Recent Developments in Cryptography and Potential Long-Term Consequences

Ben Garfinkel*

September 29, 2017

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 feasible to use the internet for transactions 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 the invention of blockchain-based technology (such as cryptocurrencies and smart contracts) and the discovery of schemes for fully homomorphic encryption (which allow arbitrary computations to be run on encrypted data). I provide an introduction to these technologies that assumes no previous knowledge of cryptography. Then, I consider eight speculative predictions about the long-term consequences these emerging technologies could have. These predictions include the views that a growing number of information channels used to conduct surveillance may "go dark," that algorithmically enforced "smart contracts" may begin to take the place of contracts enforced within traditional legal systems, that it may become easier to verify compliance with agreements without invasive monitoring, and that new political entities known as "decentralized autonomous organizations" may emerge. I include a further section discussing the relationship between these predictions and some predictions that have been made about the consequences of progress in artificial intelligence. Finally, I close by discussing some challenges that could 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.[1]

*Future of Humanity Institute, University of Oxford

[1]Thank you to Jeff Coleman, Jaan Tallinn, Anish Mohammed, Luke Muehlhauser, Carrick Flynn, Allan Dafoe, and Oge Nnadi for comments on various drafts of this document. As the impetus for writing this document was a December 2016 workshop on blockchain technology held at the Future of Humanity Institute, I would also like to thank the workshop participants not already listed: Stuart Armstrong, Shahar Avin, Mihály Bárász, Nick Bostrom, Miles Brundage, Vitalik Buterin, Owen Cotton-Barratt, Wei Dai, Owain Evans, Virgil Griffith, Georgios Piliouras, Anders Sandberg, and Vlad Zamfir.

Contents

1 Introduction

Across many disciplines, it can be useful for researchers to take technological progress into account.

For example, if one studies trends in employment, warfare, or politics, then it may matter greatly what the future of automation holds [20, 93, 111].

In this regard, some areas of technology are likely to have greater, broader, and more immediate significance than others. It may be for good reason, for example, that many more general audience articles have been written on automation than on superconductors. However, it is possible that the significance of some areas of technology is widely underappreciated.

One such area may be *cryptography*, or the study of techniques for encoding, protecting, and authenticating data. Arguably, progress in cryptography (and the complementary field, *cryptanalysis*, which focuses on decoding) has been one of the most consequential developments of the past century [10]. The most famous case may be Germany's use of then-advanced encryption during the Second World War and the corresponding British cryptanalysis effort, which at least one historian has estimated sped up victory by two to four years [64]. 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 impossible 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 or so, a number of new cryptographic technologies have begun to emerge (see Box 1). These technologies include blockchains, which allow for the creation of reliable and continually-updated records that no single party controls; cryptocurrencies, such as bitcoin, which allows users to make transactions without relying on financial institutions or traditional fiat currencies; smart contracts, which allow users to enter into agreements that are enforced by algorithms; and fully homomorphic encryption, which allows users to process data without having access to it.[2] In addition, there have recently been a number of proposals for novel or underutilized applications of existing technologies, such as the prevention of video forgery.

It remains to be seen whether these recent developments in cryptography will

[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 [95]. However, since no standard terminology has yet been adopted, I will continue to use the term "cryptography" as a catch-all.

be as significant as those that came before. However, a number of radical claims have been made about their importance. I list just a few examples: In a report, the United Kingdom's Government Office for Science has described blockchains as the first significant innovation in record-keeping since ancient times [110]. Ralph Merkle, one of the founding figures of modern cryptography, has written a paper arguing that blockchains will enable novel forms of democracy [74]. Jaan Tallinn, co-founder of Skype and the Future of Life Institute, has advocated for the use of smart contract technology to alleviate global coordination problems [21]. Elsewhere, researchers have argued that cryptocurrencies could make it much more difficult for governments to control or trace the flow of money, while others have written that homomorphic encryption could enable novel forms of surveillance that require less infringement of individuals' privacy [55, 102].

Unfortunately, discussions of such claims have often played out in scattered blog posts, technical reports, and whitepapers, and have therefore remained mostly inaccessible to the wider audience they may be relevant to. This essay is intended to be a contribution to the project of gathering and clarifying these discussions.[3]

In particular, this essay is divided into four sections: First, I provide an introduction to some recent developments in cryptography, aiming to include only slightly more than the minimum level of detail needed to discuss the relevant technologies clearly. Second, I describe several proposed consequences of these technologies' development, with an emphasis on consequences that would plausibly have large-scale political significance. Third, I consider the relationship between these proposed consequences and some proposed consequences of progress in artificial intelligence. Finally, I explain in greater detail some of the limitations that could prevent recent developments in cryptography from ultimately achieving great significance.

As this essay is designed to be somewhat modular, with individual sections referring to other sections where appropriate, readers should feel comfortable skipping between sections, to the extent that some individual topics are of greater interest than others. In addition, readers already familiar with cryptography may wish to only skim the initial technical introductions.

[3]It is worth emphasizing, as well, that I am not writing as an expert in the field of cryptography. My primary goal is to make existing discussions accessible, rather than to produce novel contributions.

2 Cryptographic technologies: definitions, explanations, and examples

As the introduction likely indicates, the set of cryptographic technologies that we will be considering is highly diverse. Of interest in this essay will be:

- Public-key cryptography
- Digital signatures
- Cryptographic hash functions
- Trusted timestamping
- Tamper-evident logs
- Blockchains
- Cryptocurrencies
- Zero-knowledge proofs and zk-SNARKs
- Smart property
- Smart contracts
- Fully homomorphic encryption

Some of these technologies are novel, having been developed primarily in just the last ten years. The first five entries are older, but either serve as core components of these newer technologies or continue to find new 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, Table 1 provides a much more abbreviated summary. In addition, for a deeper look, I can also recommend some sources for further reading.

Readers interested in more thorough or technical descriptions of well-established technologies, like public-key cryptography, can find them in any of a number of widely used textbooks, such as *Modern Cryptography* by Jonathan Katz and Yehuda Lindell [68].

Readers interested in blockchains, cryptocurrencies, and (to a lesser extent) smart property and smart contracts can find discussions of them in the textbook *Bitcoin and Cryptocurrency Technologies*, by Arvind Narayana et al. [77]. Note, however, that this book is somewhat outdated, and places its primary emphasis on Bitcoin (rather than cryptocurrencies and blockchains more generally). A few general-readership books have also been written on the same technologies, including Melanie Swan's *Blockchain: Blueprint for a New Economy* and Don and Alex Tapscott's *Blockchain Revolution* [96, 99]. However, books in this category tend to focus on cataloguing proposed applications, providing much less complete descriptions of what, precisely, the technologies are. There also

exist a handful of other books aimed primarily at developers or investors.

There are also many good blog posts on various aspects of blockchain technology, with posts written by Vitalik Buterin, an influential blockchain developer, standing out as particularly valuable .

So far as I am aware, there are not yet any books that thoroughly cover homomorphic encryption or zk-SNARKs, although there are a number of papers and blog posts that aim to serve as introductions. For homomorphic encryption, I recommend the paper "Computing Blindfolded: New Developments in Fully Homomorphic Encryption" [105]. I have not found any extremely clear introduction to zk-SNARKs that does not also assume significant familiarity with the field, but the Ethereum blog post "zkSNARKs in a nutshell" may be the best [85].

Finally, it is very important to note that the list of technologies I have chosen to investigate is not exhaustive. Among areas I have excluded, the most important are likely the subfield of *quantum cryptography*, which applies quantum phenomena to cryptographic tasks; the subfield of *secure multi-party computation*, which focuses on enabling multiple parties to compute joint functions of their own privately held data; and the topic of *functional encryption*, which allows unencrypted functions to be computed on encrypted data. These exclusions have been primarily a matter of limited space, although many of the most interesting technologies associated quantum cryptography (such as *quantum money* and *quantum copy-protection*) also stand out as particularly far from seeing practical applications.

2.1 Public-key encryption

Public-key encryption is a technology that allows users, identified by digital pseudonyms known as public keys, to communicate through code without sharing any secret information ahead of time [68].

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 email 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, or to translate it into a code that only she and Bob can understand.

The oldest class of encryption schemes, known as *symmetric key* schemes, have been used for thousands of years. These schemes rely on a single shared piece of information, known as a *key*, and a mutually understood rule for translating plain text and coded text into one another for a given key. For example, in

Technology	Origin	Functional description
Public-key encryption	1973	Allows users to communicate through code without sharing secret information ahead of time
Digital signatures	1979	Allow users to demonstrate that they are the senders of particular messages and that the messages have not been modified by others
Cryptographic hash functions	1979	Allow users to encode data as short strings that can't be mapped back to the original data, with diverse applications
Trusted timestamping	1991	Allows users to demonstrate that a given piece of data existed, in unmodified form, at a given time
Tamper-evident logs	1979 (ambiguous)	Collections of records that are difficult to edit without leaving a trace
Blockchains	2008	Tamper-evident logs maintained across multiple devices, with these devices managing disagreements about new entries through consensus
Permissionless blockchains	2008	Blockchains that anyone may participate in maintaining, with economic incentives for users to maintain them properly
Consortium blockchains	2012 (ambiguous)	Blockchains that multiple trusted parties (but not anyone) may participate in maintaining
Cryptocurrencies	2008	Digital currencies whose users' balances are recorded with permissionless blockchains; also used to incentivize proper maintenance of these blockchains
Zero-knowledge proofs	1985	Allow users to prove mathematical statements to others without conveying any additional information
zk-SNARKs	2010	Allow users to do the above succinctly and without back-and-forth interactions
Smart property	2015 (ambiguous)	Property whose ownership is electronically controlled by records in a permissionless or consortium blockchain
Smart contracts	2008 (ambiguous)[4]	Computer code, recorded in a permissionless or consortium blockchain, that can be used to specify under what conditions certain transactions will occur between users
Homomorphic encryption	1973	Allows users to perform certain computations on encrypted data, such that the outputs are also encrypted
Fully homomorphic encryption	2009	Allows users to perform arbitrary computations on encrypted data, such that the outputs are also encrypted

Table 1: Summary of cryptographic technologies

the simple "Caesar cipher" the key was a short number, X, and the rule for translating plain text to coded text was to move each individual letter forward by X places in the alphabet.[5]

The trouble with private 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.

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 further technical details need not concern us, these are the defining traits of a public-key system:

- Each party in the system has the ability 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 serve as digital pseudonyms, and the parties may or may not link these pseudonyms to other aspects of their identities.
- There is an algorithm that can take a plain-text message and the recipient's public key as inputs and then produce a coded message as an output. There is another algorithm that that can take a coded 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.[6]
- 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 substantial threat by government agencies such as, within the United States, the NSA and the FBI [8]. These agencies argued that, without the ability to read intercepted messages, they would be much less able to counter criminals, terrorists, and other adversarial actors. They pursued several strategies to either slow the technology's adoption or illegalize variants

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

[6] 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.

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 [88]. However, until recently, the vast majority of messaging services chose to use (or unintentionally used) methods of encryption that still allowed the the service provider, and therefore the government, to access its users' messages. Partially as a reaction to the 2013 Snowden leaks, this state of affairs has begun to change, and it has become much more common for services to offer (or claim to offer) secure "end-to-end encryption," which in practice means public-key encryption that does not grant the service provider special viewing privileges [40]. Over the course of just 2016, the number of end-to-end encryption users may have increased by over one billion, due largely to WhatsApp's and decision to begin enabling the feature by default, as well as Facebook messenger's decision to offer it as an optional feature [9].

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 increase the odds of this outcome.[7] Agencies in a number of Western countries now allege that the information channels they rely on are increasingly "going dark," although the extent of the loss is still controversial [46].

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 [68].

The way digital signatures work is that each party in a system agrees on a *signing function* and a *signature-verifying function*. The signing function takes a party's private key and a piece of data and outputs a code known as a *signature*. The signature verifying function 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.[8]

[7] 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 computer, 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..

[8] 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, I can send you a message consisting of the word "HELLO" followed by a signature. If you know my public key,

The use of digital signatures is currently ubiquitous online. For example, when you connect to Amazon.com, 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 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) [71]. Estonia is the most notable case, with its citizens having issued hundreds of millions of signatures since the program's inception.

