NBER WORKING PAPER SERIES

DESIGN CHOICES FOR CENTRAL BANK DIGITAL CURRENCY: POLICY AND TECHNICAL CONSIDERATIONS

Sarah Allen
Srđjan Čapkun
Ittay Eyal
Giulia Fanti
Bryan A. Ford
James Grimmelmann
Ari Juels
Kari Kostiainen
Sarah Meiklejohn
Andrew Miller
Eswar Prasad
Karl Wüst
Fan Zhang

Working Paper 27634
http://www.nber.org/papers/w27634

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
August 2020

The authors wish to thank IC3's industry partners for their support of this work. Thanks also to Jim Ballingall for helpful comments. Ittay Eyal was partially funded by ISF (1641/18) and BSF grants. Giulia Fanti was partially funded by NSF grant CIF-1705007 and ARO grant W911NF17-S-0002. Bryan Ford was partially funded by ONR grant N00014-19-1-2361 and the AXA Research Fund. Ari Juels was partially funded by NSF grants CNS-1704615 and CNS-1933655. Sarah Meiklejohn was partially funded by EPSRC Grant EP/N028104/1. Andrew Miller was partially funded by NSF grant CNS-1943499. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research. Kari Kostiainen, Karl Wüst and Srdjan Capkun were supported (in part) by the Zurich Information Security and Privacy Center (ZISC).

At least one co-author has disclosed a financial relationship of potential relevance for this research. Further information is available online at http://www.nber.org/papers/w27634.ack

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

© 2020 by Sarah Allen, Srđjan Čapkun, Ittay Eyal, Giulia Fanti, Bryan A. Ford, James Grimmelmann, Ari Juels, Kari Kostiainen, Sarah Meiklejohn, Andrew Miller, Eswar Prasad, Karl Wüst, and Fan Zhang. All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission provided that full credit, including © notice, is given to the source.

Design Choices for Central Bank Digital Currency: Policy and Technical Considerations
Sarah Allen, Srđjan Čapkun, Ittay Eyal, Giulia Fanti, Bryan A. Ford, James Grimmelmann, Ari Juels, Kari Kostiainen, Sarah Meiklejohn, Andrew Miller, Eswar Prasad, Karl Wüst, and Fan Zhang
NBER Working Paper No. 27634
August 2020
JEL No. E42,E52,E58,O31

ABSTRACT

Central banks around the world are exploring and in some cases even piloting Central Bank Digital Currencies (CBDCs). CBDCs promise to realize a broad range of new capabilities, including direct government disbursements to citizens, frictionless consumer payment and money-transfer systems, and a range of new financial instruments and monetary policy levers.

CBDCs also give rise, however, to a host of challenging technical goals and design questions that are qualitatively and quantitatively different from those in existing government and consumer payment systems. A well-functioning CBDC will require an extremely resilient, secure, and performant new infrastructure, with the ability to onboard, authenticate, and support users on massive scale. It will necessitate an architecture simple enough to support modular design and rigorous security analysis, but flexible enough to accommodate current and future functional requirements and use cases. A CBDC will also in some way need to address an innate tension between privacy and transparency, protecting user data from abuse while selectively permitting data mining for end-user services, policymakers, and law enforcement investigations and interventions.

In this paper, we enumerate the fundamental technical design challenges facing CBDC designers, with a particular focus on performance, privacy, and security. Through a survey of relevant academic and industry research and deployed systems, we discuss the state of the art in technologies that can address the challenges involved in successful CBDC deployment. We also present a vision of the rich range of functionalities and use cases that a well-designed CBDC platform could ultimately offer users.

Sarah Allen
The Initiative for Cryptocurrencies and Contracts (IC3)
2 West Loop Road
New York, NY 10044
sarahallen@cornell.edu

Ittay Eyal
Israel Institute of Technology (Technion)
Meyer 960
Haifa, 3200003
Israel
ittay@technion.ac.il

Srđjan Čapkun
Swiss Federal Institute of Technology in Zürich (ETH Zurich)
CNB F 102.2
Universitätstrasse 6
8092 Zürich
Switzerland
srdjan.capkun@inf.ethz.ch

Giulia Fanti
Carnegie Mellon University (CMU)
2118 CIC
4720 Forbes Ave
Pittsburgh, PA 15213
gfanti@andrew.cmu.edu

Bryan A. Ford
Swiss Federal Institute of Technology
in Lausanne
BC 210 (Bâtiment BC)
Station 14
CH-1015 Lausanne
Switzerland
bryan.ford@epfl.ch

