for almost any use case at all, is still unproven. Smart contracts simply might not be very useful for governing relationships within an organization. This discussion, then, should be regarded as especially speculative. As section 4.2.4 will explore, some very significant limitations stand in the way of the most radical proposed uses of smart contracts becoming feasible.

4 Limitations and skeptical views

There are some important roadblocks that could prevent several of the technologies we have discussed from achieving widespread use or achieving transformative effects. In this section, I discuss six such roadblocks, which I judge to be particularly significant: the inefficiency of methods of computing on confidential data, the difficulty of "scaling" blockchains, the threat of regulation; the inadequacies of smart contracts, the potential insecurity of permissionless blockchains, and the possibility that existing institutions are "good enough."

Other potential roadblocks, which I do not discuss here, include the arrival of quantum computers (which will render some cryptographic schemes insecure), the "just in case" data collection practices of many companies (which might make them hesitant to adopt technologies that reduce or more precisely target their data collection), limited consumer demand for privacy technologies, and the enormous volume of electricity consumed by proof-of-work protocols (which could be made unnecessary by a successful shift to proof-of-stake protocols).

4.1 Limitations of privacy-preserving technologies

4.1.1 The inefficiency of computing on confidential data

All methods of running computations on encrypted data—including homomorphic encryption, functional encryption, and secure multiparty computation—add significant overheads compared to the time and space requirements of computing on unencrypted data.

Although the overheads of secure multiparty computation, especially, are beginning to become manageable, they may never be entirely negligible. To justify their use, the demand for greater privacy must be enough to offset the economic costs of using MPC. If the demand is not great enough, then it may never see very broad use.

More troublingly, all known schemes for fully (and somewhat) homomorphic encryption are so inefficient that, at present, it is not possible to use them for anything but extremely simple computations. This, for example, rules out the possibility of using them to replace nearly any online service with a privacy-preserving version (see section 3.1) anytime soon. If fully homomorphic encryption is to find widespread use, there will need to be either major progress in discovering more efficient schemes or a many orders-of-magnitude increase in available computing power. Still, neither of these possibilities is entirely implausible due to trends in both the efficiency of fully homomorphic encryption and the growth of computing power [117].

On the other hand, as discussed in section 2.13, it may be possible to use secure multiparty computation to replicate most of the functionality that fully homomorphic encryption can offer. The extreme inefficiency of fully homomorphic encryption could ultimately turn out to be largely irrelevant.

4.2 Limitations of blockchain-based technologies

4.2.1 The difficulty of "scaling" permissionless blockchains

As mentioned in section 2.6, the most commonly used permissionless blockchains, such as Bitcoin and Ethereum, cannot process more than a dozen or so transactions per second. In comparison, a company like Visa can process tens of thousands of transactions per second by using ordinary servers.

Low transaction speeds limit the number of users a blockchain can plausibly sustain, as well limiting the complexity of the services a blockchain can provide [54]. Almost certainly, the most optimistic visions of blockchain's potential—such as the vision that decentralized cryptocurrencies will significantly compete with fiat currencies (see section 3.2.2), the vision that decentralized applications will replace a large portion of traditional online applications (see section 3.2.2), or the vision that smart contracts will enable the creation of vast new political entities (see section 3.2.4)—can only be realized by permissionless blockchains with much higher processing speeds. Low processing speeds can also indirectly limit the security of permissionless blockchains by limiting the total value of transaction fees collected by actors maintaining them: typically, the more wealth one can collect from transaction fees, the greater the incentives to maintain a given blockchain will be.

The most fundamental source of the problem, here, is that traditional blockchains such as Bitcoin require all of the nodes maintaining the blockchain to process each transaction, rather than splitting the labor between nodes. Such blockchains do not *scale*, in the sense that adding more processors or faster processors to the network does nothing to increase the system's overall processing speed. The system cannot process transactions any faster than the slowest computer (also taking into account its network connection) that is capable of serving as a node. The need for nodes to communicate with one another and come to consensus then introduces further delays.

There are two main strategies for addressing the scaling problem. The first strategy is to implement *sharding* [60]. This involves splitting a blockchain into multiple shards, with each shard handling a subset of transactions. Then, some nodes can opt to be involved in only a single shard. The more shards there are, the lower the minimum burden is on each node.

There are some obvious challenges that any successful sharding scheme must overcome: developers must prevent successful attacks on individual shards, must allow the shards to interface with each other, and must generally ensure security and consistency despite division. A great deal of research within the blockchain community is currently devoted to overcoming these challenges.

The other strategy is to pursue *layer 2 scaling*. This involves outsourcing a large portion of necessary computations to external processors rather than requiring all of the nodes maintaining the blockchain to perform them [62]. *State channels* allow groups of users to make sequences of transactions among themselves and then only publish the final re-

sult to the blockchain. For example, if ownership of a token is transferred multiple times within a short interval, then state channels could remove the need for blockchain nodes to process the intermediate transactions. Similarly, *rollups* allow external computers to process batches of transactions "off-chain" and then record only the final state update to the blockchain. Of course, to avoid a large loss in security, there needs to be some way to verify that these kinds of external computations have been performed correctly. Succinct cryptographic proofs known as zk-SNARKs (see section 2.8) are one important tool for supporting verification. These proofs, which can be published to the blockchain, allow the nodes maintaining a blockchain to verify that some computation has been carried out correctly without needing to rerun the full computation themselves.

A number of teams are currently working to upgrade the Ethereum blockchain by developing and implementing a mixture of sharding and layer 2 scaling techniques. The most optimistic community members expect these upgrades to allow Ethereum to process tens of thousands of times more transactions per second. It remains to be seen, though, just how successful these approaches to the scaling problem will be.

There are also some projects that have pursued performance gains without addressing the fundamental scaling problem. In general, these projects have set higher minimum performance standards for participating nodes, have reduced certain inefficiencies involved in decentralized consensus protocols, or have done some combination of the two. Litecoin is an early and well-known example of a blockchain project that has achieved modest performance gains by raising the minimum performance requirements for individual nodes. To explain, blockchain protocols typically aim for a roughly constant rate of block creation and typically place a hard limit on how many transactions can be included in a single block. These two design constraints then imply a limit on the rate at which transactions can be processed. For example, because the Bitcoin blockchain gains an average of six new blocks per hour, and these blocks are each only allowed to contain a few thousand transactions, there is no way for Bitcoin to process more than a dozen or so transactions per second. Litecoin and a number of other blockchains have achieved higher processing speeds, relative to Bitcoin, partly by using larger "block sizes." However, increasing the block size has also raised the minimum performance standards that nodes must meet to profitably participate in maintaining the blockchain.[50]

A more recent and promising example of a blockchain project that eschews sharding and layer-2 solutions is Solana. Solana's designers have developed a consensus protocol that requires an unusually small amount of communication and produces agreement between the participating nodes unusually quickly [184]. The overall speed of the blockchain is therefore not much slower than the speed at which an individual node can process transactions. Due to comparatively high minimum performance standards for nodes, it is reportedly able to process tens of thousands of transactions a second. As the Solana project is still in

[50] Since there is always some cost involved in maintaining a blockchain, nodes are only likely to earn a profit if they are able to process transactions at the maximum rate that the blockchain allows. A slow node that tries to participate will only be able to process a fraction of the transactions that faster nodes can. It will therefore only be able to collect a fraction of the total transaction fees that these other nodes can. Ultimately, one should expect nodes that fall below some minimum speed threshold to drop off the network.

its "beta phase," there is some lingering uncertainty about the system's security and long-term stability. Nonetheless, at the time of writing, it is perhaps the fastest decentralized blockchain system.

One general observation is that raising the minimum performance standards for a blockchain's nodes will typically allow the blockchain to support more computationally intensive applications, but will also, typically, reduce the number of actors who are capable of helping to maintain or audit the blockchain. Projects such as Solana therefore must navigate a fairly direct trade-off between performance and centralization. This trade-off should be less sharp for blockchain projects that attempt to address the fundamental scaling problem by relying on techniques such as sharding and rollouts since these projects achieve performance gains primarily by *reducing* the amount of computation that individual nodes are asked to do.

Time will tell whether any of these strategies—or some combination of them—will be enough to enable applications that are both highly decentralized and highly performant. However, the past few years of research progress seem to support an at least moderate degree of optimism.

4.2.2 The threat of restrictive regulations

A simple way to limit the use of a technology is to regulate it.