While digital signatures date back to 1979, we will see that they play an essential essential role in other emerging cryptographic technologies, such as cryptocurrencies (to be discussed in section 3.8)

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*) [68]. The function has the properties that:

- 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 information that will produce a given hash, or to find two pieces of information that produce identical hashes.

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

Arguably, hashes are primarily important as building block for other cryptographic technologies, including blockchains. Before turning to blockchains, we will also discuss two more fundamental applications of hashes: *trusted timestamping* and *tamper-evident databases*.

---

then you can apply the verifying function to "HELLO," Y, and the signature, and thereby see that I am the one who signed the message.

2.4 Trusted timestamping

One interesting application area of cryptographic hash functions is in *trusted timestamping*, or techniques for demonstrating that a given piece of data existed, in unmodified form, at a given time [56, 104]. 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 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 activist organization Wikileaks will sometimes post hashes of sensitive documents that they obtain to Twitter [97]. 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.

Note that publishing hashes to a website like Twitter is almost certainly the crudest technique for trusted timestamping.[9] More sophisticated techniques normally entrust a Time Stamping Authority (TSA) to digitally sign a hash of the relevant data, along with a message stating the time at which the TSA received it; the user can then produce this signature and their unhashed data later on, proving the data's creation time without requiring the TSA to store it. Other techniques, which have existed since the 1990s, replace the single TSA with multiple trusted parties.

In the future, as section 3.2 will discuss, there is reason to think that trusted timestamping may come to be both more important and more reliable.

On the first point, as progress in machine learning allows for the increasingly efficient and effective forgery photographs, videos, and other such data, a number of writers have worried that there may be politically destabilizing effects [2]. Trusted timestamping can help to mitigate the risks. If a photograph claims to capture a public figure on a certain day, evidence for its veracity can be provided by a trusted timestamp corresponding to that day. Further techniques might also be used in conjunction with trusted timestamping to put a lower bound on the photograph's creation and make its veracity nearly certain.

On the second point, blockchains, a recently developed technology, can further

[9] It is so crude, in fact, that it may not qualify as an instance of "trusted timestamping" under less expansive definitions of the term.

the diminish the need to rely on a single TSA—instead harnessing a vast network of users with strong economic incentives to record the hash's publication date accurately [50].

2.5 Tamper-evident logs

A *tamper-evident log* is a collection of digital records (or *transactions*) that is designed to make alterations to it easily detectable [38].

Techniques for ensuring tamper-evidence are extremely valuable, insofar as record-keeping plays an essential role in political and economic life. If you own any money that is not currently in your wallet, for example, this is only because your bank has kept accurate records of deposists and transfers. Medical records, vote counts, surveillance data, and so on are all stored in databases somewhere.

While there are diverse methods for ensuring the property of tamper-evidence, and the integrity of stored data more generally, we will describe one particularly simple method that is often used in the context of blockchains (see Figure 1) [77].

Consider a chronological log that is receiving a steady supply of new records, with old records permanently stored. This log can be subdivided into chronological *blocks*, each containing the appropriate subset of the records and a *header*. Each header contains both a hash generated from the records in the block and the hash of the previous block's header (known as a *hash pointer*).[10] Every time sufficiently many new records are added, they are collected into a new block, given a header, and added to the chain of blocks.

Now, suppose that someone edits an individual record in the log. If they do so, then the hash in the corresponding block's header will no longer match, as is easy to verify. If the attacker edits the block header to fix this problem, though, then the hash pointer in the next block's header will now be mismatched—and so on. In short, so long as the verifier remembers the hash pointer of some block, then they will be able to detect any alterations to records that precede that block.

The stricter property of *tamper-resistance*—increasing the cost of modification, rather than just reducing the cost to detect modification—can further be achieved by requiring the headers to include information that is costly to

[10] In the most basic case, the hash "generated from the records in the block" could simply be the hash of these records. However, as a technical sidenote, a designer would normally choose to make the hash the "Merkle root" of the records—generated by repeatedly hashes pairs of records, then pairs of hashes, in a tree structure, until only a single hash is left at the top of the tree. There are efficiency-related reasons for this choice.

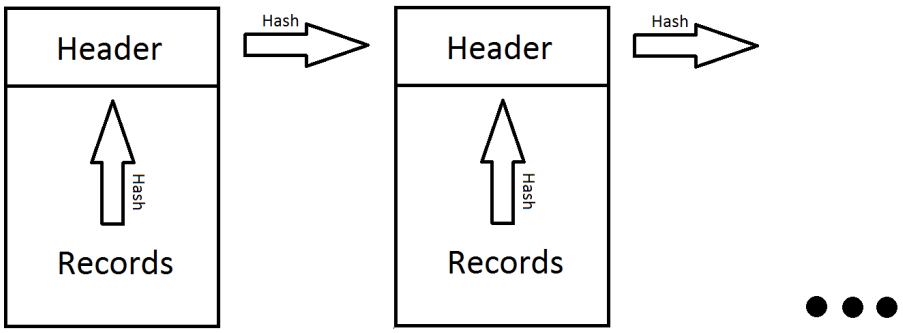

Figure 1: A simple tamper-evident log. Each block consists of a set of records and a header, and each header contains a hash generated from the corresponding records, as well as the hash of the previous block's header. Any edit to an individual record will revealed by its inconsistency with subsequent hashes.

generate. For example, each header can be required to include the solution to some brute-force computational puzzle that is generated on the basis of the previous block's header. Now, if an attacker wants to go back and edit a previous record without leaving the database clearly inconsistent—even if the verifier hasn't maintained any information about the database—then the attacker will need to spend a large amount of time and electricity solving the necessary puzzles.

Finally, if timestamps are included in the block headers, for such a log, then the log also constitutes a system for *linked timestamping* [104]. Such a system bolsters the integrity of the timestamping process. While trust in the times given by the timestamps is still based on the trust in the parties maintaining the log, trust in the order in which the timestamps were generated is based on cryptography.

2.6 Blockchains 1: Overview

In the context of this document, a *blockchain* is a tamper-evident log such that (a) there is an explicit set of standards for adding new records to the log and (b) the log is maintained across multiple devices, with these devices managing any disagreements about new records through *distributed consensus.*[11][12]

[11] We might also add, to apply common terminology, that the log must be associated with a "state" that is implied by the complete history of records. For example, if the records represent digital currency transactions between some set of users, then the state might be the ultimate quantity of the digital currency that each user owns. In a sense, the state consists of whatever core information the participants maintaining the blockchain ultimately want to agree on.

[12] The proper definition of a "blockchain" is in fact rather controversial. A large number of distinct definitions have been put forward, varying primarily in how inclusive and abstract they

Functionally, the purpose of a blockchain is not just to to prevent tampering with previously logged records, but also to ensure that *new* transactions will be will be logged in accordance with a promised set of standards.

Normally, this sort of assurance requires some trusted third party—someone who promises to maintain the log properly—and perhaps some rather complex auditing procedure. For example, the assurance you have that your bank will keep a proper record of the money in your account follows mainly from your trust in the honesty and competence of the banks that store this information. Similarly, the assurances you have about the integrity of your tax, criminal, and property records depend on your trust in various government bodies.

The innovation associated with blockchains, then, is to replace reliance on trusted third parties (TPPs) to maintain records properly with reliance on the consensus of a large network of devices, known as *nodes*, where each node might be owned by a different party [77]. In some cases, this consensus may be much more reliable than any individual third party could be. At the abstract level, then, we can say that blockchains offer two main (somewhat entangled) categories of opportunities.

First:

> **Opportunity 1:** Some records that would otherwise need to be entrusted to a (potentially unreliable or costly) third party can now be maintained without needing to trust or pay this party.

The significance of this opportunity might be measured primarily in an increase in the integrity and accessibility of certain important records and a decline in the power of institutions, like banks, technology companies, and certain government bodies, that currently play important roles as trusted record-keepers.

Second:

> **Opportunity 2:** Some records that otherwise could not be maintained together—because no willing third party could rationally be entrusted with all of them—can now be integrated and made automatically self-consistent.

The significance of this opportunity might be measured primarily in an increase in the efficiency of record-keeping—insofar as it is costly to integrate or reconcile information contained in separately-maintained records—or in new insights

---

are. For example, a more inclusive and concrete definition, used in the textbook *Bitcoin and Cryptocurrency Technologies*, describes a blockchain as essentially just a tamper-resistant log of the sort described in the previous section. Other definitions take a more abstract approach and emphasize consensus—to the extent that any technology that enables multiple parties or devices to come to reliable consensus about some evolving information "state" may qualify as a "blockchain."

or applications enabled by larger, more integrated, or more consistent sets of data.

The original blockchain, Bitcoin, was designed in 2008, and might be seen as primarily an example of the first sort of opportunity [76]. The central purpose of Bitcoin has been to record transactions in a newly-invented digital currency ("bitcoins"), with the complete log of these transactions also implying how much of the currency each users owns. Bitcoin, then, appears to be the first example of digital records of wealth maintained without a reliance on trusted third parties like banks or technology companies. It also appears to be the first example of a widely available digital payment service that does not require users to pay fees to a profit-seeking middleman.

Since Bitcoin, there have been a proliferation of blockchains, which fall into a few main categories, as summarized by the following tree [24]:

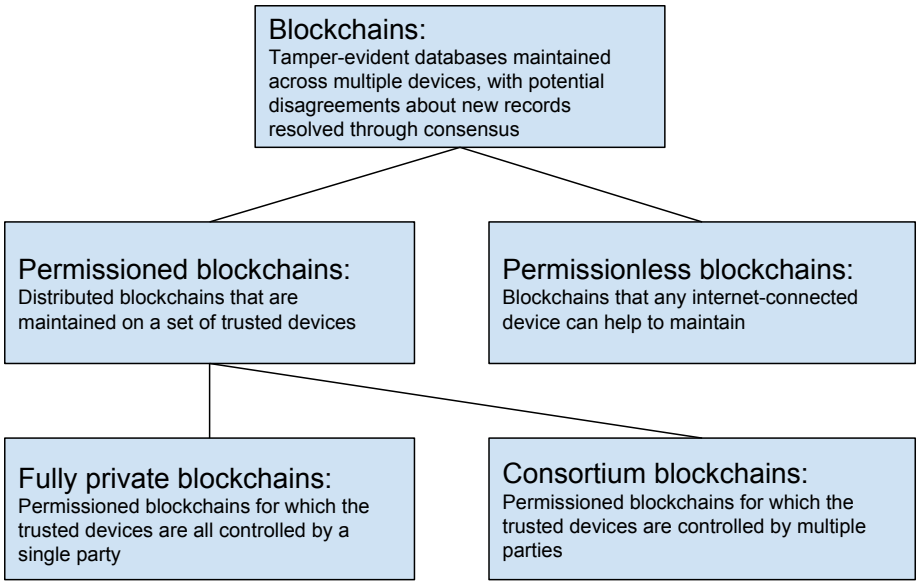

Figure 2: A taxonomy of blockchains

Bitcoin is a permissionless blockchain, and permissionless blockchains have tended to receive the bulk of media coverage. However, both varieties are significant. The question of which innovation is more significant, then, is something of a matter of who you ask. The developer community and politically motivated users have tended to focus on permissionless blockchains, and have perhaps been most actively excited by opportunities of Type 1. Meanwhile, over the past few years, large institutions like banks and governments have become increasingly interested in permissioned blockchains, due largely to opportunities

of Type 2.

Permissioned blockchains, and consortium blockchains in particular, are seen as offering particular value in cases where multiple interacting companies, or other parties, must currently spend large amounts of time and money resolving discrepancies between their records, since none of the parties can be entrusted with maintaining a 'master' version of the relevant records for everyone else [110]. For example, Accenture has estimated that the use of consortium blockchains could save investment banks billions of dollars each year in costs related to the reconciliation of independently maintained records [65]. Another broad application is in tracking goods as they move through supply chains, changing hands across several different groups. The diamond industry, for example, has already begun to explore this use of consortium blockchains, gaining an insight into the movement of individual diamonds that they could not gain before [108].