James Grimmelmann
Cornell Tech and Cornell Law School
2 West Loop Road
New York, NY 10044
james.grimmelmann@cornell.edu

Ari Juels
Cornell Tech
2 West Loop Road
New York, NY 10044
juels@cornell.edu

Kari Kostiainen
Swiss Federal Institute of Technology
in Zürich (ETH Zurich)
CNB F 103.2
Universitätstrasse 6
8092 Zürich
Switzerland
kari.kostiainen@inf.ethz.ch

Sarah Meiklejohn
University College London (UCL)
Gower Street
London WC1E 6BT
United Kingdom
s.meiklejohn@ucl.ac.uk

Andrew Miller
The University of Illinois Urbana-Champaign
461 Coordinated Science Lab
306 N. Wright St. MC 702
Urbana, IL 61801
soc1024@illinois.edu

Eswar Prasad
Dyson School of Applied Economics
and Management
Cornell University
301A Warren Hall
Ithaca, NY 14853
and NBER
eswar.prasad@cornell.edu

Karl Wüst
Swiss Federal Institute of Technology
in Zürich (ETH Zurich)
CNB F 101
Universitätstrasse 6
8092 Zürich
Switzerland
karl.wuest@inf.ethz.ch

Fan Zhang
Cornell University and Cornell Tech
2 West Loop Road
New York, NY 10044
fz84@cornell.edu

Contents

Contents 3

1 Introduction 5
- 1.1 Benefits and risks . . . 6
- 1.2 Paper roadmap . . . 7

2 Overview from a Banking Perspective 9
- 2.1 Technical definitions . . . 10
- 2.2 Why issue a CBDC? . . . 11
- 2.3 Implications for the international monetary system . . . 13

3 Ledger Infrastructure 14
- 3.1 Information security foundations: The Confidentiality, Integrity, Availability (C-I-A) triad . . . 15
- 3.2 Distribution and decentralization . . . 16
- 3.3 State machine replication for distributed ledgers . . . 17
  - 3.3.1 Centralized ledgers . . . 19
  - 3.3.2 Centralized but verifiable ledgers . . . 20
  - 3.3.3 Semi-centralized ledgers . . . 21
  - 3.3.4 Decentralized ledgers with central-bank monetary control . . . 22
- 3.4 Scalability to large transaction volumes . . . 24

4 Account and Identity Management 25
- 4.1 Who manages accounts? . . . 25
- 4.2 Approaches to digital identity verification . . . 27

5 Digital Wallets 31
- 5.1 User authentication . . . 32
- 5.2 Transaction authentication . . . 34
- 5.3 User interfaces . . . 37

6 Privacy and Transparency 38
- 6.1 Identity privacy . . . 39
- 6.2 Transaction privacy . . . 42
- 6.3 Privacy and decentralization . . . 44
- 6.4 Privacy and compliance . . . 46

7 Smart Contracts 47
- 7.1 Striking a balance between safety and extensibility . . . 49
- 7.2 Off-chain protocols and advanced cryptography . . . 53
- 7.3 Smart contracts as a two-layer architecture . . . 53

8 Secure Hardware **54**
8.1 Brief introduction to secure hardware . . . 54
8.2 Limitations of secure hardware technology . . . 56
8.3 Problems of simple use . . . 58
8.4 Better ways to leverage secure hardware . . . 59
8.5 Summary and recommendations . . . 61

9 Opportunities for Novel Financial Technology **62**
9.1 Implementing monetary policy . . . 62
9.1.1 Transparency . . . 63
9.1.2 Non-fungible money . . . 64
9.2 Monetary policy transmission . . . 65
9.3 Selective review of academic literature . . . 66
9.4 Smart contracts: realizing other novel capabilities . . . 66
9.5 Summary . . . 69

10 Legal Considerations **69**
10.1 Jurisdiction . . . 70
10.2 Compliance . . . 70
10.3 Privacy . . . 71
10.4 Fraud and mistake . . . 73
10.5 Liens . . . 75
10.5.1 Collection . . . 76
10.5.2 Locking . . . 77
10.5.3 Notice . . . 77
10.6 Tracing . . . 78
10.7 Taxation . . . 79
10.8 Conclusion . . . 79

11 Overview of Libra and Digital Yuan **80**
11.1 Libra . . . 80
11.2 The digital yuan . . . 82
11.3 What patent filings reveal about the digital yuan . . . 83

12 Summary Position **85**

References **89**

1 Introduction

Central Bank Digital Currency (CBDC)—fiat currency issued by central banks in digital form—has progressed in the past few years from a bold speculative concept to a seeming inevitability.