In the case of public-key cryptography, the oldest of the technologies we have discussed here, there is a long history of countries deliberating on how best to regulate it. As recently as the 1990s, for example, law enforcement agencies in the United States were arguing that the use of encryption that they could not themselves decrypt should be made illegal [11]. For the time being, all forms of encryption are perfectly legal to use within the United States and European Union, with most other major countries placing only relatively limited restrictions on use [140]. However, it is not certain that the status quo will never change. Especially in the wake of terror attacks or other catalyzing events, it is not uncommon for law enforcement officials or politicians to re-propose restrictions on cryptography, particular the use of end-to-end encryption [44, 92]. A fairly recent development is China's decision to block the use of WhatsApp, apparently based on WhatsApp's use of end-to-end encryption [30].

In the case of more novel cryptographic technologies, such as cryptocurrencies, legal statuses are in something of a state of flux, with regulations varying substantially by country and by state [46]. Regulations tend to focus on points of contact between permissionless blockchains and the outside world—for example, on businesses that exchange traditional currency for cryptocurrency—due in large part to the fact that these blockchains are inherently difficult to interfere with and lack any discernible party that is in "control" of them [168, 102]. For instance, *know-your-customer* (*KYC*) laws, which compel cryptocurrency exchanges to record and verify the identities of their customers, have become increasingly common.

It is unclear to what extent the regulation of emerging cryptographic technologies is likely

to limit their use, especially given that blockchains themselves are so difficult to interfere with. Still, if the technologies ever become truly threatening, for instance by creating financial instability or by making it easier for terror groups to operate, then dramatic actions by major governments are not inconceivable. A simple action in this category might be banning cryptocurrency exchanges, making it difficult for individuals to purchase cryptocurrency or to "cash out" by exchanging their cryptocurrency for traditional currency. States may also directly ban the use of cryptocurrencies or the use of certain decentralized applications. Although a complete ban would probably be very difficult to enforce, it should at least be possible, for example, to prevent large domestic businesses from accepting cryptocurrency payments.[51] The most extreme suppression strategy might be to attempt to "take over" proof-of-work blockchains by directing large amounts of computing power toward cryptocurrency mining, while attempting to disrupt other mining groups. (See section 4.2.3 for a discussion of the potential dynamics of a take-over attempt.) Efforts to vigorously regulate these technologies, rather than suppress them, might also introduce various frictions or limitations that reduce their appeal.

It seems reasonable to speculate that if regulation significantly influences the use of cryptographic technologies, it will primarily limit the uses that increase the difficulty of surveillance (see section 3.1.1), lessen the influence of economic and political institutions like central banks (section 3.2.2), make it easier for threatening actors to coordinate (section 3.2.3), or enable the creation of new decentralized political actors (section 3.2.4). In short, the uses of cryptography that are most desired by the libertarian and anarchist portions of the cryptography community may also be the ones that are most difficult to achieve.

4.2.3 The potential insecurity of permissionless blockchains

As discussed in section 2.6.3, permissionless blockchains are maintained through fairly complex consensus protocols. In short, they work by allowing any user to participate in the maintenance of the blockchain, granting these users voting power over the blockchain's contents in proportion to the amount of some scarce resource they own (such as computing power), and incentivizing them to vote honestly by making it very likely that they will earn digital currency if they do (or lose digital currency if they do not).

There are two interrelated problems with these protocols: First, there may be a natural tendency for large portions of scarce resources to eventually end up in the control of a very small number of users. Second, especially for users who control large portions of scarce resources, there may be ways to earn money or achieve desirable outcomes other than by helping to maintain the blockchain honestly [119].

We will first consider the case of the Bitcoin blockchain, which is by far the most well established. Bitcoin, as a reminder, uses a proof-of-work protocol that forces users known as "miners" to demonstrate their ownership of computing power by solving resource-intensive puzzles.

[51] A handful of states, including Pakistan and Vietnam, have banned cryptocurrency ownership and use outright [153]. At the time of writing, India is also considering a complete ban [3]. So far, though, there is not much evidence for the efficacy of bans. In Vietnam, for instance, 21% of respondents in a 2021 survey reported owning or using cryptocurrencies [33]. This is the second-highest rate in the world.

As of 2021, four mining collectives collectively controlled more than 50% percent of the computing power directed at mining Bitcoin [155]. If they decided to collude to "double spend" coins, then they could produce a dishonest version of the Bitcoin blockchain, missing records of their previous expenditures, that outpaces the honest one.

Mining operations may have a natural tendency to become centralized in this way, as races to develop and buy up specially built systems for solving puzzles can quickly become prohibitively expensive for all but a few parties. The need for mining groups to insure themselves against unlucky streaks can then provide a further incentive for them to merge.

Furthermore, at least for Bitcoin, it is apparently not the case that more than half of the relevant computing power needs to be directed in a dishonest way for a dishonest version of the blockchain to win out. Researchers have identified a strategy that colluding Bitcoin miners with only 25% control could use to springboard themselves into majority control and begin to take self-enriching actions, like spending individual coins multiple times [64].

If other users become aware that a given blockchain has been subjected to an attack of this sort, as would almost certainly happen, one plausible result is that the associated cryptocurrency would have its exchange value abruptly drop. This potential fallout might be enough to keep miners from colluding, even if they would otherwise ostensibly stand to gain, since the value of their accumulated bitcoin and their investments into specialized mining hardware depend on the exchange rate for the coin.

However, this incentive-based safeguard would not necessarily be enough for parties that wish to disrupt the blockchain for reasons other than simple financial exploitation. For instance, if the government of a large country really wanted to disrupt Bitcoin, it would need to invest in the computing power necessary to gain majority control.

In such a case, if the honest parties using a blockchain come to recognize that the majority of computing power is being directed dishonestly, which should in practice be quite conspicuous, then they can create what is known as a *fork* of the blockchain [119]. This is accomplished by having a large portion of honest users update their software to disregard blocks now known to have been proposed by dishonest users, building on top of blocks further back in the chain's history. The "fork" is a new, diverging blockchain that, if socially acknowledged, can fill the role of the initial one.

However, in the event of a fork, the dishonest users could still "spawn camp" by adopting new public-key pseudonyms and using their computing power to once again take control of the newly forked blockchain. The honest users might, as a further response, update to a new software version whose mining puzzles the attackers' hardware is less suited for—but, even if this move is taken, attackers with sufficient resources could still make the blockchain unusable. Repeatedly disrupting Bitcoin would be an expensive venture—likely costing billions of dollars—but could in principle be done.[52]

[52]As a further point, a highly empowered attacker (such as a national government) might also be able to

In short, like any other system of storing data, Bitcoin is not invincible. Its security can be thought of as roughly proportional to the total computing power devoted to mining, as the greater this number is, the more money miners have to lose by tanking the blockchain and the more money another attacker would need to spend to gain enough computing power. If insufficient computing power is invested, then Bitcoin—and other blockchains using proof-of-work—will remain insufficiently reliable to use for any truly vital applications.

Proof-of-stake blockchains have similar limitations but are perhaps more promising. For them, each node's voting power is made proportional to the quantity of cryptocurrency it "deposits." Then, the option of applying cryptocurrency penalties to parties that vote dishonestly can help strengthen incentives, and the possibility of forking to completely disregard the currency previously owned by an attacking party can help to remove the possibility of "spawn camping."

For proof-of-stake systems, security is roughly proportional to the value of the relevant cryptocurrency deposits, so it can also be expected to increase as more value is tied up in the relevant blockchain. It follows that the potential security that can be offered by a given proof-of-stake blockchain is linked to its scalability—in other words, as discussed in section 4.2.1, whether it can be made to accommodate a much larger number of users and transactions. Plausibly, a sufficiently scalable proof-of-stake blockchain—like what Ethereum aims to become—could be extremely secure. However, proof-of-stake protocols are still too new for their superiority to proof-of-work protocols to be clear. The relative merits of proof-of-stake systems and proof-of-work systems are still debated within the blockchain community.

In summary, it is not yet clear *exactly* how much security permissionless blockchains can offer. If new weaknesses are discovered, or existing weakness become more salient with time, then many users may not feel comfortable using permissionless blockchains for certain purposes. For example, even once-in-a-decade failures could be enough to heavily disincentivize the use of cryptocurrencies as stores of value.

4.2.4 The inadequacies of smart contracts

As discussed in section 3.2.3, one way to conceptualize smart contracts is as a tool for reducing the "transaction costs" associated with entering into agreements.

To return once more to an illustrative example, a chess smart contract can allow two parties to agree to have whoever loses the game pay the winner, without the need to establish trust between the parties, to risk their non-compliance, or to enlist the services of a third-party enforcer. Since there is a wide range of cases where existing institutions leave transaction costs quite high—with, for example, inefficient legal systems being a common problem—it is a natural hypothesis that smart contracts will be helpful for some of these cases.

Section 3.2.3 lists, at a rather abstract level, a number of ways in which smart contracts

decrease honest miners' power by shutting down mining facilities, restricting the sale of specialized hardware, or applying other controls.