For an overview, the United Kingdom Government Office for Science's report, "Distributed Ledger Technology: beyond blockchain," summarizes many other uses for permissioned blockchains that are currently being considered by governments and industries, such as tracking welfare payments and preventing document forgery [110]. Overall, the report expresses a high level of enthusiasm about permissioned blockchains, describing them as the first significant innovation in ledger technology since ancient times.

On the other hand, the developers of permissionless blockchains often have quite different sets of applications in mind. Although gains in efficiency and integration are also of interest, many developers are motivated by the search for ways to cut trusted institutions out of larger and larger segments of human activity—including, as section 3.3 will discuss, monetary policy, property ownership, and contract enforcement [26, 112].

However, the opportunities offered by blockchains also come with substantial downsides. In particular, most blockchains are currently able to process no more than a couple dozen new records per second, and require large expenditures of computing power and storage space, in contrast with the high processing speed and relatively low cost that can be achieved with centralized databases [37, 41]. This makes current blockchain technology inappropriate for a wide variety of applications, and, as discussed in section 5.2, raises concerns about whether existing blockchains can "scale" to accommodate much larger number of users. Addressing these issues remains a highly priority research project within the developer community, adn they may, more than anything else, be the core technical bottleneck to more widespread and sophisticated adoption.

The next section will explore the "distributed consensus protocols" that enable blockchains in greater detail.

2.7 Blockchains 2: Distributed consensus

Typically, a blockchain has the same sort of structure as the tamper-resistant logs discussed in section 2.5. It is divided into sequential blocks, with each block containing a hash generated from the records it contains, as well as the hash of the previous block's header [77].

The problem of coming to consensus about a blockchain's contents, for the nodes involved in maintaining it, is then normally reduced to the problem of coming to a consensus about the contents of each new block.

In the case of permissioned blockchains—where there is a small, presumably well-vetted set of nodes participating—there are some protocols that look not entirely unlike traditional voting [7]. The problem is in fact more significantly complicated than a traditional voting problem, though, since there is no single node in charge of tallying the votes, and one node might create confusion by reporting different votes to different to different members of the network. However, a strand of computer science research, associated with a property known as "byzantine fault tolerance" (BFT), shows that there exist techniques that can be expected to converge on an accurate block so long as no more than 1/3rd of nodes conspire to vote dishonestly. These techniques serve as baselines, and extra design effort can be put into ensuring that all participating nodes have sufficient economic incentives to be honest, that dishonest nodes are likely to be identified, that consensus can be reached sufficiently quickly, and so on.

The case of permissionless blockchains—where the set of participating nodes is not vetted—is inherently more difficult. Naively, one might attempt to employ the same sort of protocol just described. However, two problems stand out:

> **Problem 1:** A dishonest party, if sufficiently motivated, might set up many different "spoof" nodes in order to artificially increase their influence on block creation[13]
>
> **Problem 2:** Since the parties involved in running nodes are not preselected, there may be no basis for expecting them to vote honestly

In response, permissionless blockchains offer the following two solutions:

> **Solution 1:** 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.

[13] In the standard terminology, this would constitute a "Sybil attack"

Solutions 2: 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, for the curious reader, the following bullet points provide more detail, describing the working of a somewhat typical permissionless blockchain, along the lines of Bitcoin [77]. (Note that this description is not meant to describe how *all* permissionless blockchains work, as there has been increasing diversity in this area.)

- There is some peer-to-peer network that anyone is free to join, along with some software associated with the blockchain that anyone is free to run.
- Everyone within the network can send valid record candidates to everyone else. Not everyone needs to run the software, but some do.
- Each person running software keeps a copy of what they regard as the most up-to-date version of the blockchain. They also maintain a backlog of record candidates they have received since the blockchain was last updated.
- If someone in the network is "honest," this means they are running a copy of the correct software, which adds record candidates to the backlog if and only if they are valid. It also means that they do not send conflicting record candidates to the network.
- At regular intervals, a random person in the network gains the ability to turn their backlog into a new block, B, which they add to their current version of the blockchain. For many permissionless blockchains, such as Bitcoin, they gain this ability by providing a "proof of work"—the solution to a brute-force computational puzzle generated from the content of the previous block. The person, identified by a permissionless key pseudonym, also gains the ability to add a record to the new block indicating that they have just earned a certain amount of digital currency, C.
- They advertise this new version of the blockchain, V, to the others in the network. Each of these others either adopts V, if it is consistent with their previous version and the backlog they have been maintaining, or ignore it, if it is not consistent.[14]
- 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 "official." This is partly a sociological phenomenon.

[14] Technically speaking, the user might also choose to accept the new block if they have outside reasons for expecting it be superior to their own records.

- 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 proposed by honest people will become official.[15] Since anyone proposing a new block is allowed to add a record granting themselves some amount of digital currency, it also follows that people will have 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 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 "official" history of records.

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 [14]:

- The selection of block-creating nodes on the basis of computing power 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)*. PoS systems generally work such that the probability that a user is selected is proportional the the portion of existing digital currency that they "deposit."

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.

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.

[15] Intuitively, it seems as though the "sufficiently high portion of the time" ought to anything above 50%. In fact, for somewhat technical reasons, the bar may be as high as 75% [42].

In addition, at least PoW schemes have the added benefit of rendering the blockchain tamper-resistant (see section 2.5). The further back a record is in the longest version of the blockchain, the more impractically expensive it would be for anyone to go back and make an equally long version where this record is absent or replaced.

2.8 Cryptocurrency

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

Although definitions vary widely, we will define *cryptocurrencies* as digital currencies whose users' balances are determined by the contents of permissionless blockchains. Bitcoin is the most prominent example. Cryptocurrencies are both an application of permissionless blockchains and a core feature of the schemes used to keep them reliable (see above).

A simple cryptocurrency system, similar to Bitcoin, might work in the following way [77]. (Note that this is not meant to be a description of all current cryptocurrency systems.)

- 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 public keys, whose (more concise) hashes are known as "addresses" and act as pseudonyms for the users.
- 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, where X is address of whichever user will own the coin. Users can only mint coins if they have just created a new block.[17]
- Say that a user with address X would like to give a coin of value V to a user with address 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 (see section 2.3). The record will be considered valid, and

[16] In an extreme case, the remaining honest users might even agree on a software update that prevents the manipulators from spending the digital currency they've saved or that replaces the relevant mining puzzle with a new version that the manipulators' computers are less suitable for.

[17] An alternative cryptocurrency system might associated a certain public key with a "central bank" that can mint new coins, or more might allow new coins to be created on the basis of a complex voting process.

> therefore be added to the blockchain, 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]

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 near-instantly to users anywhere else in the world, without needing to go through any institutions such as 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") and people with libertarian leanings [107, 66]. Cryptocurrencies may also allow users to get around politically motivated restrictions on the use of traditional payment services—as in the case where Wikileaks managed to sustain itself through cryptocurrency donations after major credit card companies blocked payments to it [72].

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 systems, coins can only be minted ("mined") by users who create blocks, meaning that there is no central bank, and anyone is free to mint or receive coins using an indefinite number of public key pseudonyms. The initial user base for Bitcoin had a strongly libertarian (and often "crypto-anarchist") bent, as much of its appeal was the the opportunity to make transactions without having to rely on large institutions, without having to use currencies managed by central banks, and without having to attach one's real name to one's pseudonym [80]. However, over time, the currency has slowly crept further into mainstream use. At the time of writing, more than $50 billion worth of bitcoin exist.

Many other cryptocurrencies have also been created since 2009, although at the moment only one other cryptocurrency, associated with the Ethereum blockchain, has a userbase of comparable size. (I will discuss Ethereum in greater detail below, in section 2.11.)

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

[18] As this description may suggest, one of the primary ways that a dishonest party might try to manipulate a permissionless blockchain is to generate an alternative version of the chain that lacks a record of them giving a coin away—thereby allowing them to, in practice, "double spend" the same coin. Since tying transactions to digital signatures is enough to prevent the users of a digital currency from forging transactions in each other's names, it is mostly this risk of a "double spend attack" that implies the need for a trusted log of previous transactions.

Although there is much more that can be said about cryptocurrencies (and more will indeed be said in below), I will close this section with a brief discussion of the relationship between cryptocurrencies and privacy.

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 keys, and this makes all the difference [77]. 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 easier to attach pseudonyms to real-world identities by analyzing patterns of payments [84]. Furthermore, once a pseudonym is linked to a name, an extraordinary amount about the user may be revealed by their complete transaction history. 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 outcome of these transactions.

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

2.9 Zero-knowledge proofs and zk-SNARKs

Zero-knowledge proofs, of which *zk-SNARKs* are one variety, are proofs of mathematical statements that do not convey any information other than that the statements are true [52].

This might sound impossible. However, even without entering into technical details, it is possible to make the existence of zero-knowledge proofs less counterintuitive by considering an analogous example [45].

Say that 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 predict how many balls will be in the bucket if she pours either cup in. She allows Bob to pick, at random, one of the two cups to pour in, and she allows Bob to count the balls. Since her prediction 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 prover her proposition about the balls in the cups (probabilistically) while still keeping their quantity a secret.

This case closely echoes the procedures used in the class of zero-knowledge proofs known as *interactive zero-knowledge proofs*. These proofs require a sequence of interactions to take place between a prover and a "verifier," who is tasked with providing randomly selected inputs as part of process. Interactive zero-knowledge proofs were the first class of zero-knowledge proofs to be discovered, and are now relatively well-understood. They find applications in authentication processes, as they allow users to demonstrate that they possess the information required to access certain systems without also sharing enough information to allow others to replicate their demonstrations [43].

The more interesting class of proofs, however, may be *non-interactive zero-knowledge (NIZK) proofs*, and in particular *zk-SNARKs* ("zero-knowledge Succinct Non-interactive ARguments of Knowledge") [17, 12]. Non-interactive zero-knowledge proofs, as the name suggests, are zero-knowledge proofs that do not require interaction between provers and verifiers (although they do still require a source of randomness). zk-SNARKs are a sub-variety of NIZK proofs that are particularly short, efficient to check, and therefore practical to use.

Compared to interactive zero-knowledge proofs, zk-SNARKs are poorly understood. Belief in their reliability is based on assumptions that are stronger than those required for the interactive variety, for instance requiring trust that the source of randomness the prover relies on really is random. In practice, the parties that establish a platform for zk-SNARKs must also be trusted to destroy (rather than exploit) some sensitive information used in the platform's set-up [13].[19]

Still, the applications of zk-SNARKs could be much more significant. Interactive zero-knowledge proofs only serve as proofs for individual verifiers, who engage in a back-and-forth communication with the prover, and for anyone who trusts that these individual verifiers have not colluded with the prover.[20] In contrast, zk-SNARKs allow mathematical statements to be proved to large masses of people.

This is the property that makes them of such great interest for developers attempting to increase the privacy of blockchain transactions. Of particular significance, zk-SNARKs make it possible to construct an electronic payment system where everyone can confirm the validity of payments but no one beyond the participants has access to the payment details [89].

The first example zk-SNARKs being used in this way is ZCash, a cryptocurrency developed by American and Israeli academics and launched late last year.

[19] This vulnerability provides particular cause for concern. However, over the past year there has been some promising into research into a variant of zk-SNARKs, zk-STARKs, that might help to resolve it [11].

[20] To return to the cups example, above, a verifier who colludes with the prover could consistently pick the cup that will make the prover's predictions true (rather than picking a cup truly randomly).

However, the developers of one of the most widely used existing blockchains, Ethereum, have begun implementing zk-SNARKs into their own platform, and it could be only a matter of time until the average cryptocurrency transaction truly is private [86].

Plausibly, the significance of this development may be to push financial transactions even further outside of this influence and watch of traditional institutions.

2.10 Smart property

Cryptocurrency is not the only form of property whose ownership can be determined by the state of a blockchain.

As a simple 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, the "Bitland" project in Ghana aims to record local land deeds on a blockchain in order to decrease the risk of meddling by corrupt officials [4].

Additionally, as a slightly more novel case, a company can also create "tokens," whose ownership is recorded in just the same way cryptocurrency ownership is, and sell them to users with the promise that they can be exchanged for a particular good or service in the future.