More than 80% of central bank respondents to a Bank for International Settlements survey in 2019 reported engagement in CBDC projects [1]. One in ten of these banks, representing approximately one-fifth of the world's population, deemed it likely that they would offer CBDCs within the next three years. The People's Bank of China, whose plans are well in advance of that of other major economic powers, has begun to pilot a digital yuan [2]. Hearings on CBDC have taken place this year in the U.S. House Committee on Financial Services [3]. The European Central Bank has initiated a project to explore CBDC development [4] while Sweden (an E.U. but not Eurozone member), has begun testing a CBDC known as the e-krona [5].

At the same time, a Facebook-initiated fiat-backed cryptocurrency called Libra has raised the prospect of an industry alternative. Regulator concerns about the project [6] and (perhaps incorrect) speculation that it has catalyzed CBDC development [7] highlight Libra's overlap in goals with CBDC.

Various forms of CBDC have in a sense existed for years, but as wholesale facilities available exclusively to financial institutions [8]. What is striking and potentially transformative about many recent CBDC initiatives is their retail focus, that is, their aim of democratizing central bank account holdings to individual consumers or, at a minimum, making digital central bank liabilities available to households and businesses. Our focus in this paper is on retail CBDCs.

While the goals of cryptocurrencies such as Bitcoin differ dramatically from that of CBDC, they offer evidence of feasibility and technical idea for retail deployment of digital currency. Their technical foundations underpin Libra, have to some extent influenced CBDC design plans, and strongly inform the findings and recommendations of this paper.

Paper scope: This position paper investigates and explains the design choices, mainly technical, but also financial and legal, that central banks will unavoidably encounter in their exploration of CBDCs. Contributing authors include experts in computer science, economics, and law whose research and practical experience has a strong bearing on the design of digital currencies. We highlight not just choices, but challenges that we believe will constitute the main impediments to CBDC deployment or define the main limiting factors in CBDC realization.

This work is geared toward readers who may be only lightly conversant with the technical concepts behind digital currencies. It does not assume specialized technical knowledge. It can also serve as a reference work, as individual sections are largely self-contained.

At a minimum, we suggest reading section 2 for background on CBDC from a banking perspective and basic terminology and section 12 for a summary position of the authors of this work. In this introductory section, we briefly review the main

benefits and risks of CBDC (section 1.1) and present a roadmap of the paper (section 1.2).

1.1 Benefits and risks

Among the main potential benefits spurring central bank exploration are:

- *Efficiency:* CBDC can reduce friction in existing payment systems, potentially lowering the monetary cost and increasing the speed of transactions while ensuring finality. The prospect of instantaneous payments has proven attractive in the U.S., for instance, in view of the challenges of disbursing financial aid during the current pandemic [9].

- *Broader tax base:* CBDC can potentially bring more economic activity into the tax net, limiting tax evasion and boosting tax revenues. Moreover, the traceability of digital transactions would inhibit the use of CBDC for illicit purposes such as money laundering and terrorism financing.

- *Flexible monetary policy:* The zero lower-bound constraint on monetary policy (interest rates set by central banks) could in principle be relaxed, with a central bank instituting a negative nominal interest rate by reducing CBDC account balances at a pre-announced rate. Similarly, CBDC would ease the implementation of non-distortionary helicopter drops or withdrawals of central bank money (without relying on fiscal transfers).

- *Payment backstop:* CBDC could act as a backstop to private sector managed payment systems, avoiding breakdown of payments systems in times of crisis of confidence and rise in counterparty risk.

- *Financial inclusion:* CBDC could serve as a gateway for unbanked and underbanked individuals to have access to electronic payment systems and, potentially, to other financial products and services as well.

We highlight an additional benefit in this paper, namely opportunities for novel financial technologies, particularly for regulators.

The many potential benefits of CBDC should be weighed against a number of potential risks, both financial and technical. They include:

- *Disintermediation of the banking system:* Many CBDC plans seem to be gravitating toward a two-layer architecture (see, e.g., [10]), in which the CBDC itself serves as a basic functional layer, while existing non-governmental financial institutions interface manage a second layer that interfaces with users. Nonetheless, by reducing transaction frictions and possibly even providing interest-bearing accounts [8], CBDCs could disintermediate significant swaths of the banking system.

- *Miscalibration of government involvement:* One acknowledged benefit of a two-layer architecture is the opportunity for financial institutions to innovate on top of a CBDC [10]. A CBDC design that arrogates to a central bank activities such as payments that can be cheaply and efficiently be managed by the private sector could limit innovation. At the same time, systemic risks and incompatibilities could arise without adequate central bank involvement.