Search costs	Bargaining costs	Commitment costs
• Verifying smart contract traits • Establishing confidence in smart property • Learning about cryptographic tools	• Formulating complete contracts • Translating contracts into computer code	• Paying fees to full nodes • Paying fees to information services • Protecting against coercion and theft of digital assets • Risking failure of smart contract to function as expected

Table 5: New transaction costs associated with smart contract use

may reduce transaction costs. The section then considers the possibility that, on this basis, they may be able to solve some collective action problems that existing institutions either cannot solve or can only solve relatively inefficiently.

At the same time, it is important to note that smart contracts introduce new transaction costs of their own. These costs imply that using smart contracts may also, in many cases, be a far inferior alternative to existing institutions. I will consider search costs, bargaining costs, and then commitment costs.

Concerning search costs, there will be the cost of verifying that a smart contract conforms to the relevant parties' intentions. The importance of verification is illustrated by the failure of the multimillion-dollar venture capital fund "the DAO" (discussed in section 3.2.4). An unfortunate programming mistake in the relevant smart contracts, not noticed until it was too late, left open a loophole that allowed one user to siphon a large portion of the money out of the fund [67]. The more complex a smart contract is, the more difficult the work of detecting such flaws will be. While the techniques of *formal verification* can help in checking that a smart contract has certain mathematically well-defined properties, there will still remain the more nebulous task of checking that the potential judgements of a smart contract all conform to common sense [21]. In this vein, the fact that traditional agreements can be filtered through human interpretation, which is capable of navigating ambiguities and grasping obvious intentions, can be seen as an important cost-saving feature.

In cases where smart contracts involve smart property, there will also be the search cost of establishing trust that a piece of smart property does in fact respond to the relevant blockchain in the promised way. Conceivably, a regulatory system could be required to establish this trust (see section 4.2.2). Another simple search cost, which we might regard

as something of a fixed cost, is the cost of learning about smart contracts and how to use them. As the length of this report may help to demonstrate, this cost should be regarded as non-trivial.

Concerning bargaining costs, it may often be very difficult to develop smart contracts that are *complete*, meaning that they precisely specify the obligations of each party for every possible case. As smart contracts are computer code, designed to execute automatically, they do not leave room for ambiguity or underspecification. However, many legal scholars and economists hold that incomplete contracts are dramatically more common than complete ones [163, 89]. The reason is that, for complex transactional relationships, it can be exceptionally difficult to translate each party's obligations into something so precise as computer code while sufficiently accounting for every possible future contingency. Should the need arise for a traditional contract, ambiguities and contingencies can be effectively filled in after an initial agreement, through ad hoc renegotiation and the cooperative norms that characterize informal "relational contracts" [10]. In contrast, smart contracts place the full burden on working out the contract's details on the initial drafters. In many cases, this is likely to be impractical.[53]

We also have particular reason to doubt that most traditional organizations could be efficiently reconstructed as smart-contract-based "decentralized autonomous organizations" (see section 3.2.4). A widely held theory in economics is that hierarchical firms arise, in large part, because of the impracticality of drafting complete contracts [89, 180]. As a trivial example, a company's employees do not need to treat each idiosyncratic e-mail they send out as a service requiring formal contractual representation. The members of a DAO, at least a DAO that makes truly extensive use of smart contracts, may not have this luxury.

Then, there are commitment costs.

First, there are the costs associated with the need to maintain the integrity and availability of the underlying blockchain. These costs consist of fees for creating and interacting with smart contracts, which, at least in the case of permissionless blockchains, feed into the cryptocurrency rewards used to incentivize full nodes to store and execute these contracts (see section 2.6.2). Since all of the nodes must perform the same work, at least in traditional blockchain systems, these fees can become quite significant. For example, in any case where a traditional online service is not associated with either extremely high profit margins or significant regulatory burdens, an equivalent decentralized application will normally be much more expensive to use.

Second, in cases where users would like to make a contract conditional on features of the outside world, such as one party's success in finishing a construction project on time, there will be costs associated with ensuring that this information is recorded properly. The relevant parties may need to acquire the services of a trusted third-party arbiter or

[53]Another cause for concern comes from the long-standing field of "computational law," which has examined the possibility of translating laws and contracts into computer code and found that, while translation may be feasible within some domains (including electronic commerce), judgements often require case-based, analogical, or inductive reasoning that it is very difficult to represent with computer code [71, 147].

apply a distributed consensus protocol that uses cryptocurrency payouts to incentivize multiple parties to converge on the truth. As mentioned in section 2.10, some developers are currently trialing early versions of these consensus protocols. However, it remains to be seen how effective "distributed oracle systems" will be. In general, large expenditures could be required to create sufficient incentives for accurate inputs.

Third, there will be costs associated with the need to ensure that theft or coercion cannot be applied to counteract a smart contract. For example, the relevant parties must protect against their private keys being stolen, as well as having any of their smart property stolen and rewired. More bluntly, the parties must protect themselves against attempts to threaten them into signing unfavorable contracts. The necessary security measures may be expensive. These security concerns are also, of course, good arguments for the continued relevance of existing law enforcement institutions. The potential need to interact with these institutions adds its own costs.

Finally, there will be the costs associated with the risk that a contract does not function as expected. Despite the relevant parties' best efforts, a contract's code may contain errors, the consensus protocols used to maintain the blockchain may fail (see section 4.2.3), the external information the contract relies on may be inaccurate, the parties may experience theft or coercion, or the expected significance of the relevant digital assets may fail to hold, for example, if a cryptocurrency loses its value or if a piece of smart property does not respond as promised. Collectively, these risks may be quite significant.

Together, all these considerations suggest that, while smart contracts may help to reduce some transaction costs, they also come with very significant costs of their own. There is not yet an obvious basis for predictions that they will heavily encroach on the territory already covered by existing institutions (see section 3.2.2).

4.2.5 The possibility that existing institutions are "good enough"

The previous section explored the various inadequacies of smart contracts. A somewhat symmetrical viewpoint—which pushes against the idea that smart contracts, and decentralized applications generally, will begin filling many of the functions filled by existing institutions (see section 3.2.2)—is that many existing institutions are actually highly effective [156].

Overall, the most compelling arguments for the use of permissionless blockchain technology seems to be an argument from commitment costs (see section 3.2.3). In particular, in certain circumstances, this technology can limit the costs associated with the need to establish trust in other parties to provide services, as well as with the risk of having this trust violated.

Nevertheless, the case can be made that, in the current institutional environment, trust is often fairly easy to come by. As Vitalik Buterin writes in his essay, "The Problem of Trust," "At least in the developed world, if you put your money in a bank, it's safe.... From such a perspective, one can easily see how the traditional 'centralized system' is serving people just fine" [39].

To be sure, there are many cases where trust is not easy to come by. Section 3.2.3 discusses some of them, including cases where the relevant parties are located in countries with unusually dysfunctional institutions. It is unclear, though, just how far these cases extend, or exactly how effective blockchain systems could be for resolving them. While polls show that citizens in many countries report distrusting major political and economic institutions, their levels of trust may still, as a relative matter, be much greater than they would be for complicated and unproven cryptographic technologies [127, 88].

In his essay, Buterin acknowledges these points, but also takes a long-run perspective to caution against what he might consider excessive skepticism. First, he writes, decentralized applications could eventually establish much greater reputations for trustworthiness than they possess today:

> *Who would you really trust more: [well-vetted banks] or a group of mining firms of unknown quantity and size with no real-world reputations, 90% of whose chips may be produced in Taiwan or Shenzhen? For mainstream securities settlement, the answer that most people in the world would give seems rather clear. But then, in ten years' time, if the set of miners or the set of anonymous stakeholders of some particular currency proves itself trustworthy, eventually banks may warm up to even the more "pure cryptoanarchic" model – or they may not.*

Second, in some cases, it may be more appropriate to think of blockchain technology as cutting the costs a new institution must face in earning sufficient quantities of trust:

> *Rather than concentrating on the lack of trust, here we emphasize the barrier to entry in becoming a locus of trust. Sure, billion dollar companies can certainly become loci of trust just fine, and indeed it is the case that they generally work pretty well.... However, their ability to do so comes at a high cost.... The key promise of decentralized technology, under this viewpoint, is not to create systems that are even more trustworthy than current large institutions.... Rather, the key promise of decentralized technology is to provide a shortcut to let future application developers get there faster.... A [simple cryptographic protocol] may well have a lower probability of failure than all but the largest of institutions – and at a millionth of the cost. Blockchain-based applications allow developers to prove that they are honest – by setting up a system where they do not even have any more power than the users do.*

This consideration seems to suggest that, if decentralized applications do eventually achieve a prominence comparable to existing centralized institutions, it may not be by directly displacing them. Instead, these systems might fill voids left by institutions that suffer losses of trust, or offer future services that no institution yet provides. The rise of blockchain technology in political and economic life could be gradual, like the turnover of cells in a body.