The link between blockchain-based records and property ownership can also be made more concrete, however, through the creation of *smart property:* internet-connected devices that respond to the state of a permissionless or consortium blockchain [77].

Consider the following system:

- The device is initially associated with some address[21], 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 sign it with their private key (see section 2.2). They meant send this message through a channel such as a Bluetooth connection or a card reader slot.
- Say that the owner has address X, another person has address Y, and the device is denoted by D. If the owner would like to give the device to the person with address 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

[21] As a reminder, an "address" is generated by taking the hash of a user's public key.

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" (i.e. the trend of more and more devices having internet connections), it seems like the large-scale implementation of physical smart property could be feasible [32]. This implementation could be achieved using permissionless blockchains or consortium blockchains maintained by the devices' producers.[22]

However, it remains to be seen whether there will be any significant user interest in smart property. As with cryptocurrency, some might find it appealing that smart property can reduce the demands on institutions that record the ownership of property, protect it, and facilitate its use and transfer (although it is unclear by how much). Smart property might also offer value by reducing inefficiencies in economic transactions and consolidating records of ownership.

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 conerns than less comprehensive centrralized databases. This would seem to be an important limitation.

On the other hand, there could be some cases where (at least partial) transparency would be a desirable feature of smart property. For example, we can imagine the usage and possession of some sensitive dual-use technology being tracked as it moves through supply chains and enters operation.

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 which, by design, a government-held private key is always capable of shutting down or transferring ownership for, should an appropriate legal order be given.

[22] The advantage of blockchains in this case that they would prevent users from needing to trust any individual company with maintaining ownership records—and potentially with the power to change access rights to a given device or, as occurred in the case of the "smart home" company Revolv, to shut down the service that allows the device to function at all [58]. In addition, blockchains could provide a unified, trusted place for all relevant records to be kept, making it easier for pieces of smart property produced by different manufacturers to interact.

2.11 Smart contracts

A *smart contract* is a piece of computer code, recorded in a permissionless or consortium blockchain, that can be used to specify under what conditions certain transactions will occur between users.[23]

One core application of smart contracts is to execute agreements between blockchain users, reducing the level of trust that needs to be placed in third party institutions (such as courts) to enforce the agreements, while also potentially making the agreements more efficient to execute.

As a toy example of this application, consider the following case:

- A user with address X would like to sell a smart-property car, denoted by D, to a user with address Y, at the cost of V units of cryptocurrency.
- To accomplish this transaction, the seller can submit a record to the ledger of form "X agrees to give D to Y, if X agrees to give Y a coin of value V", along with a digital signature. If the buyer submits a corresponding record and signs it, then ownership of the car and ownership of the the coin will be recognized as having 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 on their end of the deal.

Naturally, the conditions for transferring an item through a smart contract can only be expressed in terms of the content of 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 whether it will rain tomorrow. A simple way of accomplishing this is to make the bet conditional on whether a trusted friend with a particular public key will submit a message tomorrow claiming that is raining. Trusted services could also be set to print information about the weather onto the ledger, to facilitate weather-related contracts.

There are also services in development that would use a mixture of reputation systems and consensus protocols to allow groups of strangers to provide reliable

[23] It is worth noting that, as with blockchains themselves, there is significant diversity is how different groups define the term "smart contracts." The initial definition of the term, given by Nick Szabo, did not specifically concern blockchains and stresses the analogy to traditional contracts: "a computerized transaction protocol that executes the terms of a contract" [98]. Other definitions conceptualize "smart contracts" as something closer to artificial agents that interact with blockchain users.

inputs to others' smart contracts. These services are known as "distributed oracle systems," and have so far been most extensively explored in the context of a betting market platform known as Augur, which is associated with the Ethereum blockchain [78].

Blockchains differ greatly from one another in the variety of the contracts that they allow users to create. Ethereum, a permissionless blockchain launched last year, has the important distinction of being the first blockchain that allows for the creation of any smart contract that is expressible in computer code [27].[24] It has also served as the basis for several early consortium blockchain designs being explored by technology and finance companies interested in implementing the same smart contract features [82].

Some significant technical challenges arise when the users who maintain a blockchain are forced to process computationally intensive contracts. Ethereum's designers have tried to solve these challenges by creating a variable cryptocurrency fee that users must pay to add contracts to the blockchain, or to interact with them, which rises as a function of the contracts' lengths and run-times. Still, as section 5.2 will discuss, there remains the problem, common to all current blockchains, of scaling up the relevant techniques as the number of users and smart contracts grow larger.

Finally, it is important to note that smart contracts need not resemble anything like traditional legal contracts, and in many cases may be conceptually closer to digital agents that "live" on a blockchain or to building blocks for web applications, which help to ensure that these web applications will behave in a certain way and not unexpectedly become unavailable [35].

A frivolous example is a set of smart contracts designed to enable games of chess: users take turns submitting moves, and money is transferred from one to the other if certain conditions are met. If placed on a blockchain, this "distributed application" for playing games of chess could be designed to remain available so long as the blockchain continues to be active, and users could be certain it will make only valid transfers by examining its code. More advanced applications include those associated with Augur, the betting market platform mentioned above; proposed platforms that allow users to rent out space on other users' computers for encrypted cloud storage; and even, as section 3.8 will discuss, what are known as "decentralized autonomous organizations" [27][25]

[24] More precisely, it is the first blockchain that allows users to create smart contracts in a language that is "Turing complete."

[25] At the moment, computational constraints prevent these entities from being very complex, to the extent that the chess example is actually very non-trivial to implement. However, some further methods for navigating these constraints have recently been proposed. One particularly interesting proposal is known as TrueBit, which may allow some of the work involved in executing and verifying a given smart contract to be outsourced to a subset of parties much smaller than the total numbers of nodes maintaining the blockchain [100].

A smart contract-centric view of blockchains has also led Vitalik Buterin, Ethereum's primary creator, to offer the following alternative (and aspirational) definition of blockchains [25]:

> *A blockchain is a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong cryptoeconomically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies.*

Watching how Ethereum and consortium variants of it develop may be a good way of judging how substantial the potential for smart contract use is.

2.12 Homomorphic encryption

Finally, *homomorphic encryption* refers to methods of encrypting data that allow computations to be run on it (such that the outputs of the computations are also encrypted) [105].

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. Schemes that are homomorphic under only addition or only multiplication have been known for several decades.[26]

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 [48]. 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 for one party to offer sophisticated data processing services to another without needing to have access to the data. As examples, we can imagine applications like the following: a cloud computing service that does not need to know what it is being asked to compute; medical companies that flag health concerns on the basis of individuals' DNA, without needing to know their genetic data; and dating companies that recommend user pages on the basis of reported preferences, without needing to know their preferences.

As section 3.6 will discuss, fully homomorphic encryption might also have important applications in surveillance, for instance allowing governments to extract information about criminal behavior from data collected by technology compa-

[26] In fact, one of the very first public-key encryptions schemes, RSA, is homomorphic under multiplication.

nies without needing to access to the data directly.

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 several orders of magnitude blow-ups in the size of the encrypted data, as it is operated on, and in the time the operations take to complete. For an example, as least 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 (addition is much less costly) [5].

Some help is provided by *somewhat homomorphic encryption* schemes, also recently developed, which offer significant (but not decisive) speedups at the cost of allowing for only a finite number of operations. Over time, progress in discovering more 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, the applications remain relatively narrow [75].

3 Speculative consequences

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

Since the list of such proposals is very large, I have limited myself to consequences that I consider to have *plausible large scale political significance*. For example, although many financial institutions are investigating the possibility that consortium blockchains will allow them to reconcile their records more efficiently, it is not clear that this development would have much relevance outside of the industry.

I will be discussing the following eight predictions:

1. Information channels used to conduct surveillance may "go dark"
2. It may become more difficult to forge convincing photographs and videos
3. Reliance on banks, courts, and other trusted institutions may diminish
4. Borders may become less significant
5. In a variety of domains, it may become possible to solve "coordination problems" that existing institutions cannot
6. Privacy-preserving online services and surveillance may become more feasible
7. Privacy-preserving agreement verification may become more feasible
8. New, decentralized political entities may emerge

There are of course reasons to be skeptical of all of these predictions. I discuss some reasons for skepticism within this section, but section 5, "Limitations and skeptical views," describes several in significantly greater detail.

3.1 Information channels used to conduct surveillance may "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 cryptocurrency systems that apply zero-knowledge proofs and other techniques meant to obscure economic transactions; and, third, the potential of homomorphic encryption and blockchain-based applications 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 [40]. This, further, prevents government agencies from gaining access via the service provider.[27] Just over the course of 2016, the number of end-to-end encryption users likely increased by more than a billion. This is due to a number of major providers, most notably WhatsApp, beginning to offer the service by default, and other apps specifically designed for secure messaging, such as Whisper, developing much larger audiences [9].[28] In response to this trend, a number of prominent of prominent officials, including former FBI director James Comey, have sounded an alarm that the information channels they rely on are "going dark" [36, 46].

The develop and adoption of privacy-preserving cryptocurrencies might 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 [84, 77]. Although they do allow users to make transactions under pseudonyms (hashes of their public keys), it is in practice fairly easy to connect pseudonyms to

[27] That is, it prevents the government from gaining access to messages sent while the service provider is genuinely offering end-to-end encryption. It may still be possible for agencies to require or pressure the service provider to weaken protections on users through a (potentially innocuous-seeming) software update.

[28] 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 into revealing their password. In addition, the end-to-end encryption might not in fact be perfectly implemented. For instance, it may be still be possible for agencies to require or pressure the service provider to weaken protections on users through a (potentially innocuous-seeming) software update.

real-world actors. This is chiefly because a record of every single transaction ever made using either of these cryptocurrencies is recorded on the relevant blockchain, and, through network analysis, it practical to identify parties, for example through the patterns of transactions they engage in. Furthermore, once a pseudonym is connected to the proper identity, all of the user's behavior using that particular pseudonym will be known at once. Finally, users who do not use anonymizing services such as Tor can often still be identified through their IP addresses, 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.

If cryptocurrencies do ultimately make surveillance much more difficult, then it may be the result of cryptocurrencies that apply zk-SNARKs. As discussed in section 2.9, at least one new cryptocurrency, launched in just the past year, claims to offer users the ability to make transactions whose contents and participants are obscured from other users. This cryptocurrency, ZCash, is not yet widely used, in part because it is relatively inefficient, but the developers of Ethereum have plans to incorporate its features into their own platform in the future.[29] Some security and practicality concerns remain, in part because zk-SNARKs are still poorly understood, but in the long run it seems plausible that cryptocurrency transactions will become significantly less transparent. In addition, even if zk-SNARKs do not become widely used, other a handful of other novel methods of obscuring transactions, including "mixing services" and "state/payment channels," could achieve similar effects.[30]

In addition, homomorphic encryption and blockchain-based applications might, in the long run, 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 to private companies' economic incentives to collect user data and due to the "internet of things" (IoT), the growing collection of internet-connected, sensor-equipped devices that potential surveillance targets interact with [46]. While these points are likely robust in the short run, they may not be in the long run. As will be discussed in section 3.6, if sufficient technical advances are made, then homomorphic encryption could singificantly reduce economic incentives to collect unencrypted user data, by making it possible to offer web services and even learn from data without gaining access to it in unencrypted form. Furthermore, the use of smart property, with much of its behavior mediated through blockchain-based applications, could limit the need for device manufacturers to store the associated

[29] In addition, ZCash, which allows users to make either "shielded" (i.e. anonymized) or "unshielded" transactions, is still primarily used for unshielded transactions.

[30] As a reminder, mixing services work by executing arbitrary 'coin swaps' between users, to make network analysis more difficult. State/payment channels, which offer efficiency as well as privacy gains, work by allowing sets of users to conduct sequences of transactions outside the blockchain, then only record the final result on the blockchain when they are ready to 'settle up.'

data themselves, and could limit their ability to turn the devices, at government request, into tools of surveillance [32].[31]

On the other hand, it is not impossible to imagine a future in which cryptographic developments make certain forms of surveillance easier.

First, the widespread use of cryptocurrency and smart contract systems that do not apply zk-SNARKs could make transactions more transparent than if they were conducted through other means (such as physical cash or institutions that keep private and independent records).