- *Financial risks due to lack of regulatory expertise and capacity:* With increased speed and efficiency—and especially financial innovation—come new risks, financial and technical, many enumerated above. Regulators may struggle to develop the tools and expertise to address these risks in the face of a dramatic change in the basic operation of the financial system.

- *Loss of privacy:* Given the complexity and performance limitations of current privacy-enhancing technologies, it seems likely that a true retail CBDC will expose new forms of sensitive information to its operators. CBDC designers should consider legal and technical mitigations from the outset.

- *Technological vulnerabilities or entrenched design mistakes:* Even with conservative design, CBDCs will represent a technical experiment, whose risks of information security failures and fundamental design mistakes should not be underestimated.

Our focus in this paper on the technical choices and risks involved in CBDC deployment, as well as key financial and legal considerations, emphasizes exploration of the last two of these risks.

1.2 Paper roadmap

The foundation of a digital currency is a digital record of all of the transactions that have taken place in the system. Such a record is often referred to as a *digital ledger*, and may be viewed abstractly as a digital bulletin board to which all transactions in the currency system are posted. The set of transactions in the ledger cumulatively determine the *account balances* in the system. The set of all account balances at a given time may be regarded as a snapshot of what is sometimes called the *state* of the ledger.

To ensure against ambiguity in account balances at a given time, the ledger must also include a *sequencing* of transactions—generally based on their time of receipt—that determines their order of execution. In the view of a ledger as a bulletin board, new transactions may be thought of as appended to an ever-growing, ordered transaction list.[1] See fig. 1 for a conceptual diagram.[2]

[1] A ledger may be thought of as a database with an append-only structure, i.e., in which no transactions are deleted. Most databases, however, allow records to be deletes, and are thus not digital ledgers in sense that the term is currently used. Such databases, despite their weaker data-integrity assurances, could be used to realize a CBDC. Many of the findings in this paper would still

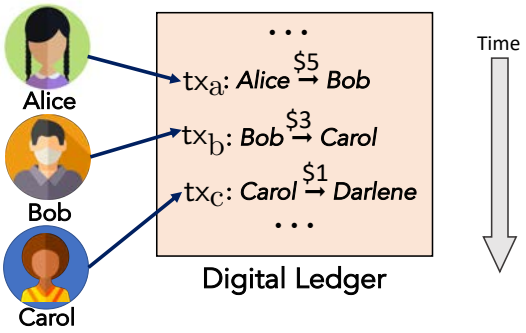

Figure 1: Conceptual diagram of a digital ledger with a sequence of three of its posted transactions. (Clipart attribution: flaticon.com.)

In section 3 ("Ledger Infrastructure"), we describe the various types of digital ledgers in common use today and their underlying technologies, classifying them broadly according to their degree of *centralization* or *decentralization*, i.e., the diversity of the set of entities in which system control is vested. We discuss the security and privacy features offered by different types of digital ledgers.

A retail CBDC assigns account balances to individual users, necessitating a regime for *account management*, with a supporting notion of *identity*, concepts treated in section 4 ("Account and Identity Management"). Critical design elements include the choice of entities to verify users' real-world identities and translate them into digital form and the mechanisms by which the system *authenticates* enrolled users, i.e., permits use of an account only by its assigned users. To access the CBDC system, users need specialized applications, typically referred to as *digital wallets*, and discussed in section 5 ("Digital Wallets"). Wallets serve as the endpoints for user authentication, and provide *user interfaces* that guide users in their interaction with the CBDC and allow them to initiate transactions, view account balances, etc. Wallets may also perform *transaction authentication* facilitating the CBDC's verification of the validity of submitted transactions.

The term "bulletin board" suggests a publicly visible medium. Indeed transactions in cryptocurrencies such as Bitcoin are readable by any user, resulting in strong *transparency*. Central banks, however, may have more stringent requirements for the *privacy* of users transactions, and are unlikely to embrace a fully public model. Digital ledgers can be designed to reveal information selectively and/or only to authorized entities.

The tension between transparency and privacy is the focus of section 6 ("Privacy and Transparency"), which discusses how privacy relates to *identities* and *transactions*. That section explains limitations in the widely embraced privacy model of

be relevant in this case.

[2]The diagram in fig. 1 also illustrates the significance of transaction ordering and why unambiguous ordering is important. Suppose that balances prior to tx_a are: Alice: \$5; Bob: \$0; Carol: \$0. If processed in the displayed order, all transactions are valid, i.e., have adequate balances in originating users' accounts. Were tx_a processed *after* tx_b, however, then both tx_b and tx_c would be invalid.

pseudonymous accounts, discusses techniques for enforcing strong privacy, and addresses issues of privacy as it relates to regulatory compliance.