Nevertheless, even this more moderate view is highly speculative. As the preceding sections have discussed, there remain difficult technical and legal roadblocks to large-scale blockchain use, and the utility of smart contracts is still extremely unclear. A vast gap separates the technology's present level of maturity and the level of maturity it will need

to achieve to offer plausible alternatives to most of the services provided by centralized institutions.

Since blockchain technology is only a dozen years old, and has attracted widespread attention for perhaps five years, it would appear premature to place a cap on its potential. However, it would also be premature to forecast radical visions with any degree of confidence.

References

[1] S. Aaronson, "Quantum copy-protection and quantum money," in *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE, 2009, pp. 229–242, doi:10.1109/CCC.2009.42.

[2] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-preserving data mining*, C. C. Aggarwal and S. Y. Philip, Eds. New York: Springer, 2008, pp. 11–52.

[3] A. Ahmed and N. Anand, "India to propose cryptocurrency ban, penalising miners and traders," *Reuters*, 2021. [Online]. Available: https://www.reuters.com/article/uk-india-cryptocurrency-ban/india-to-propose-cryptocurrency-ban-penalising-miners-traders-source-idUSKBN2B60QP

[4] G. Allen and T. Chan, "Artificial intelligence and national security," Belfer Center for Science and International Affairs, Harvard Kennedy School, 2017. [Online]. Available: https://www.belfercenter.org/publication/artificial-intelligence-and-national-security

[5] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," arXiv preprint arXiv:1606.06565, 2016. [Online]. Available: https://arxiv.org/abs/1606.06565

[6] J. V. Andrew J. Coe, "Why arms control is so rare," *American Political Science Review*, vol. 114, no. 2, pp. 342–355, 2020, doi:10.1017/S000305541900073X.

[7] J. Andrews, "An introduction to AZTEC," 2019. [Online]. Available: https://medium.com/aztec-protocol/an-introduction-to-aztec-47c70e875dc7

[8] L. J. Aslett, P. M. Esperança, and C. C. Holmes, "A review of homomorphic encryption and software tools for encrypted statistical machine learning," 2015, arXiv preprint arXiv:1508.06574. [Online]. Available: https://arxiv.org/abs/1508.06574

[9] M. Atzori, "Blockchain technology and decentralized governance: Is the state still necessary?" 2015, doi:10.2139/ssrn.2709713.

[10] G. Baker, R. Gibbons, and K. J. Murphy, "Relational contracts and the theory of the firm," *The Quarterly Journal of Economics*, vol. 117, no. 1, pp. 39–84, 2002, doi:10.1162/003355302753399445.

[11] D. Banisar, "Stopping science: The case of cryptography," *Health Matrix*, vol. 9, no. 2, pp. 253–287, 1999. [Online]. Available: https://scholarlycommons.law.case.edu/healthmatrix/vol9/iss2/4/

[12] B. Barrett, "The year encryption won," *Wired*, 2016. [Online]. Available:

https://www.wired.com/2016/12/year-encryption-won/

[13] C. P. Bauer, *Secret history: The story of cryptology*. Boca Raton, FL: CRC Press, 2013.

[14] C. Baum, I. Damgård, and C. Orlandi, "Publicly auditable secure multi-party computation," in *International Conference on Security and Cryptography for Networks*. Springer, 2014, pp. 175–196, doi:10.1007/978-3-319-10879-7_11.

[15] BBC Editors, "What are NFTs and why are some worth millions?" *BBC News*, 2021. [Online]. Available: https://www.bbc.com/news/technology-56371912

[16] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway, "Everything provable is provable in zero-knowledge," in *Advances in Cryptology–CRYPTO 1988*. Springer-Verlag, 1990, pp. 37–56, doi:10.1007/0-387-34799-2_4.

[17] E. Ben-Sasson, "Zerocash, bitcoin, and transparent computational integrity," Jan 2017.

[18] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "SNARKs for C: Verifying program executions succinctly and in zero knowledge," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 90–108, doi:10.1007/978-3-642-40084-1_6.

[19] E. Ben-Sasson, A. Chiesa, M. Green, E. Tromer, and M. Virza, "Secure sampling of public parameters for succinct zero knowledge proofs," in *2015 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2015, pp. 287–304, doi:10.1109/SP.2015.25.

[20] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 142–157, doi:10.1007/978-3-662-53357-4_10.

[21] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. Zanella-Béguelin, "Formal verification of smart contracts: Short paper," in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*. ACM, 2016, pp. 91–96, doi:10.1145/2993600.2993611.

[22] P. Birnstill, S. Bretthauer, S. Greiner, and E. Krempel, "Privacy-preserving surveillance: An interdisciplinary approach," *International Data Privacy Law*, vol. 5, no. 4, pp. 298–308, 2015, doi:10.1093/idpl/ipv021.

[23] C. Blum and C. I. Zuber, "Liquid democracy: Potentials, problems, and perspectives," *Journal of Political Philosophy*, vol. 24, no. 2, pp. 162–182, 2015, doi:10.1111/jopp.12065.

[24] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*.

ACM, 1988, pp. 103–112, doi:10.1145/62212.62222.

[25] D. Bogdanov, M. Jöemets, S. Siim, and M. Vaht, "How the Estonian tax and custom board evaluated a tax fraud detection system based on secure multi-party computation," in *International Conference on Financial Cryptography and Data Security*, 2015, pp. 227–234, doi:10.1007/978-3-662-47854-7_14.

[26] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. P. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft, "Secure multiparty computation goes live," in *Financial Cryptography and Data Security*. Springer, 2009, pp. 325–343, doi:10.1007/978-3-642-03549-4_20.

[27] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography*. Springer, 2011, pp. 253–273, doi:10.1007/978-3-642-19571-6_16.

[28] N. Bostrom, *Superintelligence: Paths, dangers, strategies*. Oxford: Oxford University Press, 2014.

[29] N. Bostrom, A. Dafoe, and C. Flynn, "Policy desiderata in the development of machine superintelligence." [Online]. Available: https://www.fhi.ox.ac.uk/wp-content/uploads/Policy-Desiderata-in-the-Development-of-Machine-Superintelligence.pdf

[30] K. Bradsher, "China blocks whatsapp, broadening online censorship," *The New York Times*, 2017. [Online]. Available: https://www.nytimes.com/2017/09/25/business/china-whatsapp-blocked.html

[31] I. Braverman, "Passing the sniff test: Police dogs as surveillance technology," *Buffalo Law Review*, vol. 61, no. 1, pp. 81–167, 2013. [Online]. Available: https://digitalcommons.law.buffalo.edu/journal_articles/336

[32] M. Brundage, S. Avin *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," 2017. [Online]. Available: https://maliciousaireport.com/

[33] K. Buchholz, "How common is crypto?" *Statista*, 2021. [Online]. Available: https://www.statista.com/chart/18345/crypto-currency-adoption/

[34] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Master's thesis, The University of Guelph, 2016. [Online]. Available: http://hdl.handle.net/10214/9769

[35] R. Burn-Callander, "Skype inventor Jaan Tallinn wants to use Bitcoin technology to save the world," *The Telegraph*, Jun 2016. [Online]. Available: http://www.telegraph.co.uk/business/2016/06/20/skype-inventor-jaan-tallinn-wants-to-use-bitcoin-technology-to-s/

[36] V. Buterin, "DAOs, DACs, DAs and more: An incomplete terminology guide," Ethereum Blog, May 2014. [Online]. Available: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/

[37] ——, "An introduction to futarchy," Aug 2014. [Online]. Available: https://blog.ethereum.org/2014/08/21/introduction-futarchy/

[38] ——, "Visions, part 1: The value of blockchain technology," Ethereum Blog, Apr 2015. [Online]. Available: https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/

[39] ——, "Visions, part 2: The problem of trust," Ethereum Blog, Apr 2015. [Online]. Available: https://blog.ethereum.org/2015/04/27/visions-part-2-the-problem-of-trust/

[40] ——, "Why cryptoeconomics and x-risk researchers should listen to each other more," 2016. [Online]. Available: https://medium.com/@VitalikButerin/why-cryptoeconomics-and-x-risk-researchers-should-listen-to-each-other-more-a2db72b3e86b

[41] ——, "Ethereum: A next-generation smart contract and decentralized application platform," 2017. [Online]. Available: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[42] M. Calabresi, "Iran nuclear deal: Verification is trickiest part," *Time*, 2015. [Online]. Available: http://time.com/3946860/iran-nuclear-deal-inspections/

[43] B. Caplan, "The logic of collective belief," *Rationality and Society*, vol. 15, no. 2, pp. 218–242, 2003, doi:10.1177/1043463103015002003.