Second, certain applications of consortium blockchains could make it easier to track the flow of goods and detect illicit trade or forgery. (To repeat a minor example, the diamond industry has already begun to use consortium blockchains to track individual diamonds as they change hands between companies [108].) Furthermore, if the property being tracked is smart property, then records could also conceivably be kept of its use.

Third, as will be discussed in sections 3.6 and 3.7, certain uses of homomorphic encryption and zero-knowledge proofs 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 effective surveillance without running up against civilian privacy concerns.

Fourth, digital currency, smart property, or smart contract systems could be set up that grant parties holding certain keys special viewing or transfer rights—for example, requiring any transaction of a certain form to be signed off on by a government regulator. In such a case, it could become possible to obtain much more information about citizens' behaviors, as well as much more control.

Finally, the trend of associating individual citizens with government-granted cryptographic keys, as seen in countries like Estonia, could also enable surveillance, in the event that individuals are required to use their digital signatures (or other forms of national ID) when engaging in a growing range of activities [71].

[31] As a further note, the authors Berkman Center report also mentions two other lines of argument against the "going dark" narrative. First, they write: "Software ecosystems tend to be fragmented. In order for encryption to become widespread and comprehensive, far more coordination and standardization than currently exists would be required." Second, they write: "Metadata is not encrypted, and that vast majority is likely to remain so. This data would need to stay encrypted in order for systems to operate: location data from cell phones and other devices, telephone calling reocrds, header information in e-mail, and so on." If blockchain-based applications become widespread, with blockchains serving the roles of "application connectors," then the first point may become less true [113]. On the second point, there are currently a number of projects, such as the MIT-based project Vuvuzela, working to develop messaging systems that obscure metadata in addition to the bodies of messages [40, 106].

3.2 It may become more difficult to forge convincing photographs and videos

As mentioned in section 2.4, one application of trusted timestamping is to help users provide evidence that their photographs and videos have have not been doctored.

Timestamps put an upper bound on when the image or sequence of images was created, providing assurance that, at the very least, it was not doctored after the vent supposedly being depicted. This assurance can also be supplemented by including, in the images, data that the creator could not possibly have had access to before a certain point in the past. If the gap between the lower and upper bounds on the images' creation is sufficiently small, then this can be very compelling evidence that the images were genuinely recorded in this window, rather than being very rapidly rendered.

For a single photograph, a simple method would be to show some unpredictable and reliably timestamped real-time data source (such as block hashes for a permissionless blockchain) in the scene being photographed, then immediately post a hash of the photograph to somewhere that reliably and permanently logs the dates of posts (such as the same permissionless blockchain).[32] The shorter the window between the timestamp on the relevant block and the timestamp on the photograph's hash, the costlier it would have been for someone to fabricate the image. The cost will also go up substantially if what is being shared is a video, rather than a single still image.[33]

While there may not currently be a great need to use this scheme, it could become very useful in the relatively near future. Progress in machine learning has made it possible to produce doctored images, video, and audio that are increasingly realistic [33]. Similarly, although at great expense and using rather different techniques, Hollywood studios have recently become able to produce nearly photorealistic renderings of long-dead actors. In the long run, it seems plausible that it will no longer be possible to trust the authenticity of important images or videos, with significant negative implications for the news media, intelligence services, and other such groups.

The scheme described here could be useful for avoiding this negative outcome.[34]

[32] This scheme was first described to me by participants of a workshop at the Future of Humanity Institute. A similar idea is explored in a recent article on preventing video forgery in the specific case of dashcam recordings of car accidents [49].

[33] As an example of an even simpler version of this technique, Vitalik Buterin once posted a Twitter photograph of himself holding a handwritten version of the most recent Ethereum block hash to dispel rumors of his death.

[34] Other techniques for verifying authenticity include machine-learning algorithms designed to detect tell-tale signs of manipulation and hardware-based timestamping, which requires trust in the digital camera used to collect the relevant images [94]

3.3 Reliance on banks, courts, and other trusted institutions may diminish

Cryptocurrencies, smart property, and smart contracts may diminish the extent to which many groups rely on trusted institutions when transferring or maintaining ownership of property.

As noted earlier, one key feature of cryptocurrencies such as Bitcoin is that they allow users to make virtual payments without relying on banks or credit card companies as intermediaries, and without relying on central banks to manage the currency system generally. If a significant number of people find it appealing from efficiency, liberty, or privacy standpoint to avoid relying on these institutions, when possible, then the role that they play could shrink.

One early example of a decline in influence is banks' and credit companies' inability to prevent payments to Wikileaks, back in 2012, when all of the major companies decided to refuse payments to the activist group. As already mentioned, 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 [72]. In the same vein, a sign of the potential decline of central banks' influence is given by Venezuela, where a very large number of citizens have begun to use bitcoin, rather than the hyperinflated national currency [28]. Here, cryptocurrency use has already had a significant undermining effect on the country's strict currency control policies.

Similarly, the use of smart contract applications can reduce reliance on tech companies that provide web services, since such applications can be guaranteed to continue running as promised once placed on the blockchain [25]. For example, there are a number of proposed or early-stage blockchain services to allow users to rent out storage space for encrypted files, using smart contracts, as an alternative to services like Google Drive [109]. If scalability issues can be resolved (see section 5.2), then some writers have suggested that decentralized, potentially profit-less alternatives to the services provided by companies like Uber might also become possible [87].

Finally smart contracts, especially when used in conjunction with smart property, might allow people to reduce their reliance on the legal system when entering into agreements. As the legal scholar Lawrence Lessig has noted, it is sometimes possible to achieve a particular end through either legal code or computer code [34]. In this sense, smart contracts would appear to significantly increase the space of possibilities for what can be achieved through computer code. If people can much more easily and frequently enter into agreements with one another that they trust will be enforced, without having to trust that police officers or a court system will enforce them, then this would seem, at least to

some extent, to diminish the importance of the traditional legal system.[35]

These views are further explored in the paper "Decentralized Blockchain Technology and the Rise of Lex Cryptographia," by Aaron Wright and Primavera De Filippi [112]. They argue that smart contracts will functionally constitute a new subset of law, termed "Lex Cryptographia," and that "centralized authorities, such as governmental agencies and large multinational corporations, could lose the ability to control and shape the activities of disparate people through existing means."

At the same time, as section 5.4 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 legal contracts.

The potential of decentralized services might also be undermined by regulation (see section 5.3), issues with the consensus protocols used in permissionless blockchains (see section 5.5), or the possibility that many existing trusted institutions are "good enough" that there is no strong incentive to seek alternatives (see section 5.6).

3.4 Borders may become less significant

This point is primarily an implication of the above one.

To the extent that there is a decline in reliance on services provided by local institutions—such as fiat currencies and contract law—it seems that there should also be a decline in the importance of physical geography.

If an individual is primarily using cryptocurrencies and services provided by distributed applications, and if the individual finds smart contracts sufficient for most agreements they enter, then it ought to matter at least somewhat less where they are residing.

Another source of the notion that borders may become less significant comes from the notion of "e-residency" [90]. In 2014, Estonia launched a program to allow individuals around the world to become "e-residents" of Estonia, by applying to receive a cryptographic IDs that allow one to sign legally binding documents, engage in Estonian e-banking, and establish Estonian companies and manage them from anywhere in the world. While this project is an idiosyncratic one, some commentators see it as suggesting a potential trend in the evolution of nation-states. For example, one recent report, associated with the University of Oxford Cyber Studies Programme, argues, "Perhaps individual

[35] As another framing, contract enforcement mechanisms may be either public (for instance, a formal legal system) or private (for example, Mafia enforcers or reputation harm) [54]. The use of smart contracts may increase the relative prominence of private mechanisms.

identity should be based less on one's place of physical birth or residence and more on intangible values and senses of belonging. In time, the e-Residency project may radically alter the perception of belonging so that it is no longer anchored to the territorial nation-state" [70].

The notion of e-residency might also be connected to the more radical notion of "decentralized autonomous organizations," discussed in section 3.8. These would be borderless organizations, potentially political organizations, that allow their members to coordinate through complex systems of smart contracts. However, it is unclear that such entities are plausible.

3.5 In a variety of domains, it may become possible to solve "coordination problems" that existing institutions cannot

It sometimes happens that parties would, in theory, be interested in entering into agreements with each other, but find themselves unable for one of a number of reasons. For example, there may be no way to ensure that the other parties will uphold their ends of the agreement, or there may be no way to aggregate the various parties' preferences efficiently enough.

These cases constitute "coordination problems," and one way to conceptualize institutions is as political actors capable of resolving coordination problems between others [91]. For example, social contract theorists view states as enforcing tacit agreements between individuals to respect one another's rights, and the realist school of international relations tends to view mutually undesirable phenomena like war, arms races, and environmental "tragedies of the commons" as arising from the absence of a global sovereign capable of playing a similar role [53, 73].

To the extent that smart contracts can enable coordination in the absence of trusted institutions, we might hope that these new technologies will help to resolve some of the remaining coordination problems that existing institutions cannot.

While it is fairly common for smart contract advocates to describe the potential of the technology in terms of coordination problems, there have not yet been many concrete proposals along these lines [21].

To gain a better sense of the space of possible proposals, we can ask which sorts of coordination problems might not already be solvable by relying on traditional contracts and existing institutions responsible for carrying out or enforcing them. For cases where the relevant actors are located within a single region, or under the same domestic sovereignty, we divide up the relevant cases into subsets where:

1. the desired agreement would be *too insignificant* for it to be worth involving third parties.
2. the available institutions *cannot be trusted* to carry out or enforce the agreement, due to accessibility issues, corruption, weak rule of law, or other dysfunction.
3. the available institutions are *incapable* of carrying out or enforcing the agreement, because it is too cumbersome to manage with traditional mechanisms (ex. it involves too many parties, requires too much information aggregation, would likely result in a lengthy dispute process, etc.).
4. the available institutions are *not willing* to carry out or enforce the agreement, because it would enable activities that are considered illegal or immoral (ex. enforcement of agreements within organized crime or dissident groups).

We can also add, as its own somewhat

5. the relevant parties are spread across the globe, and no international organizations or states are sufficiently trustworthy, able, and willing to carry out or enforce the agreement.

It may be fair to say that cases of type (1) and (2) most plausibly present opportunities for smart contract use, with (2) being the more important of the two. Cases of type (3) suggest potential applications in collective action—where large numbers of people pledge to engage in a certain behavior if others do—but the practicality of this is unclear. Cases of type (4) are concerning, with some authors arguing that "criminal smart contracts" could be used to facillate the leakage of confidential information, assassinations, and terrorism [67]. However, it is also unclear to what extent criminal organizations are handicapped by trust issues that smart contracts could plausibly help resolve.

Use cases of type (5) seem, at least at first glance, both the most significant and the most implausible. We can, for example, imagine countries signing onto a smart contract to cut carbon emissions, with automatic financial penalties if other countries, a distributed oracle system, or a set of internet-connected sensors judge that they have violated it; the relevant blockchain could be a well-established permissionless blockchain or a consortium blockchain held between several countries. More fancifully, we can imagine a pair of countries signing onto a smart contract that will deactivate their "smart property" weapons systems at the same time. However, to be used in such cases, smart contracts would need to become *extremely* reliable, to an extent that, as section 5.4 will discuss, it may not be reasonable to expect.

3.6 Privacy-preserving online services and surveillance may become more feasible

As discussed above (see section 2.12), homomorphic encryption makes it possible for one party to process another party's data for them without needing direct access to it.

If fully homomorphic or somewhat encryption becomes practical, or if partially homomorphic encryption finds more significant applications, then the effect could be a very significant increase in user privacy across many different online services [75]. For example, it is not in principle necessary for a service that targets products to you, in response to your browsing habits, to have access to these habits—just as, with end-to-end encryption, for a service that sends messages for you to have access to these messages.

We can imagine further applications, as well.

First, companies and other groups might also use homomorphic encryption to *learn* from their users' collective data without accessing it. Currently, many valuable pieces of software are created by training machine learning algorithms on large collections of user data. However, as described in a recent essay by Andrew Trask, the use of homomorphic should make it possible to apply learning algorithms—particularly, neural networks—to multiple users' data without ever needing to access this data or reveal the features of the learning algorithm to these users [101].[36] This development would further strengthen individual privacy, and further diminish the incentives for companies to collect user data.