The transactions recorded on digital ledgers can transfer money between accounts, but can also perform more complex actions. *Smart contracts*, discussed in section 7 ("Smart Contracts"), are computer applications that execute on top of and can greatly expand the capabilities of digital ledgers and thus CBDCs. We explain how the history of smart contracts in cryptocurrencies illuminates potential benefits, including the opportunities to create powerful, novel financial instruments, but also highlights the potential pitfalls of enriched CBDC capabilities.

We discuss *secure hardware* in section 8 ("Secure Hardware"), specifically a powerful new set of security features available in recent-model computer chips. Little known outside information security circles, secure hardware can serve as a powerful addition to the security architecture of a variety of high-trust systems, including CBDCs. We discuss the limitations and recognized vulnerabilities of secure hardware and consequently where it is and is not appropriate to incorporate it into CBDC architectures.

The transaction information in a CBDC could offer unprecedented visibility into monetary flows, given financial regulators new insights into the functioning of a national economy. We discuss such opportunities in section 9 ("Opportunities for Novel Financial Technology"), as well as ways that enhancements to CBDC systems, such as non-fungible tokens (e.g., currency with attached spending conditions) and smart contracts can serve as vehicles for innovative financial interventions.

A number of legal considerations will have a significant bearing on CBDC design choices, as discussed in section 10 ("Legal Considerations"). Existing laws, particularly in the U.S., would seem to offer fairly wide latitude in the degree of privacy afforded by a CBDC. Legal requirements for remediating erroneous or fraudulent transactions and enforcing liens will require careful consideration of appropriate technical provisions in a CBDC and may motivate the enactment of supporting legislation.

While still in their infancy, given their maturity by comparison with related projects, study of the digital yuan and Libra can help elucidate critical design choices. We outline what is known publicly about their technical designs in section 11 ("Overview of Libra and Digital Yuan"), and also discuss hypothetical capabilities revealed in published patent applications relating to the digital yuan.

2 Overview from a Banking Perspective

The basic functions of money are that it serves as a unit of account, medium of exchange, and store of value. While money is associated in the popular mind with physical cash issued by central banks, broader monetary aggregates that serve some of these functions include bank deposits created by commercial banks when they make loans. Thus, while currency banknotes and coins are physical forms of money, much of the stock of money in modern economies is already in digital form. Even digital central bank money has already existed for a long time. Electronic balances held by commercial banks (and, occasionally, other financial institutions) at central

banks, referred to as reserves, are used to facilitate payments and settlement through interbank payment systems managed by the central bank.

The specific innovation that we consider in this paper is the replacement of central bank issued money that can be used for retail transactions with their digital counterparts, which have come to be referred to as CBDC. In short, CBDC are fiat currencies issued by central banks in digital form in place of, or as a complement to, physical currency (banknotes and coins).

One simple form of CBDC is e-money. This can take the form of specific amounts downloaded to a mobile phone app by designated financial institutions (in exchange for cash or transfers from bank accounts) and that can be used for making payments at approved businesses. In an alternative formulation, all agents in an economy would have access to central bank accounts, where the balances could in principle be interest-bearing. The central bank would in effect become the manager of a sophisticated payments system that would also allow it, depending on the structure of this CBDC, to implement conventional and unconventional monetary policy in nonstandard ways and, in some respects, more effectively.

The first option is easiest to implement and, in combination with mobile phones that have become ubiquitous even in low-income economies, has significant potential to improve financial inclusion and reduce dependence on cash. The second option is technologically and conceptually more complicated but has greater potential to be scaled up into a payments system that serves as a backup to the private payments infrastructure.

One concern about central bank deposit accounts is the possible disintermediation of the banking system–a subject that will be explored in more detail later in this paper. Recognizing this risk, some central banks that are experimenting with CBDC are taking a hybrid two-tier approach. Under this approach, central banks would disseminate CBDC to commercial banks–just as they now do with cash–and commercial banks would distribute these to individuals and businesses by setting up and managing digital wallets.

Some governments, such as those of the Marshall Islands and Venezuela, have ventured to develop what they refer to as *Official cryptocurrencies*. The Venezuelan government has created the Petro, a digital currency backed by the country's oil reserves, which is ostensibly a cryptocurrency that could help avoid financial sanctions imposed by the United States. It is far from clear whether such digital currencies can be considered the equivalent of fiat currency and how they would help get around international financial sanctions.