[44] N. Cardozo, "The state of crypto law: 2016 in review," Electronic Frontier Foundation, Jan 2017. [Online]. Available: https://www.eff.org/deeplinks/2016/12/crypto-state-law-end-2016

[45] H. Chen, W. Chung, J. Jie Xu, G. Wang, Y. Qin, and M. Chau, "Crime data mining: A general framework and some examples," *Computer*, vol. 37, no. 4, pp. 50–56, 2004, doi:10.1109/MC.2004.1297301.

[46] U. W. Chohan, "Assessing the differences in bitcoin & other cryptocurrency legality across national jurisdictions," Critical Blockchain Research Initiative, 2017, 10.2139/ssrn.3042248.

[47] A. Choudhury, J. Loftus, E. Orsini, A. Patra, and N. P. Smart, "Between a rock and a hard place: Interpolating between MPC and FHE," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2013, pp. 221–240, doi:10.1007/978-3-642-42045-0_12.

[48] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi:10.1109/ACCESS.2016.2566339.

[49] J. S. Chung, A. Jamaludin, and A. Zisserman, "You said that?" 2017, arXiv preprint arXiv:1705.02966. [Online]. Available: https://arxiv.org/abs/1705.02966

[50] M. C. Clutter, "Dogs, drones, and defendants: The fourth amendment in the digital age," *George Mason Law Review*, vol. 21, pp. 557–571, 2013.

[51] J. B. Comey, "Going dark: Are technology, privacy, and public safety on a collision course?" Speech, Federal Bureau of Investigation, 2014. [Online]. Available: https://www.fbi.gov/news/speeches/going-dark-are-technology-privacy-and-public-safety-on-a-collision-course

[52] C. Comley, M. Comley, P. Eggins, G. George, S. Holloway, M. Ley, P. Thompson, and K. Warburton, "Confidence, security and verification: The challenge of global nuclear weapons arms control," Atomic Weapons Establishment, Tech. Rep. AWE/TR/2000/001, 2000. [Online]. Available: http://fissilematerials.org/library/awe00.pdf

[53] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *European Transactions on Telecommunications*, vol. 8, no. 5, pp. 481–490, 1997, doi:10.1002/ett.4460080506.

[54] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125, doi:10.1007/978-3-662-53357-4_8.

[55] S. A. Crosby and D. S. Wallach, "Efficient data structures for tamper-evident logging," in *USENIX Security Symposium*, 2009, pp. 317–334. [Online]. Available: https://www.usenix.org/legacy/event/sec09/tech/full_papers/crosby.pdf

[56] S. Davidson, P. De Filippi, and J. Potts, "Economics of blockchain," 2016, doi:10.2139/ssrn.2744751.

[57] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976, doi:10.1109/TIT.1976.1055638.

[58] G. Eder, "Digital transformation: Blockchain and land titles," in *2019 OECD Global Anti-Corruption & Integrity Forum*, 2019. [Online]. Available: https://www.oecd.org/corruption/integrity-forum/academic-papers/Georg%20Eder-%20Blockchain%20-%20Ghana_verified.pdf

[59] K. Ermoshina, F. Musiani, and H. Halpin, "End-to-end encrypted messaging protocols: An overview," in *International Conference on Internet Science*. Springer, 2016,

pp. 244–254, doi:10.1007/978-3-319-45982-0_22.

[60] "Sharding FAQ," Ethereum Wiki, 2017. [Online]. Available: https://eth.wiki/sharding/Sharding-FAQs

[61] "Decentralized finance (DeFi)," Ethereum.org, 2021. [Online]. Available: https://ethereum.org/en/defi/

[62] "Layer 2 scaling," Ethereum.org, 2021. [Online]. Available: https://ethereum.org/nb/developers/docs/layer-2-scaling/

[63] "Non-fungible tokens (NFT)," Ethereum.org, 2021. [Online]. Available: https://ethereum.org/en/nft/

[64] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 436–454, doi:10.1007/978-3-662-45472-5_28.

[65] J. D. Fearon, "Rationalist explanations for war," *International organization*, vol. 49, no. 3, pp. 379–414, 1995, doi:10.1017/S0020818300033324.

[66] J. Feigenbaum and B. Ford, "Multiple objectives of lawful-surveillance protocols," in *Cambridge International Workshop on Security Protocols*, 2017, doi:10.1007/978-3-319-71075-4_1.

[67] K. Finley, "A $50 million hack just showed that the DAO was all too human," *Wired*, Jun 2016. [Online]. Available: https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/

[68] B. Fisch, D. Freund, and M. Naor, "Physical zero-knowledge proofs of physical properties," in *Advances in Crypotology–CRYPTO 2014*. Springer, 2014, pp. 313–336, doi:10.1007/978-3-662-44381-1_18.

[69] K. B. Frikken and M. J. Atallah, "Privacy preserving electronic surveillance," in *Proceedings of the 2003 ACM workshop on Privacy in the Electronic Society*. ACM, 2003, pp. 45–52, doi:10.1145/1005140.1005148.

[70] U. Gasser, N. Gertner, J. L. Goldsmith, S. Landau, J. S. Nye, D. O'Brien, M. G. Olsen, D. Renan, J. Sanchez, B. Schneider, L. Schwartzol, and J. L. Zittrain, "Don't panic: Making progress on the "going dark" debate," Berkman Center for Internet & Society at Harvard Law School, 2016. [Online]. Available: http://nrs.harvard.edu/urn-3:HUL.InstRepos:28552576

[71] M. Genesereth, "Computational law: The cop in the backseat," CodeX: The Center for Legal Informatics, Stanford University, 2015. [Online]. Available: http://logic.stanford.edu/publications/genesereth/complaw.pdf

[72] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of*

the Forty-First Annual ACM symposium on Theory of Computing, 2009, pp. 169–178, doi:10.1145/1536414.1536440.

[73] A. Gerber and P. C. Wichardt, "Providing public goods in the absence of strong institutions," *Journal of Public Economics*, vol. 93, no. 3-4, pp. 429–439, 2009, doi:10.1016/j.jpubeco.2008.10.006.

[74] D. Ghupta, "Practical and deployable secure multi-party computation," Ph.D. dissertation, Yale University, 2016. [Online]. Available: https://www.cs.yale.edu/homes/jf/Debayan-thesis.pdf

[75] S. Gibbs, "EU seeks to outlaw 'backdoors' in new data privacy proposals," *The Guardian*, Jun 2017. [Online]. Available: https://www.theguardian.com/technology/2017/jun/19/eu-outlaw-backdoors-new-data-privacy-proposals-uk-government-encrypted-communications-whatsapp

[76] A. Glaser, B. Barak, and R. J. Goldston, "A zero-knowledge protocol for nuclear warhead verification," *Nature*, vol. 510, no. 7506, pp. 497–502, 2014, doi:10.1038/nature13457.

[77] E. Goldberg, "Getting beyond intuition in the probable cause inquiry," *Lewis & Clark Law Review*, vol. 17, no. 3, pp. 789–838, 2013. [Online]. Available: https://law.lclark.edu/live/files/15322-lcb173art3goldbergpdf

[78] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich, "Reusable garbled circuits and succinct functional encryption," in *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*. ACM, 2013, pp. 555–564, doi:10.1145/2488608.2488678.

[79] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich, "How to run Turing machines on encrypted data," in *Advances in Cryptology–CRYPTO 2013*. Springer-Verlag, 2013, pp. 536–553, doi:10.1007/978-3-642-40084-1_30.

[80] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989, doi:10.1137/0218012.

[81] H. R. Gow, D. H. Streeter, and J. F. Swinnen, "How private contract enforcement mechanisms can succeed where public institutions," *Agricultural Economics*, vol. 23, no. 3, pp. 253–265, 2000, doi:10.1111/j.1574-0862.2000.tb00277.x.

[82] K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans, "When will AI exceed human performance? Evidence from AI experts," arXiv preprint arXiv:1705.08807, 2017. [Online]. Available: https://arxiv.org/abs/1705.08807

[83] A. Greenberg, "Zcash, an untraceable bitcoin alternative, launches in alpha," *Wired*, Jun 2017. [Online]. Available: https://www.wired.com/2016/01/zcash-an-

untraceable-bitcoin-alternative-launches-in-alpha/

[84] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Advances in Cryptology–CRYPTO 1990*. Springer, 1990, pp. 437–455, doi:10.1007/3-540-38424-3_32.