Second, and potentially much more significant, is the possibility that homomorphic encryption could eventually enable less invasive and more effective forms of government surveillance. Some early work has shown that the use of even just additively homomorphic encryption may enable new forms of privacy-preserving surveillance systems, such as a facial recognition system that reports only whether a given person has their face recorded in a dataset (and not who, particularly, the person is) [39]. With fully homomorphic encryption becomes sufficiently practical, the applications could be much greater [1]. On this point, another essay by Trask describes the following non-invasive surveillance scheme [102]:

- A government agency designs a classifier system intended to predict a certain criminal behavior from collected data. (As a concrete toy case, we can imagine it predicts illegal spamming behavior on the basis of sent emails). After being trained using a relevant dataset, which includes in-

[36] Trask is currently leading OpenMined, a project related to this idea. In the long run, his team aims to create a blockchain application that allows users to grant others the opportunity to train machine learning models on their encrypted data.

stances of criminal and non-criminal behavior, the classifier will have the function of flagging individuals for further search whenever it judges the probability of criminal behavior to exceed a certain threshold. To ensure that the number of false positives does not exceed the intended tolerance, the trained system can also be checked by applying it to a test set.

- This procedure is overseen by an independent auditing body. The body confirms that the classifier has the properties claimed.
- Now the agency homomorphically encrypts the *parameters* of the classifier. A new version of the classifier is created, which maps these encrypted parameters and unencrypted user data to encrypted classifications. The agency provides a version of this classifier to companies that collect user data, such as messaging service providers.
- These companies apply the classifiers to their users' data, producing encrypted classifications. The companies send these classifications back to the agency for decryption. The agency gains a list of suspicious individuals, who can be subjected to further search, without ever directly having access to user data. The company does not gain access to this list, unless the agency shares it with them, and does not gain access to the confidential parameters of the classifier (which, if leaked, could make tricking the classifier easy).
- Different classifiers could be deployed for different crimes. The explicitly coded threshold for reasonable suspicion could also vary with the severity of the crime, and could be decided through explicit public discussion.
- Finally, the legal justification for using such a classifier would be the same as the legal justification for using drug-sniffing dogs. In particular, the use of dogs is also grounded in the fact that they are trained to only output binary information—"high likelihood of drugs" or "low likelihood of drugs"—and have sufficiently low false positive rates.

If this scheme is ever becomes feasible, then it could allow for much more privacy-preserving methods of providing security. This may be especially true as machine learning progress enables increasingly accurate classifiers.

However, some important caveats are in order.

First, as discussed in sections 2.12 and 5.1, fully homomorphic encryption is currently extremely resource-intensive. This means that, at the present date, applying complex classifiers of this sort to large numbers of users' data is not practical. The main question, then, is whether it will never become practical, or whether it will simply take many years.[37]

[37] Nonetheless, some simple classifiers might still be useful. For instance, in his essay, Trask demonstrates for the toy case of spam detection it is enough to use encryption that is homomorphic only under addition. The case of privacy-protecting facial recognition with additively homomorphic encryption, above, is another valuable proof-of-concept.

Second, for many crimes, it may not be practical to develop classifiers that are sufficiently and demonstrably accurate. This is particularly true for crimes with very small numbers of true positives, such as terrorism. The viability of the scheme also requires, to some extent, continued progress in artificial intelligence to raise the achievable accuracy level.

Third, it may be difficult to obtain suitable datasets for training and testing without violating users' privacy in the first place.

Fourth, in this version of the scheme, there is still a need for the public to trust the auditing body to be truly independent and capable of detecting dishonesty, on the part of the agency, about the features of the algorithm that has been given to companies. Trask has suggested that smart contracts and other cryptographic technologies could be used to provide additional assurances about how the classifiers are created or applied, but these solutions would also be non-trivial.

Finally, 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. There may also be other cryptographic solutions here, which reduce the need to trust service providers, but these solutions would again be non-trivial.

In short, the possibility of non-invasive surveillance through homomorphic encryption, especially fully homomorphic encryption, would require significant technical advances. There may be reason for optimism, though, at least in the long-run. The intersection between homomorphic encryption and surveillance is a mostly unexplored research area, and there appears to be significant room for the exploration of new ideas in the coming years.

3.7 Privacy-preserving agreement verfication may become more feasible

Just as homomorphic encryption might enable less invasive government surveillance, in the long run, it could also might also enable less invasive monitoring associated with verifying agreements.

A toy verification system—although currently very far from feasible, due to the present limitations of both homomorphic encryption and machine learning—could be a set of cameras in a nuclear facility, hooked up to an encrypted machine learning classifier capable of identifying improper use of the facilities or tampering with the cameras. The encrypted output could then be sent to an inspection body, which then decrypts it to determine whether any violations have occurred.

Besides homomorphic encryption, zero-knowledge proofs might also have applications to verification. This connection is relatively straightforward in cases where the relevant behavior involves a digital object that lends itself to mathematical proofs—for example, if there is an agreement concerning property maintained on a blockchain. However, zero-knowledge proofs might also applications to more concretely physical cases.

In particular, in the past few years some work has begun to address the notion of "physical zero-knowledge proofs," or demonstrations that objects possess certain physical properties that do not reveal other relevant physical properties [45]. The cup and ball example described in section 2.9, in which it is possible to demonstrate that two cups contain the same number of balls without revealing what this number is, is in fact an example of a physical zero-knowledge proof.

Recent papers have proposed a method of demonstrating that a country is disposing of a genuine nuclear warhead without revealing its design details, as well as a method demonstrating that a subject's DNA does not match that found at a crime scene without revealing the subject's DNA [51].

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 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.

3.8 New, decentralized political entities may emerge

Some writers have speculated that smart contract technology could allow a new variety of political actor to emerge. In particular, 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 a large organization, possessing property and lacking a single leader, in which the protocols that individual members must follow are enforced by smart contracts [22]. [38]

Like a corporation, a DAO would function as essentially a unified entity, with its own property and (in some sense) its own goals. However, unlike a corporation, a DAO would have a foundational set of protocols that are immutable and automatically enforced. Furthermore, depending on the nature of the blockchain

[38] Some alternative definitions are loose to the extent that permissionless blockchains themselves constitute DAOs, with the nodes maintaining them acting as their 'members.'

the DAO is maintained on and on the nature of the relevant smart contracts, individuals could be able to participate in a DAO under public key pseudonyms. These two facets would arguably make a DAO significantly different from any existing variety political actor.

So far the only prominent attempt to construct a DAO has been a venture capital firm, which attracted $150 million worth of cryptocurrency investments before being abruptly hacked due to an initially unnoticed bug [81]. Due to this auspicious beginning, and the lack of ambitious follow-up attempts, we do not yet know what a practical large-scale DAO might look like, or even if large-scale DAOs are practical at all.

Still, it is interesting to consider what the implications may be if successful DAOs come to be created. What could a DAO tech company look like? How about a DAO political party, a DAO criminal organization, or even a DAO country?

Some writers have made quite radical claims about the feasibility of replacing traditional systems of governance with DAOs. However, these claims are often somewhat 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" [74].

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 [57]. 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 indicate 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 to 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 the use of a consortium blockchains to maintain the betting records seems like a potentially useful way of increasing their integrity, but also somewhat tangential to the overall vision.

In a similar vein, however, other writers have argued that smart contracts could be a useful tool for experimenting with novel forms of democracy [23, 96]. 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, at least on particular sets of decisions; this representative might in turn pass their accumulated voting on to a representative they deem

to be even more well-equipped to vote well, and so on [16]. Both of these forms of democracy must almost certainly be implemented electronically, and require unified databases that are extraordinarly secure—since the databases must be trusted to track, for instance, active updates concerning who as the power to vote for whom, as well as to report ultimate election outcomes accurately. Blockchain technology, therefore, is seen as significantly lowering the trust threshhold for implementing these political systems, and making it relatively easy for small (and potentially geographically dispursed) political organizations to trial them. These organizations could constitute DAOs.

A number of writers have also 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 3.3 and 3.4. Marcella Atzori's paper "Blockchain Technology and Decentralized Governance: Is The State Still Necessary?" surveys some such claims, although she finds them unpersuasive [6].

Finally, we should note that although most current speculation about the political implications of DAOs seems to be associated with positive visions, it 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 particular incentives to transform into DAOs if this was feasible. As a toy example, we can imagine a DAO that is set up to create smart contract rewards for terror attacks on the basis of their pseudonymous members' votes, or even to continue offering rewards after its initial members stop interacting with it.

It must be reiterated, however, that the basic practicality and usefulness of DAOs is still entirely unproven. In fact, as section 5 will explore, some (potentially quite severe) limitations stand in the way of many of the most radical potential uses of smart contract technology ever being feasible.

4 Relevance of progress in artificial intelligence

The significance that technological developments in one field have often depends on the developments that 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 some of these interaction effects, this section will briefly describe some of the ways in which progress in artificial intelligence and progress in cryptography could be relevant to one another. Although artificial intelligence

is certainly not the only field I could consider here, it may be particularly worth considering, as the field has recently experienced a large surge in research activity and the effects of AI-based automation could conceivably touch every domain of human activity.

I list six potential interaction points.

4.1 AI systems may both enable and require more effective surveillance

Progress in AI could enable more effective surveillance by decreasing the cost of extracting information from collected data [30, 2]. It is plausible, for example, that machine learning algorithms applied to sets of personal messages could become fairly effective at automatically identifying criminals, dissidents, or other groups that state actors would have an interest in discovering without needing to sift through the relevant data by hand. However, as discussed in section 3.1, this trend might be undermined by developments in cryptography—specifically, by a move toward greater and more careful use of end-to-end encryption, toward the use of cryptocurrencies capable of obscuring transaction details, and toward uses of smart contracts and homormorphic that reduce the need for companies to collect personal data.

On the other hand, as discussed in section 3.7, if some practical limitations are overcom, then homomorphic encryption could also help to provide the best of both worlds for surveillance and security, by allowing government agencies to run narrowly targeted machine-learning algorithms on individuals' data without requiring direct access to it.

Finally, we can note that the relevance of AI progress in this area may not be limited only to the forms of surveillance it enables; AI progress may also increase the desire for effective surveillance by creating new security concerns, for instance the creation of highly disruptive malware or the automated use of weaponized drones [19].

4.2 AI systems may increase the need for anti-forgery schemes

As discussed in section 3.2, progress in AI continues to make fake photographs and videos more convincing and cheap to produce [33]. 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. Schemes of the sort described in section 3.2, which uses trusted timestamping to provide evidence for the veracity of images, could be an important

tool for avoiding these consequences.

4.3 Safe AI design and safe smart contract design may have formal similarities

One broad problem in the emerging field of "AI Safety" is the problem of designing AI systems that will behave in reliably beneficial ways [3, 2]. 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 wide 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 important tasks, techniques for avoiding accidents could become even more important.

Similarly, as will be discussed in section 5.4, one important factor restricting the applications of smart contracts is the need to ensure that they will behave as intended. For example, last year, a flaw in the design of a smart-contract based investment fund (a "DAO") infamously led it to collapse and lose tens of millions of dollars in the process [81].

With these 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 others.

As a point of disanalogy, however, it is important to note that current "scalability" constraints (see section 5.2) 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 judge who has won a game of chess. This means that smart contracts are quite distinct from the powerful AI systems that AI Safety researchers primarily have in mind. Specifically, this also means that, barring the development of significantly new methods for executing smart contracts, or extraordinarily large increases in available computing power, no advanced AI system could actually be run as a

smart contract.[39][40]

4.4 New coordination and verification mechanisms may be useful for governing AI systems

Generally, if progress in AI threatens to have highly disruptive effects—for instance, through enabling the creation of highly advanced and autonomous weapons systems, enabling more effective methods of surveillance and forgery, causing large-scale unemployment, or increasing the severity and frequency of AI accidents—then there could be a need for laws and international agreements controlling its application and development [59, 18]. If smart contracts ever become sufficiently reliable, then it is possible—although, for reasons discussed in section 5.4, not necessarily plausible—that they could have applications in enforcing such agreements. Similarly, it is possible that zero-knowledge proofs or homomorphic encryption could make it easier to verify compliance without requiring the relevant parties to share too much sensitive information (see sections 3.6 and 3.7).