2.1 Technical definitions

Kumhof and Noone [11] provide a useful definition of CBDC to distinguish it from reserves and cash. They define CBDC as electronic central bank money that: (i) can be accessed more broadly than reserves, (ii) has functionality for retail transactions, (iii) can be interest bearing (with a rate different from that on reserves), and (iv) has a separate operational structure relative to other forms of central bank money.

Yao [12] offers a more technologically-oriented definition, positing that a CBDC is

"a credit-based currency in terms of value, a crypto-currency from a technical perspective, an algorithm-based currency in terms of implementation, and a smart currency in application scenarios." He argues that cryptographic technology is essential for security and credibility of the DFC. He also notes that CBDC is not just a digital version of cash but has the potential to make money "smarter."

Bjerg [13] lays out a broad definition of CBDC as electronic, universally accepted, central bank issued money and discusses three possible scenarios. In the first one, the CBDC serves as electronic cash, complementing cash and bank deposits and, thus, fulfilling the role of medium of exchange. The central bank would maintain parity and free convertibility among CBDC, cash, and bank deposits. In a second scenario, the CBDC would serve as universal reserve and fulfill the role of store of value, replacing cash. The central bank would maintain parity but not free convertibility between CBDC and bank deposits. In a third design, CBDC serves as sovereign account money and as the unit of account, potentially replacing bank deposits. In this scenario, the central bank takes the sole responsibility of creating and issuing money in the economy, maintaining free convertibility between CBDC and bank deposits. The central bank could effectively use monetary policy to create or destroy liquidity in the system based on the state of the economy.

Bordo and Levin [14] present two designs for CBDC as a medium of exchange. In the first, the central bank circulates "CBDC tokens," supported by distributed ledger technology for ownership verification and payment transactions. In the second, the central bank maintains "CBDC accounts" that facilitate electronic holding of funds for individuals and follow a simple debiting and crediting transaction protocol that is instantaneous and costless. The authors then explore three alternatives for a secure store of value. First, similar to paper currency, the central bank would issue CBDC with "constant nominal value" and earning zero interest. This would constrain the central bank from implementing a negative nominal interest rate. Second, the central bank would retain "stable real value" of CBDC through price level indexation of CBDC, which would also constrain policy at the zero lower bound. Third, the central bank would provide an interest-bearing CBDC where the interest rate would be positive in a growing and stable price economy. The authors argue that such a CBDC would serve as a stable unit of account with the help of flexible price-level targeting monetary policy.

The sampling of definitions above suggests that there is no clear consensus yet on the definition of a CBDC, with both conceptual and technological issues still being sorted out. Both of these sets of issues are tied in to the motivation for a central bank to issue a CBDC.

2.2 Why issue a CBDC?

The key motives for issuing retail CBDC range from broadening financial inclusion to increasing the efficiency and stability of payment systems. In Sweden, an economy where the use of cash is fast disappearing, the central bank's consideration of retail CBDC, in the form of an e-krona, seems to be driven primarily by concerns about financial stability. The sharp decline in the use of cash for retail payments

has occurred in tandem with a shift toward privately-managed payment systems and consolidation among a small number of commercial participants, payment services, and infrastructures.

The Riksbank notes than an e-krona could alleviate the problem of concentration of the payments infrastructure and also its potential vulnerability to loss of confidence. The digital currency would be based on a separate infrastructure that would also be open to private agents willing to offer payment services linked to the e-krona. The general public would have access to the e-krona with both payment suppliers and fintech companies having access to the network. Thus, an e-krona system would promote competition, innovation, and financial stability.

A primary motivation for emerging market economies to consider issuing CBDC seems to be related to financial inclusion. An app-based CBDC that takes advantage of mobile technologies can increase access to financial services for the poor, rural households, and other segments of the population that may be underserved by the banking system.

There are a number of ancillary benefits to a CBDC. Paper currency is vulnerable to counterfeiting. CBDCs could in principle reduce this risk, although the risk of electronic counterfeiting on an even more massive scale through hacking is a major concern for governments that intend to take this route.

Another potential advantage of a CBDC is that it would discourage illicit activity and rein in the shadow economy by reducing the anonymity of transactions now provided by the use of currency banknotes, a point made forcefully by Rogoff [15], especially in the context of high-denomination banknotes. This would also affect tax revenues, both by bringing more activities out of the shadows and into the tax net and also by enhancing the government's ability to collect tax revenues more efficiently.