[85] B. H. Hall and B. Khan, "Adoption of new technology," Working Paper 9730, National Bureau of Economic Research, 2003, doi:10.3386/w9730.

[86] R. Hanson, "Shall we vote on values, but bet on beliefs?" *Journal of Political Philosophy*, vol. 21, pp. 151–178, 2003, doi:10.1111/jopp.12008.

[87] G. Hardin, "The tragedy of the commons," *Journal of Natural Resources Policy Research*, vol. 1, no. 3, pp. 243–253, 2009, doi:10.1080/19390450903037302.

[88] M. Harrington, "Survey: People's trust has declined in business, media, government, and NGOs," Jan 2017. [Online]. Available: https://hbr.org/2017/01/survey-peoples-trust-has-declined-in-business-media-government-and-ngos

[89] O. D. Hart, "Incomplete contracts and the theory of the firm," *Journal of Law, Economics, & Organization*, vol. 4, no. 1, pp. 119–139, 1988, doi:10.1093/oxfordjournals.jleo.a036940.

[90] A. Hern, "Revolv devices bricked as Google's Nest shuts down smart home company," *The Guardian*, Apr 2016. [Online]. Available: https://www.theguardian.com/technology/2016/apr/05/revolv-devices-bricked-google-nest-smart-home

[91] ——, "Google's DeepMind plans bitcoin-style health record tracking for hospitals," *The Guardian*, Mar 2017. [Online]. Available: https://www.theguardian.com/technology/2017/mar/09/google-deepmind-health-records-tracking-blockchain-nhs-hospitals

[92] ——, "UK government can force encryption removal, but fears losing, experts say," *The Guardian*, Mar 2017. [Online]. Available: https://www.theguardian.com/technology/2017/mar/29/uk-government-encryption-whatsapp-investigatory-powers-act

[93] R. Higgott, "International political institutions," in *The Oxford handbook of Political Institutions*, S. A. Binder, R. A. W. Rhodes, and B. A. Rockman, Eds. Oxford: Oxford University Press, 2006, doi:10.1093/oxfordhb/9780199548460.003.0031.

[94] F. H. Hinsley, "The influence of ULTRA in the Second World War," Seminar Transcript, 1993. [Online]. Available: http://www.cdpa.co.uk/UoP/HoC/Lectures/HoC_08e.PDF

[95] J. Hovi, M. Greaker, C. Hagem, and B. Holtsmark, "A credible compliance enforcement system for the climate regime," *Climate Policy*, vol. 12, no. 6, pp. 741–754, 2012, doi:10.1080/14693062.2012.692206.

[96] A. Irrera and J. Kelly, "Blockchain could save investment banks up to $12 billion a year: Accenture," *Reuters*, Jan 2017. [Online]. Available: http://uk.reuters.com/article/us-banks-blockchain-accenture/blockchain-could-save-investment-banks-up-to-12-billion-a-year-accenture-idUKKBN1511OU

[97] ITU-T, "Distributed ledger technologies and financial inclusion," Focus Group Technical Report, International Telecommunications Union, 2017. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/dfs/Documents/201703/ITU_FGDFS_Report-on-DLT-and-Financial-Inclusion.pdf

[98] A. Juels, A. Kosba, and E. Shi, "The Ring of Gyges: Investigating the future of criminal smart contracts," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 283–295, doi:10.1145/2976749.2978362.

[99] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL: CRC Press, 2014.

[100] B. Kellman, D. S. Gualtieri, and E. A. Tanzman, "Disarmament and disclosure: How arms control verification can proceed without threatening confidential business information," *Harvard International Law Journal*, vol. 36, pp. 71–126, 1995.

[101] L. Kello, *The Virtual Weapon and International Order*. New Haven, CT: Yale University Press, 2017.

[102] T. I. Kiviat, "Beyond bitcoin: Issues in regulating blockchain tranactions," *Duke LJ*, vol. 65, pp. 569–608, 2015. [Online]. Available: https://scholarship.law.duke.edu/dlj/vol65/iss3/4/

[103] J. A. Kroll, J. Huey, S. Barocas, E. W. Felten, J. R. Reidenberg, D. G. Robinson, and H. Yu, "Accountable algorithms," *University of Pennsylvania Law Review*, vol. 165, no. 3, pp. 633–705, 2016. [Online]. Available: https://www.pennlawreview.com/2017/02/23/accountable-algorithms/

[104] E. Kuo, B. Iles, and M. R. Cruz, "Ampleforth: A new synthetic commodity," Ampleforth, Tech. Rep., 2018. [Online]. Available: https://www.ampleforth.org/papers/

[105] L. Lessig, *Code and Other Laws of Cyberspace*. New York: Basic Books, 1999.

[106] M. C. Libicki, *Cyberdeterrence and Cyberwar*. Santa Monica, CA: Rand Corporation, 2009. [Online]. Available: https://www.rand.org/pubs/monographs/MG877.html

[107] "White paper," Libra Association, 2019. [Online]. Available: https://www.diem.com/en-us/white-paper/

[108] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009,

doi:10.29012/jpc.v1i1.566.

[109] C. A. Mack, "Fifty years of Moore's Law," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 2, pp. 202–207, 2011, doi:10.1109/TSM.2010.2096437.

[110] A. Martin and I. Martinovic, "Security and privacy impacts of a unique personal identifier," University of Oxford Cyber Studies Programme, 2016. [Online]. Available: https://ora.ox.ac.uk/objects/uuid:90cf14a1-beb3-4322-b18d-deffe8c7f861

[111] J. Matonis, "WikiLeaks bypasses financial blockade with bitcoin," Aug 2012. [Online]. Available: https://www.forbes.com/sites/jonmatonis/2012/08/20/wikileaks-bypasses-financial-blockade-with-bitcoin

[112] D. M. McEvoy, "Enforcing compliance with international environmental agreements using a deposit-refund system," *International Environmental Agreements: Politics, Law and Economics*, vol. 13, no. 4, pp. 481–496, 2013, doi:10.1007/s10784-013-9209-2.

[113] R. Merkle, "DAOs, democracy and governance," *Cryonics Magazine*, vol. 37, no. 4, pp. 28–40, 2016. [Online]. Available: https://www.alcor.org/docs/cryonics-magazine-2016-04.pdf#page=28

[114] R. C. Merkle, "Secrecy, authentication, and public key systems," Information Systems Laboratory, Stanford University, Tech. Rep., 1979.

[115] M. Minzner, "Putting probability back into probable cause," *Texas Law Review*, vol. 87, no. 5, pp. 913–962, 2009. [Online]. Available: https://digitalrepository.unm.edu/law_facultyscholarship/478/

[116] M. Möser, K. Soska, E. Heilman, H. Lee, Kevin Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin, "An empirical analysis of linkability in the Monero blockchain," arXiv preprint arXiv:1704.04299, 2017. [Online]. Available: https://arxiv.org/abs/1704.04299

[117] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud Computing Security Workshop*. ACM, 2011, pp. 113–124, doi:10.1145/2046660.2046682.

[118] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[119] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ: Princeton University Press, 2016.

[120] D. Nguyen, "Venezuela's bitcoin story puts it in a category of one," NASDAQ, 2020. [Online]. Available: https://www.nasdaq.com/articles/venezuelas-bitcoin-story-puts-it-in-a-category-of-one-2020-11-11

[121] J. Nielsen, "Nielsens law of internet bandwidth," Nielsen Norman Group, 1998. [Online]. Available: https://www.nngroup.com/articles/law-of-bandwidth/

[122] D. C. North, "Institutions, transaction costs and economic growth," *Economic Inquiry*, vol. 25, no. 3, pp. 419–428, 1987, doi:10.1111/j.1465-7295.1987.tb00750.x.

[123] M. Olson, *The logic of collective action: Public Goods and the Theory of Groups*. Cambridge, MA: Harvard University Press, 2009.

[124] A. O'Neill, "Definitional issues in functional encryption," IACR Cryptology ePrint Archive, Report 2010/556, p. 556, 2010. [Online]. Available: https://eprint.iacr.org/2010/556

[125] P. O'Neill, *Verification in an Age of Insecurity: The Future of Arms Control Compliance*. Oxford: Oxford University Press, 2009.

[126] OpenMined, "Openmined welcome package," GitHub, 2021. [Online]. Available: https://github.com/OpenMined/OM-Welcome-Package

[127] E. Ortiz-Ospina and M. Roser, "Trust," Our World in Data. [Online]. Available: https://ourworldindata.org/trust

[128] E. Ostrom, J. Burger, C. B. Field, R. B. Norgaard, and D. Policansky, "Revisiting the commons: Local lessons, global challenges," *Science*, vol. 284, no. 5412, pp. 278–282, 1999, doi:10.1126/science.284.5412.278.