4.5 Changes to the political landscape, generally, may impact the governance of AI systems

As an extreme case, if people like Ralph Merkle are correct in predicting that blockchain-based technology will lead to new forms of government, such as "DAO democracies"—or even if there is just a relative decline in the influence of centralized institutions like technology companies and banks—then this would be relevant to discussions of AI governance (see section 3.8). More concretely, however, we ought to expect the nature of any governance to depend on which

[39] One potentially promising avenue for creating more computationally intensive smart contracts is outsourcing smart contract execution to small sets of users, who run the necessary computations on their own computers and report the outputs to the blockchain, rather than requiring each device maintaining the blockchain to run all of the computations itself. The apparent flaw here is that this would require trusting a large enough portion of users to accurately report their outputs. However, there have been some proposals for systems of incentives, reputation scores, and verification methods that are intended to make inaccurate reporting much less unlikely. One example is TrueBit [100]. The reliability and practicality of these systems remains to be seen.

[40] AI systems and blockchains could still be connected in other ways, though. Smart contracts could be used for the acquisition of cloud computing power or data used to train AI systems, and an internet-connected AI system could be designed to provide inputs to the blockchain (for instance, for contract verification purposes) or to respond to its state. An AI system could also conceivably be designed to make blockchain transactions as an economic agent. For instance, bitcoin contributor Mike Hearn has a "thought experiment" about self-driving car that will drive people when they make cryptocurrency payments to its personal address, will make payments at charging stations to sustain itself, and will distribute its profits in accordance with smart contracts [63].

applications of AI technology the relevant parties either desire or fear. It would likely be relevant, for example, whether a given country could expect AI systems to dramatically increase their surveillance capabilities, or whether the proliferation of certain cryptographic technologies would create severe limitations (see section 3.1).

4.6 Fully homomorphic encryption may have applications in AI Safety

As described in a recent essay by Andrew Trask, fully homomorphic encryption could make it possible to train AI systems using encrypted data or encrypted virtual environments [101]. 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 be less likely to cause accidents, for example by being deployed or shared before their safety is sufficiently well-established. Eventually, once safety has been established, the owners of the systems could use their keys to decrypt its parameters and produce non-encrypted versions. 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 5.1).

5 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 fully homomorphic encryption; the difficulty of "scaling" blockchains; the threat of regulation; the impossibility of truly "trustless" smart contracts; the potential insecurity of permissionless blockchains; and the possibility that existing trusted institutions make many potential applications of blockchain technology redundant.

Other potential roadblocks, which I do not discuss here, include the arrival of quantum computers (which will render some cryptographic schemes insecure) and the enormous volume of electricity consumed by the proof-of-work protocols (which could be made unnecessary by a successful shift to proof-of-stake protocols).

5.1 The inefficiency of fully homomorphic encryption

First, as discussed above, all presently known schemes for fully (and somewhat) homomorphic encryption require extremely large quantities of computing power, compared to what would be required to perform the equivalent computations on unencrypted data.

This makes them impractical for all but very simple computations, and, for example, rules out the possibility of using them to achieve broad privacy-preserving surveillance (see section 3.7) 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 order-of-magnitude increase in available computing power. Fortunately, there has already been tremendous progress in finding schemes that are more efficient than Gentry's original 2009 scheme, and there do yet appear to be any solid theoretical arguments for why much more progress could not be made [75]. Similarly, for many decades the growth of computing power has been exponential, so in future decades there could plausibly be vastly greater resources available.

In addition, partially homomorphic encryption, which is much less computationally intensive, may also have significant privacy-preserving applications of its own [39].

5.2 The difficulty of "scaling" blockchains

As mentioned in section 2.6, most existing permissionless blockchains do not support more than a dozen or so transactions per second, compared to the thousands per second that, for example, a company like Visa can process.

Such limitations put a bound on how many users these blockchains can plausibly sustain, as well as how complex or active the services built on top of them can be [37]. Almost certainly, the most radical visions of blockchain's potential—such as the vision that smart contracts will enable the creation of vast new political entities (see section 3.8)—are impossible with these limitations in place. In addition, these limitations put a bound on the security that permissionless blockchains can achieve, since the more wealth is intertwined with them the greater the total force of the incentives to maintain them properly will be.

The source of the problem, here, is that existing blockchains require all of the nodes maintaining the blockchain to verify *each* transaction that occurs, rather than splitting the labor between them.

While there are a number of proposals for techniques to somewhat increase

the capacity of blockchains, the ones with the largest potential effect concern "sharding" [41]. In these techniques, a blockchain is split into multiple shards, with each shard containing a subset of users and smart contracts. Then, some nodes can opt to be involved only in verifying transactions that directly involve the users or smart contracts in their shard. The more shards there are, the lower the burden is on each node.

Naive sharding proposals face obvious problems, like facilitating interaction between the shards, preventing successful attacks on individual shards, and generally ensuring security and consistency despite the division. However, more sophisticated proposals show promise, and Ethereum's developers believe they are ultimately on track to implement it for the Ethereum blockchain.

Time will tell whether sharding—or perhaps other techniques—will ultimately enable the scaling of blockchains.

5.3 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 decode should be made illegal [8]. For the time being, all forms of encryption are perfectly legal to use within the United States and European Union, with some other major countries, like China, placing only relatively limited restrictions on use [88]. However, it 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 [29, 62].

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 [31]. 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 part to the fact that these blockchains are inherently difficult to interfere with and lack any discernible party that is in "control" of them [103, 69]. The most significant example of regulation so far has been a temporary ban on the sale of new cryptocurrencies instituted by the Chinese government, this past month, in reaction to concern that "initial coin offerings" were been used to carry out scams and circumvent regulations on funding companies [60]. US regulators have also been closely examining regulatory possibilities, also with a significant focus on "initial coin offerings" [61].

It is unclear to what extent the regulation of emerging cryptographic technologies is likely to limit their use. However, if the technologies ever become truly threatening, for instance by making it easier for terror groups to operate or creating financial instability, then dramatic actions by major governments are not inconceivable. Actions in this category might include the establishment of severe legal penalties for the ownership or sharing of restricted cryptographic technology, coordination with the manufacturers of the specialized hardware used to maintain proof-of-work blockchains, and arrests of high-profile developers and owners of blockchain-based property.

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), lessen the influence of national economic and political institutions like central banks (section 3.4), make it easier for threatening actors to coordinate (see section 3.5), or enable the creation of new decentralized political actors (section 3.8). In short, the uses of cryptography that are most desired by the libertarian and politically radical portions of the cryptography community may also be the ones that are most difficult to achieve.

5.4 The impossibility of "trustless" smart contracts

One core application of smart contracts is the potential to reduce the need to rely on trust when carrying out agreements.

However, although the word "trustless" is very often applied to smart contracts, their use does require some forms of trust as well. Whenever it is more difficult to establish these forms of trust than to trust traditional institutions, the appeal of smart contracts—at least as an alternative to these institutions—will be limited.

First, especially for smart contracts that run on permissionless blockchains (such as Bitcoin and Ethereum), there is a need to trust that these blockchains will not be disrupted. One might be concerned about such disruptions due to either the threat of regulation (see section 5.3) or the possibility that the consensus protocols used to maintain the blockchains are unreliable (see section 5.5).

Second, there is a need to trust that the code for any complex smart contract sufficiently captures what the parties signed on to it intend—especially since it cannot be modified after the fact. The importance of this form of trust is illustrated by the failure of a multi-million dollar smart-contract based venture capital firm (also mentioned in section 3.8). An unfortunate programming mistake, not noticed until it was too late, left open a loophole that allowed one user to siphon a third of the firm's money [44]. Plausibly, the more complex a smart contract is, and the more that hangs on it, the harder this form of trust

will be to come by. 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 [15]. In this vein, the fact that traditional contracts can be filtered through human interpretation, which is capable of navigating ambiguities and grasping obvious intentions, can be a very valuable feature. The small but somewhat long-standing field of "computational law" 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 appropriately with computer code [47, 92].

Third, 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, they will still need to trust third parties to accurately report that information. This trust might be placed in individual third parties, or in a further, potentially jury-like distributed consensus protocol involving multiple parties.[41] As mentioned in section 2.11, some Ethereum developers are currently attempting to design such protocols, with the goal of building a "distributed oracle system" [78]. However, it remains to be seen how successful such systems will be.

Fourth, there is a need to trust in the real-world significance of cryptocurrency and smart property. For example, a contract may only be worth signing if you trust that businesses will continue to recognize the value of the digital currency into the future, or if you trust that a given piece of physical smart property (such as the hypothetical blockchain-connected car described earlier) will respond to transactions the way its designers claim it does. Plausibly, regulation would be necessary for establishing this form of trust, although regulation could also make it much more difficult (see section 5.3).

Fifth, there is a need to trust that force or coercion cannot be applied to counteract a smart contract. For example, the threat of thieves physically stealing an individual's smart property (and potentially rewiring it so it can be operated without the proper private key), or holding a gun to the head of someone who refuses to digitally sign an unfavorable smart contract, is of course a compelling argument for the value of external, physically-instantiated law enforcement institutions. More simply, there is a need to ensure that users' private keys cannot be stolen.

[41] As a very simple example example of how a consensus protocol like this could work, imagine two users who want to bet about what the highest temperature will be tomorrow in San Francisco. 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.

All these considerations suggest that, while smart contracts have clear utility, they do have very significant limitations, there is not yet a sound basis for predictions that they will encroach heavily on the territory ordinarily covered by legal systems or other traditional institutions (see section 3.3).

5.5 The potential insecurity of permissionless blockchains

As discussed in section 2.7, 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 blockchains' 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 sorts of 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 [77].

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.

As of 2016, three Chinese mining collectives collectively controlled more than 50% percent of the computing power directed at mining bitcoin [83]. If they decided to collude—for example, to "double spend" currency by replacing records of their own previous transactions—then they could produce a dishonest version of the Bitcoin blockchain 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 [42].

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 their accumulated bitcoin and in their investments into specialized mining hardware depends 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 a national government really wanted to disrupt Bitcoin, it seems that there would be nothing stopping it from investing 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 [77]. 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 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. [42] 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. [43]

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.

The case of proof-of-stake is similar, but perhaps somewhat more promising. In this case, 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

[42] This terminology comes from a short talk given by Vitalik Buterin at the December 2016 Future of Humanity Insitute workshop.

[43] As a further point, a highly empowered attacker (such as a national government) might also be able to decrease honest miners' power by shutting down mining facilities, restricting the sale of specialized hardware, or applying other controls.

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 5.2, 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 not yet been sufficiently well-tested or theoretically well-explored to make this conclusion an uncontroversial one among developers [79].

In summary, it is not yet clear just how much security can be offered by permissionless blockchains. This issue should be a key factor in determining the extent to which users feel comfortable tying their economic activities to such blockchains.

5.6 Compared to blockchain technology, trusted institutions may be "good enough"

Section 3.3 discussed the possibility that blockchain-based systems begin to take on many of the roles currently played by large centralized institutions.

For such visions to come true, it seems that these systems must begin to do a superior job of providing services that already available. However, because many existing institutions are highly reliable, convenient, and efficient, this is a difficult standard to meet.

As Vitalik Buterin writes in his essay, "The Problem of Trust" [26]:

> *Ironically enough, unlike in "crypto land", where [cryptocurrency] exchanges seem to routinely disappear with millions of dollars in customer funds, sometimes after apparently secretly being insolvent for years, businesses in the real world don't seem to have any of these problems. Sure, credit card fraud exists, and is a major source of worry at least among Americans, but the total global loss is a mere $190 billion – less than 0.4% of global GDP.... 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.*

To be sure, there are individual cases where blockchain-based services can offer clear advantages over centralized institutions. If you are particularly eager to

ensure that some web application will be available into the indefinite future, that your payments to a politically controversial group will not be blocked or traced, and so on, then there may not be any institution that you could trust more than a blockchain-based solution. It is unclear, though, how far such use cases extend.