Ensuring compliance with anti-money laundering/combating financing of terrorism (AML/CFT) regulations has been a major challenge for government authorities. The elimination of physical cash could assist in these efforts, although the likely shifting of illicit fund transfers to decentralized payment systems and intermediated through anonymous, decentralized cryptocurrencies could vitiate this progress. This is one reason why central banks might seriously consider issuing CBDCs so they can retain control of or at least oversight over payment systems that could as easily be used for illicit as for licit purposes.

These benefits come at the potential cost of loss of privacy in commercial transactions if these can be intermediated only through private or government-managed electronic payments systems. While various encryption technologies in principle allow users of retail CBDC to retain privacy, it is likely that these are subject to the same technological vulnerabilities as nonofficial cryptocurrencies, where privacy has been difficult to ensure.

Some of the trade-offs between physical and electronic forms of fiat currency issued by central banks are analyzed by Mishra and Prasad [16] in the context of a simple general equilibrium model. The key differences between these two forms of central bank-issued outside money include transaction costs (lower for CBDC), possibilities for tax evasion (higher for cash, but with a positive probability of being caught and penalized), and nominal rates of return (zero for cash; potentially positive or negative

for CBDC). They show the conditions under which cash and CBDC can co-exist and also show how different combinations of government policies, such as the level of taxes and the penalty for being caught undertaking tax evasion, can influence the relative holdings of cash and CBDC. The model provides a framework that can eventually be extended to evaluate conditions under which different forms of government-backed and privately-issued currencies can coexist, conditional on the attributes of each of those currencies and also government policies.

2.3 Implications for the international monetary system

The advent of CBDC, cryptocurrencies, and other new financial technologies could have implications over the long run for certain aspects of the international monetary system. One of the major benefits of improved electronic payment and settlement systems that would go with the proliferation of digital currencies is the increase in speed and security of transactions, along with a reduction in their costs. This would mark a substantial improvement for settlement of trade-related transactions as well as remittances. Even cross-border settlement of other types of financial transactions could benefit from these developments. DLTs offer the potential for reliable tracking of different stages of trade and financial transactions, reducing one of the frictions associated with such transactions. Such changes might simply increase the efficiency and lower the cost of transactions routed through banks and other traditional financial institutions rather than displacing such institutions.

Both banks and nonbank financial institutions could expand the geographical scope of their operations across national borders using the new technologies. This creates new challenges for supervision and regulation. One complication is the lack of clarity about the domicile of informal financial institutions and the geographical locus of the supervisory authority of national regulators. The second is the potential accentuation of cross-border financial stability risks as more institutions operate across national borders. Some of these challenges could be overcome by the greater transparency of transactions if they are conducted using a public DLT or if the regulator has access to the relevant private ledgers.

For emerging market economies, the expansion of conduits for cross-border financial flows with greater efficiency and lower costs could be a double-edged sword, making it easier for them to integrate into global financial markets but at the risk of higher capital flow and exchange rate volatility. Such volatility, in part related to spillovers of monetary and other policies from the U.S. and other advanced economies, has often caused significant stresses for corporate and sovereign balance sheets in these economies. These challenges could become greater if new payments systems and digital currencies increase both the volumes and fluctuations in cross-border capital flows and make capital controls less potent, adding to such volatility. The intensification of global financial cycles would not only engender more capital flow and exchange rate volatility, but could also constrain monetary policy independence, even for central banks that practice inflation targeting backed up by flexible exchange rates. New channels for transmitting payments across borders more quickly and cheaply are likely to make it more difficult to regulate and control capital flows.

3 Ledger Infrastructure

The goal of a digital currency system is to track the balance of its users, allowing each to transact only her coins. One cannot map the technique of physical currency transaction to the digital world. A coin cannot be a simple file, and a transactions cannot be a transmission of the file from one user to another: Had it been done this way, the sender could have kept her copy, thus keeping the coin while also sending it.

Instead, contemporary digital currency systems maintain a global state, comprising the balances of all their users. This includes everything from banks' per-client balance tables to cryptocurrencies like Bitcoin [17] and Ethereum [18]. Updates to the state are called *transactions* – these could be simple transfers of funds or interaction with *smart contracts*. The transactions are serialized in a single *ledger*. The state of the system is the result of processing the transactions in the ledger according to their order. The transactions are typically aggregated into so-called *blocks*, each containing many transactions, and the blocks are linked to form a chain, imaginatively called a *blockchain* (Figure 2).