[129] A. Pasternack, "Police body cameras will do more than just record you," *Fast Company*, Mar 2017. [Online]. Available: https://www.fastcompany.com/3061935/police-body-cameras-livestreaming-face-recognition-and-ai

[130] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, "Augur: A decentralized oracle and prediction market platform (v2.0)," arXiv preprint arXiv:1501.01042, Forecast Foundation, 2019. [Online]. Available: https://arxiv.org/abs/1501.01042

[131] N. Popper, *Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money*. New York: HarperCollins, 2016.

[132] ——, "A hacking of more than 50 million dashes hopes in the world of virtual currency," *The New York Times*, Jun 2016. [Online]. Available: https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html

[133] R. Powell, "War as a commitment problem," *International Organization*, vol. 60, no. 1, pp. 169–203, 2006, doi:10.1017/S0020818306060061.

[134] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social networks*, Y. Altshuler, Y. Elovici, A. B. Cremers, N. Aharony, and

A. Pentland, Eds. New York: Springer, 2013, pp. 197–223, doi:10.1007/978-1-4614-4139-7_10.

[135] C. Reitwiessner, "An update on integrating Zcash on Ethereum (ZoE)," Ethereum Blog, Jan 2017. [Online]. Available: https://blog.ethereum.org/2017/01/19/update-integrating-zcash-ethereum/

[136] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978, doi:10.1145/359340.359342.

[137] S. Rosenberg, "Can an arcane crypto ledger replace Uber, Spotify and AirBnB?" *Wired*, Jan 2016. [Online]. Available: https://www.wired.com/2016/01/can-an-arcane-crypto-ledger-replace-uber-spotify-and-airbnb/

[138] J. Rowlatt, "How Bitcoin's vast energy use could burst its bubble," *BBC News*, 2021. [Online]. Available: https://www.bbc.co.uk/news/science-environment-56215787

[139] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*. ACM, 2014, pp. 475–484, doi:10.1145/2591796.2591825.

[140] N. Saper, "International cryptography regulation and the global information economy," *Northwestern Journal of Technology and Intellectual Property*, vol. 11, no. 7, pp. 673–688, 2013. [Online]. Available: https://scholarlycommons.law.northwestern.edu/njtip/vol11/iss7/5

[141] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474, doi:10.1109/SP.2014.36.

[142] T. C. Schelling, *The Strategy of Conflict*. Cambridge, MA: Harvard University Press, 1980.

[143] N. Schick, *Deep Fakes and the Infocalypse: What You Urgently Need to Know*. London: Hachette UK, 2020.

[144] A. Schotter, *The Economic Theory of Social Institutions*. Cambridge, UK: Cambridge University Press, 1981.

[145] A. Segal, "Design and implementation of privacy-preserving surveillance," Ph.D. dissertation, Yale University, 2016. [Online]. Available: https://search.proquest.com/openview/781bcd9c8443aaa28a63b061e81a74e4/

[146] A. Segal, B. Ford, and J. Feigenbaum, "Catching bandits and only bandits: Privacy-preserving intersection warrants for lawful surveillance," in *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI 14)*, 2014. [Online]. Available: https://www.usenix.org/conference/foci14/workshop-program/presentation/segal

[147] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory, "The British Nationality Act as a logic program," *Communications of the ACM*, vol. 29, no. 5, pp. 370–386, 1986, doi:10.1145/5689.5920.

[148] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979, doi:10.1145/359168.359176.

[149] E. Silfversten, M. Favaro, L. Slapakova, S. Ishikawa, J. Liu, and A. Salas, "Exploring the use of Zcash cryptocurrency for illicit or criminal purposes," RAND, 2020. [Online]. Available: https://www.rand.org/pubs/research_reports/RR4418.html

[150] G. Singh, "FIFE: A framework for investigating functional encryption," Master's thesis, Massachusetts Institute of Technology, 2016. [Online]. Available: https://dspace.mit.edu/handle/1721.1/106375

[151] N. P. Smart, *Cryptography Made Simple*. Cham, Switzerland: Springer, 2016.

[152] D. J. Solove, *Nothing to Hide: The False Tradeoff between Privacy and Security*. New Haven: Yale University Press, 2011.

[153] Staff of Global Legal Research Directorate, "Regulation of cryptocurrency around the world," The Law Library of Congress, Tech. Rep., 2018. [Online]. Available: https://www.loc.gov/law/help/cryptocurrency/world-survey.php

[154] J. Stark, "Making sense of cryptoeconomics," CoinDesk, Aug 2017. [Online]. Available: https://www.coindesk.com/making-sense-cryptoeconomics/

[155] "Distribution of Bitcoin's network hashrate in the last 24 hours until February 17, 2021," Statista, 2021. [Online]. Available: https://www.statista.com/statistics/731416/market-share-of-mining-pools/

[156] K. Stinchcombe, "Ten years in, nobody has come up with a use for blockchain," *Hackernoon*, 2017. [Online]. Available: https://medium.com/@kaistinchcombe/ten-years-in-nobody-has-come-up-with-a-use-case-for-blockchain-ee98c180100

[157] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA: O'Reilly, 2015.

[158] J. Swearingen, "Something weird (or weirder than normal) is happening at WikiLeaks," *New York*, Nov 2016. [Online]. Available: http://nymag.com/selectall/2016/11/wikileak-hashes-dont-match-so-whats-going-on.html

[159] N. Szabo, "The idea of smart contracts," 1997. [Online]. Available: https://nakamotoinstitute.org/the-idea-of-smart-contracts/

[160] R. Talviste, "Applying secure multi-party computation in practice," Ph.D. dissertation, University of Tartu, 2016. [Online]. Available: https://core.ac.uk/download/pdf/144708931.pdf

[161] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang, "Privacy loss in Apple's implementation of differential privacy on MacOS 10.12," arXiv preprint arXiv:1709.02753, 2017. [Online]. Available: https://arxiv.org/abs/1709.02753

[162] M. Taylor and S. Singleton, "The communal resource: Transaction costs and the solution of collective action problems," *Politics & Society*, vol. 21, no. 2, pp. 195–214, 1993, doi:10.1177/0032329293021002004.

[163] J. Tirole, "Incomplete contracts: Where do we stand?" *Econometrica*, vol. 67, no. 4, pp. 741–781, 1999, doi:10.1111/1468-0262.00052.

[164] P. Todd, "Cypherpunk desert bus: My role in the 2016 Zcash trusted setup ceremony," Nov 2016. [Online]. Available: https://petertodd.org/2016/cypherpunk-desert-bus-zcash-trusted-setup-ceremony

[165] A. Trask, "Building safe A.I.: A tutorial for encrypted deep learning," March 2017. [Online]. Available: https://iamtrask.github.io/2017/03/17/safe-ai/

[166] ——, "Safe crime detection: Homomorphic encryption and deep learning for more effective, less intrusive digital surveillance," Jun 2017. [Online]. Available: https://iamtrask.github.io/2017/06/05/homomorphic-surveillance/

[167] A. Trask, E. Bluemke, B. Garfinkel, C. G. Cuervas-Mons, and A. Dafoe, "Beyond privacy trade-offs with structured transparency," arXiv preprint arXiv:2012.08347, 2020. [Online]. Available: https://arxiv.org/abs/2012.08347

[168] M. Tsukerman, "The block is hot: A survey of the state of Bitcoin regulation and suggestions for the future," *Berkeley Technology Law Journal*, vol. 30, no. Annual Review, pp. 1127–1170, 2015. [Online]. Available: https://btlj.org/data/articles2015/vol30/30_AR/1127-1170_Tsukerman_Final%20111915.pdf

[169] C. Turley, "Moloch evolved: V2 primer," *Raid Guild*, 2020. [Online]. Available: https://medium.com/raid-guild/moloch-evolved-v2-primer-25c9cdeab455

[170] M. Une, "The security evaluation of time stamping schemes: The present situation and studies," IMES Discussion Papers Series 2001-E-18, Institute for Monetary and Economic Studies, Bank of Japan, 2001. [Online]. Available: https://www.imes.boj.or.jp/research/papers/english/01-E-18.pdf

[171] V. Vaikuntanathan, "Computing blindfolded: New developments in fully homomorphic encryption," in *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2011, pp. 5–16, doi:10.1109/FOCS.2011.98.

[172] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: Scalable private messaging resistant to traffic analysis," in *Proceedings of the 25th Symposium on Operating Systems Principles*. ACM, 2015, pp. 137–152, doi:10.1145/2815400.2815417.

[173] P. Vigna and M. J. Casey, "Bitcoin for the unbanked," *Foreign Affairs*, 2015. [Online].