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, blockchain-based services could eventually establish themselves as much more reliable than they are 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 as pre-empting the need for new institutions to develop sufficient amounts 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 blockchain systems do eventually achieve a prominence comparable to existing centralized institutions, it may not be by directly displacing them. Instead, blockchain 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 systems in political 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 ten years old, and has attracted significant attention for perhaps fewer than 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] Carlos Aguilar-Melchor, Simon Fau, Caroline Fontaine, Guy Gogniat, and Renaud Sirdey. Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain. *IEEE Signal Processing Magazine*, 30(2):108–117, 2013.

[2] Greg Allen and Taniel Chan. Artificial intelligence and national security, 2017.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[4] Aanchal Anand. Colored coins: Bitcoin, blockchain, and land administration. In *Annual World Bank Conference on Land and Poverty*, 2016.

[5] Louis JM Aslett, Pedro M Esperança, and Chris C Holmes. A review of homomorphic encryption and software tools for encrypted statistical machine learning. *arXiv preprint arXiv:1508.06574*, 2015.

[6] Marcella Atzori. Blockchain technology and decentralized governance: Is the state still necessary?, 2015.

[7] Arati Baligi. Understanding blockchain consensus models, 2017.

[8] David Banisar. Stopping science: The case of cryptography. *Health Matrix*, 9:253, 1999.

[9] Brian Barrett. The year encryption won, Jun 2017.

[10] Craig P Bauer. *Secret history: The story of cryptology*. CRC Press, 2013.

[11] Eli Ben-Sasson. Zerocash, bitcoin, and transparent computational integrity, Jan 2017.

[12] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for c: Verifying program executions succinctly and in zero knowledge. In *Advances in Cryptology–CRYPTO 2013*, pages 90–108. Springer, 2013.

[13] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 287–304. IEEE, 2015.

[14] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*, pages 142–157. Springer, 2016.

[15] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, et al. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pages 91–96. ACM, 2016.

[16] Christian Blum and Christina Isabel Zuber. Liquid democracy: Potentials, problems, and perspectives. *Journal of Political Philosophy*, 24(1), 2015.

[17] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.

[18] Nick Bostrom, Allan Dafoe, and Carrick Flynn. Policy desiderata in the development of machine superintelligence.

[19] Miles Brundage, Shahar Avin, et al. Preventing and mitigating the misuse of artificial intelligence, 2017.

[20] Erik Brynjolfsson and Andrew McAfee. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014.

[21] Rebecca Burn-Callander. Skype inventor jaan tallinn wants to use bitcoin technology to save the world, Jun 2016.

[22] Vitalik Buterin. Daos, dacs, das and more: An incomplete terminology guide, May 2014.

[23] Vitalik Buterin. An introduction to futarchy, Aug 2014.

[24] Vitalik Buterin. On public and private blockchains, Aug 2015.

[25] Vitalik Buterin. Visions, part 1: The value of blockchain technology, Apr 2015.

[26] Vitalik Buterin. Visions, part 2: The problem of trust, Apr 2015.

[27] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. *URL https://github.com/ethereum/wiki/wiki/White-Paper*, 2017.

[28] Kamilia Lahrichi in Caracas. Growing number of venezuelans trade bolivars for bitcoins to buy necessities, Dec 2016.

[29] Nate Cardozo. The state of crypto law: 2016 in review, Jan 2017.

[30] HSINCHUN CHEN, WINGYAN CHUNG, Jennifer JIE XU, GANG WANG, YI QIN, and Michael CHAU. Crime data mining: A general framework and some examples. *Computer*, 37(4):50–56, 2004.

[31] Usman Chohan. Assessing the differences in bitcoin & other cryptocurrency legality across national jurisdictions. SSRN, 2017.

[32] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[33] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017.

[34] L Lessig Code. Code and other laws of cyberspace, 1999.

[35] Jeff Coleman. Designing blockchain applications on ethereum, 2016.

[36] James B Comey. Going dark: Are technology, privacy, and public safety on a collision course?, 2014.

[37] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

[38] Scott A Crosby and Dan S Wallach. Efficient data structures for tamper-evident logging. In *USENIX Security Symposium*, pages 317–334, 2009.

[39] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International Symposium on Privacy Enhancing Technologies Symposium*,

pages 235–253. Springer, 2009.

[40] Ksenia Ermoshina, Francesca Musiani, and Harry Halpin. End-to-end encrypted messaging protocols: An overview. In *International Conference on Internet Science*, pages 244–254. Springer, 2016.

[41] Ethereum. Sharding faq, 2017.

[42] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*, pages 436–454. Springer, 2014.

[43] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[44] Klint Finley. A 50 million hack just showed that the dao was all too human, Jun 2017.

[45] Ben Fisch, Daniel Freund, and Moni Naor. Physical zero-knowledge proofs of physical properties. In *International Cryptology Conference*, pages 313–336. Springer, 2014.

[46] Urs Gasser, Nancy Gertner, Jack L Goldsmith, Susan Landau, Joseph S Nye, David O'Brien, Matthew G Olsen, Daphna Renan, Julian Sanchez, Bruce Schneider, et al. Don't panic: Making progress on the" going dark" debate, 2016.

[47] Michael Genesereth. Computational law.

[48] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[49] Bela Gipp, Jagrut Kosti, and Corinna Breitinger. Securing video integrity using decentralized trusted timestamping on the bitcoin blockchain. In *MCIS*, page 51, 2016.

[50] Bela Gipp, Norman Meuschke, and André Gernandt. Decentralized trusted timestamping using the crypto currency bitcoin. *arXiv preprint arXiv:1502.04015*, 2015.

[51] Alexander Glaser, Boaz Barak, and Robert J Goldston. A zero-knowledge protocol for nuclear warhead verification. *Nature*, 510(7506):497, 2014.

[52] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[53] John Wiedhofft Gough. *The Social Contract: a Critical Study of Devel-*

opment. Oxford: Clarendon Press, 1963.

[54] Hamish R Gow, Deborah H Streeter, and Johan FM Swinnen. How private contract enforcement mechanisms can succeed where public institutions. *Agricultural economics*, 23(3):253–265, 2000.

[55] Andy Greenberg. Zcash, an untraceable bitcoin alternative, launches in alpha, Jun 2017.

[56] Stuart Haber and W Scott Stornetta. How to time-stamp a digital document. In *Conference on the Theory and application of Cryptography*, pages 437–455. Springer, 1990.

[57] Robin Hanson et al. Shall we vote on values, but bet on beliefs? *Journal of Political Philosophy*, 2003.

[58] Alex Hern. Revolv owners furious as google shuts down smart home company, Apr 2016.

[59] Alex Hern. Revolv owners furious as google shuts down smart home company, Apr 2016.

[60] Alex Hern. Revolv owners furious as google shuts down smart home company, Apr 2016.

[61] Alex Hern. Revolv owners furious as google shuts down smart home company, Apr 2016.

[62] Alex Hern. Uk government can force encryption removal, but fears losing, experts say, Mar 2017.

[63] Alyssa Hertig. What is a dao?, Mar 2017.

[64] F Harry Hinsley. The influence of ultra in the second world war. In *Cambridge Security Group Seminar*, 1993.

[65] Anna Irrera and Jemima Kelly. Blockchain could save investment banks up to 12 billion a year: Accenture, Jan 2017.

[66] ITU-T. Distributed ledger technologies and financial inclusion, 2017.

[67] Ari Juels, Ahmed Kosba, and Elaine 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*, pages 283–295. ACM, 2016.

[68] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[69] Trevor I Kiviat. Beyond bitcoin: Issues in regulating blockchain tranactions. *Duke LJ*, 65:569, 2015.

[70] Taavi Kotka, Carlos Ivan Vargas Alvarez del Castillo, and Kaspar Korjus. Estonian e-residency: Redefining the nation-state in the digital era, 2015.

[71] Andrew Martin and Ivan Martinovic. Security and privacy impacts of a unique personal identifier, 2016.

[72] Jon Matonis. Wikileaks bypasses financial blockade with bitcoin, Feb 2013.

[73] John J Mearsheimer. *The tragedy of great power politics*. WW Norton & Company, 2001.

[74] R Merkle. Daos, democracy and governance.

[75] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.

[76] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[77] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[78] Jack Peterson and Joseph Krug. Augur: a decentralized, open-source platform for prediction markets, 2015.

[79] Andrew Poelstra et al. Distributed consensus from proof of stake is impossible, 2014.

[80] Nathaniel Popper. *Digital gold: Bitcoin and the inside story of the misfits and millionaires trying to reinvent money*. HarperCollins, 2016.

[81] Nathaniel Popper. A hacking of more than 50 million dashes hopes in the world of virtual currency, Jun 2016.

[82] Nathaniel Popper. Business giants to announce creation of a computing system based on ethereum, Feb 2017.

[83] Rob Price. The 18 companies that control bitcoin in 2016, Jun 2016.

[84] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.

[85] Christian Reitwiessner. zksnarks in a nutshell, Dec 2016.

[86] Christian Reitwiessner. An update on integrating zcash on ethereum (zoe), Jan 2017.

[87] Scott Rosenberg. Can an arcane crypto ledger replace uber, spotify and airbnb?, Jun 2017.

[88] Nathan Saper. International cryptography regulation and the global information economy. *Nw. J. Tech. & Intell. Prop.*, 11:xv, 2012.

[89] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.

[90] Eric B Schnurer. E-stonia and the future of the cyberstate. *Foreign Affairs*, 2015.

[91] Andrew Schotter. The economic theory of social institutions. *Cambridge Books*, 1981.

[92] Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, Frank Kriwaczek, Peter Hammond, and H Terese Cory. The british nationality act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.

[93] Peter Warren Singer. *Wired for war: The robotics revolution and conflict in the 21st century*. Penguin, 2009.

[94] KN Sowmya and HR Chennamma. A survey on video forgery detection. *International Journal of Computer Engineering and Applications*, 9(2):17–27, 2015.

[95] Josh Stark. Making sense of cryptoeconomics, Aug 2017.

[96] Melanie Swan. *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc., 2015.

[97] Jake Swearingen. Something weird (or weirder than normal) is happening at wikileaks, Nov 2016.

[98] Nick Szabo. The idea of smart contracts, 1997.

[99] Don Tapscott and Alex Tapscott. *Blockchain Revolution: How the technology behind Bitcoin is changing money, business, and the world*. Penguin, 2016.

[100] Jason Teutsch and Christian Reitwießner. A scalable verification solution

for blockchains. 2017.

[101] Andrew Trask. Building safe a.i., March 2017.

[102] Andrew Trask. Safe crime detection, Jun 2017.

[103] Misha Tsukerman. The block is hot: A survey of the state of bitcoin regulation and suggestions for the future. *Berkeley Tech. LJ*, 30:1127, 2015.

[104] Masashi Une. The security evaluation of time stamping schemes: The present situation and studies. In *IMES Discussion Papers Series 2001-E-18*. Citeseer, 2001.

[105] Vinod Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 5–16. IEEE, 2011.

[106] Jelle Van Den Hooff, David Lazar, Matei Zaharia, and Nickolai Zeldovich. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, pages 137–152. ACM, 2015.

[107] Paul Vigna and Michael J Casey. Bitcoin for the unbanked. *Foreign Affairs*, 2015.

[108] Gian Volpicelli. How the blockchain is helping stop the spread of conflict diamonds. `http://www.wired.co.uk/article/blockchain-conflict-diamonds-everledger`, Feb 2017.

[109] David Vorick and Luke Champine. Sia: Simple decentralized storage, 2014.

[110] Mark Walport. Distributed ledger technology: beyond block chain. Technical report, United Kingdom Government Office for Science, 2015.

[111] Samuel C Woolley and Philip N Howard. Automation, algorithms, and politics| political communication, computational propaganda, and autonomous agents: Introduction. *International Journal of Communication*, 10:9, 2016.

[112] Aaron Wright and Primavera De Filippi. Decentralized blockchain technology and the rise of lex cryptographia, 2015.

[113] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. The blockchain as a software connector. In *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*, pages 182–191. IEEE, 2016.