Available: https://www.foreignaffairs.com/articles/2015-02-26/bitcoin-unbanked

[174] J. Vogler, *The Global Commons: Environmental and Technological Governance*. West Sussex, England: Wiley, 2000.

[175] G. Volpicelli, "How the blockchain is helping stop the spread of conflict diamonds," *Wired UK*, Feb 2017. [Online]. Available: http://www.wired.co.uk/article/blockchain-conflict-diamonds-everledger

[176] D. Vorick and L. Champine, "Sia: Simple decentralized storage," 2014. [Online]. Available: https://sia.tech/sia.pdf

[177] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin, "FALCON: Honest-majority maliciously secure framework for private deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 1, pp. 188–208, 2021, doi:10.2478/popets-2021-0011.

[178] M. Walfish and A. J. Blumberg, "Verifying computations without reexecuting them," *Communications of the ACM*, vol. 58, no. 2, pp. 74–84, 2015, doi:10.1145/2641562.

[179] M. Walport, "Distributed ledger technology: beyond block chain," United Kingdom Government Office for Science, 2015. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf

[180] O. E. Williamson, *The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting*. New York: Macmillan, 1985.

[181] B. Wittes and G. Blum, *The Future of Violence: Robots and Germs, Hackers and Drones–Confronting A New Age of Threat*. New York: Basic Books, 2015.

[182] A. Wright and P. De Filippi, "Decentralized blockchain technology and the rise of lex cryptographia," 2015, doi:10.2139/ssrn.2580664.

[183] L. Xie, "A beginner's guide to DAOs," Mirror, 2021. [Online]. Available: https://linda.mirror.xyz/Vh8K4leCGEO06_qSGx-vS5lvgUqhqkCz9ut81WwCP2o

[184] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0.8.14," Solana Labs, 2018.

[185] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science*. IEEE, 1982, pp. 160–164, doi:10.1109/SFCS.1982.38.

[186] "What are zk-SNARKs?" Zcash. [Online]. Available: https://z.cash/technology/zksnarks.html

[187] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," arXiv preprint arXiv:1506.03471, 2015. [Online].

Available: https://arxiv.org/abs/1506.03471

A Relevance of progress in artificial intelligence

The significance of technological developments in one field often depends on what developments occur in other fields. For example, the significance of last century's developments in cryptography would not have been nearly so great if it hadn't been for the creation of the internet and other novel communications technologies.

To take into account these interaction effects, this section will briefly consider some of the ways in which progress in artificial intelligence and progress in cryptography could be relevant to one another.

I list seven potential interaction points.

A.1 AI systems may enable more effective surveillance

Progress in AI could enable more effective surveillance by decreasing the cost of extracting information from collected data [45, 4]. It is plausible, for example, that AI systems applied to videos, messages, and social networks could become fairly effective at automatically identifying criminals, dissidents, or other groups that state actors would have an interest in discovering, without the need for humans to sift through the relevant data by hand. However, as discussed in section 3.1.1, this trend might be undermined by developments in cryptography—specifically, by a move toward greater and more effective use of end-to-end encryption, cryptocurrencies capable of obscuring transaction details, and methods of computing on confidential data which reduce the need for companies to collect personal information.

A.2 AI systems may help to make privacy-preserving surveillance feasible

If AI systems find greater applications in surveillance, then this could also open the door for surveillance that offers a greater degree of privacy.

As discussed in section 3.1.2, one way to achieve this result is to apply techniques for computing on confidential data. These techniques could make it possible to extract security-relevant information from surveillance data without having access to it in unencrypted form.

In addition, AI systems could also enable privacy-preserving surveillance more directly. The point, here, is that encryption is only one way to prevent users from gaining access to extraneous information in surveillance data. It is also possible to obfuscate data in a more direct manner, for example by removing labels or, in differential privacy techniques, by adding random alterations [2]. Similarly, AI systems may be able to perform more precise censorship of collected data. An early example of this idea is the proposed use of face-blurring algorithms for police body-camera footage [129].

A.3 AI systems may increase the need for anti-forgery schemes

Progress in AI continues to make fake photographs and videos more convincing and cheap to produce [49]. If this trend continues, it could become increasingly difficult to distinguish true claims from false ones, with important negative consequences for politics, law enforcement, and news reporting [4]. Schemes of the sort described in section 3.1.4, which use trusted timestamping to provide evidence for the veracity of images, could be important tools for avoiding these consequences.

A.4 Methods of computing on confidential data could reduce barriers to developing certain AI systems

Developing AI systems using machine learning methods typically requires access to large volumes of relevant data. Confidentiality concerns are therefore one natural barrier to AI development in certain domains. For example, for good reason, it is typically rather difficult to gain access to large volumes of medical data, even if this data could be used in a socially beneficial fashion.

Methods of computing on confidential data (see sections 2.11–2.13), particularly secure multiparty computation techniques, could reduce confidentiality concerns as a barrier to AI development. While other techniques in privacy-preserving machine learning, such as differential privacy, do exist, they are less generally applicable and suffer from significant trade-offs between how effective they are at preserving privacy and how much they still allow machines to learn [2].

One illustrative project here is OpenMined, which aims to make it much easier for users to contribute their data to machine learning projects that are conducted in a privacy-preserving fashion [126].

A.5 The problems of safe AI design and safe smart contract design may be connected

One broad problem in the emerging field of "AI safety" is the problem of designing AI systems that will not exhibit unintended harmful behaviors [5, 4]. For instance, there is not yet any general method for ensuring that an AI system trained or programmed to behave well in a limited set of environments will not cause accidents if it is deployed in a wider range of real-world environments. We can already point to examples of AI accidents such as the 2010 "flash crash," in which the behavior of automated trading systems caused a trillion-dollar stock market crash, and fatal collisions that have occurred with self-driving cars. In the future, as AI systems are used to automate increasingly complex and crucial tasks, techniques for avoiding accidents could become much more important [28].

Similarly, as discussed in section 4.2.4, one important factor restricting the applications of smart contracts is the need to ensure that they will behave as intended. The problem here is made especially severe by the fact that smart contracts cannot be modified once created. The infamous collapse of the $150 million DAO venture capital fund, described in section 3.2.4, is a good illustration of the need to get smart contracts right [132].

With such incidents in mind, Vitalik Buterin has written that the problem of designing reliable AI systems and the problem of designing reliable smart contracts overlap, and that researchers working on each of these problems could benefit from talking to those working on the other [40].

As a point of disanalogy, however, it is important to note that current "scalability" constraints (see section 4.2.1) severely limit the amount of computing power that smart contracts can draw from, to the extent that it is non-trivial to implement something even so simple as a smart contract that judges who has won a game of chess. This means that smart contracts are quite distinct from the advanced AI systems that AI safety researchers primarily have in mind. Specifically, this also means that, barring extremely large increases in available computing power, no non-trivial AI system could actually be run as a smart contract.[54]

A.6 New coordination and verification mechanisms may be useful for governing AI systems

Generally, if progress in AI creates new security challenges—for example, by enabling autonomous weapons systems, more damaging categories of cyberweapons, or other systems associated with substantial accident risks—then there could be a need for international agreements and other forms of global governance to guide its application and development [90, 29, 32]. If smart contracts ever become sufficiently reliable, then it is possible—although, for reasons discussed in section 4.2.3, probably not very likely—that they could have applications in enforcing such agreements. Similarly, it is possible that zero-knowledge proofs or methods for computing on confidential data could make it easier to verify compliance without requiring the relevant parties to share too much sensitive information (see section 3.1.3).

A.7 Fully homomorphic encryption may have applications in AI safety and security

As described in an essay by Andrew Trask, fully homomorphic encryption could make it possible to train AI systems using encrypted data or encrypted virtual environments [165]. The result of such training would be systems that cannot interact with the world, as they are only capable of processing encrypted inputs and providing encrypted outputs. Plausibly, such systems would offer more security against attempts at theft and premature real-world use (e.g., before safety properties are guaranteed). The idea is that the system would only be able to interact with specific encrypted digital environments, which might still be useful for testing or further training, until the system is itself decrypted. Note, again, that the future practicality of such a scheme would depend on the amount of computing power required for fully homomorphic encryption, as well as the amount of computing power available (see section 4.1.1).

[54] At the same time, if smart contracts that "outsource" computations to other users become more common (see section 4.2.1), then the problems of AI safety and smart contract safety could become more concretely intertwined. If the relevant techniques become sufficiently efficient and reliable, then there could arise smart contracts that pay others to run more powerful AI systems, so that the systems are likely to continue running so long as the smart contract does.

May 2021

Centre for the Governance of AI,
Future of Humanity Institute,
University of Oxford

www.governance.ai