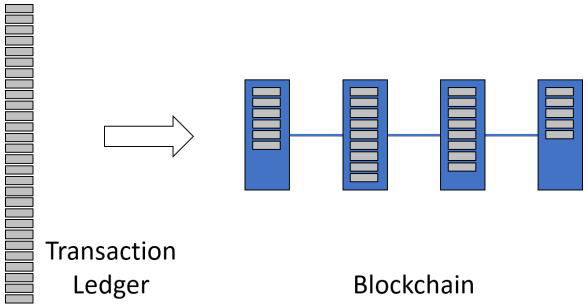

Figure 2: Each block contains transactions and the block order determines the global transaction order.

The system should allow participants to add blocks in a serial order, so the system progresses in a well defined manner. It should also allow all and only transactions that abide by its predefined rules, and prevent the removal of a block, which would have implied a reversal of history.

To avoid vulnerability to the crash or misbehavior of one or several machines, the blocks and state should be *replicated*, i.e., stored and processed concurrently by multiple machines. The challenge is thus to orchestrate these machines so they all agree on this order and behave like a single coherent machine, despite network latencies and arbitrary misbehavior by a subset of them. This is called *State Machine Replication* (*SMR*) in distributed systems literature. Even in a centralized setting, there are hardware failures, so it is prudent IT system design to incorporate multiple nodes for fault tolerance.

The design of the underlying SMR and networking layers affects the performance and the security of the system, but perhaps most significantly defines its ethos – how decentralized the system is. This choice determines how open the system is to participants, and how much it is in the control of one or a few entities, which can redefine its behavior, stop its operation, withhold certain transactions, etc.

There is a spectrum of decentralization designs, from bank balance tables, through semi-centralized systems like Ripple [19] and Libra [20], to cryptocurrencies, and there are clear trade-offs to be considered when designing a CBDC. The rise of cryptocurrencies since 2009 incited rapid advancement across this spectrum.

In the rest of this section, we first outline basic information security principles that are important in understanding and evaluating digital currency designs (§3.1), then refine the distinction between distribution and decentralization (§3.2), and proceed to review the design choices for the SMR mechanism (§3.3) and their trade-offs.

An independent challenge is to increase the system throughput beyond the capacity of a single machine. In order to accommodate the needs of a CBDC, the system should process a large number of transactions quickly. We briefly discuss approaches to scaling ledgers to large transaction volumes (§3.4), such as by *sharding* or partitioning of system state.

3.1 Information security foundations: The Confidentiality, Integrity, Availability (C-I-A) triad

No digital currency will remain operable and in use for long without satisfying certain fundamental *information security* properties. In particular, classical information security principles define three orthogonal and complimentary "dimensions" of information security: *confidentiality*, *integrity*, and *availability*, often collectively referred to as the *C-I-A triad*, illustrated in Figure 3.

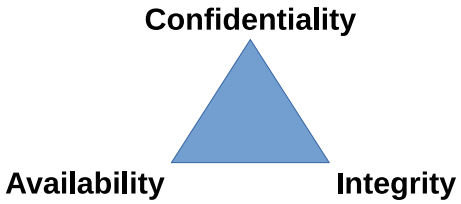

Figure 3: The three main protection goals of classic information security: Confidentiality, Availability, and Integrity.

In brief, confidentiality means that the information system does not leak information to those who should not have access to it. Integrity means that the system should store information correctly and produce correct results to computations, allowing neither to be tampered with maliciously for example. Availability means that the system should respond to users promptly when requested to retrieve data or perform some action, such as committing a digital currency transaction.

But how do we ensure that each of these dimensions of information is satisfied, and what types and degrees of costs are we willing to incur to guarantee these information security properties? This is where techniques for distribution and decentralization, fundamental to cryptocurrencies and CBDCs alike, come into play.

3.2 Distribution and decentralization

Distributed systems: A *distributed* system fundamentally consists of multiple devices communicating and coordinating over a network. There are innumerable varieties of distributed systems and countless functions they can perform. The most relevant type of distributed system for a CBDC is, of course, a *distributed ledger*, or group of devices cooperating to maintain a transaction history. Using *state-machine replication* or *consensus* algorithms as described below in Section 3.3, the devices comprising a digital ledger maintain copies or *replicas* of the transaction history and keep them synchronized. This replication protects the availability of the ledger by ensuring that even if some replicas fail, the other devices can continue servicing user transactions.

Traditional client/server and cloud computing infrastructures tend to use distribution in this way primarily to protect availability, and place the greatest investments toward this goal. Geographically distributed systems, for example, distribute the replicas of a service (such as a ledger) across data centers located in different cities or regions, so that a network or power outage affecting one entire data center does not make the entire service unavailable. Though widely distributed, cloud infrastructure is still typically highly *centralized* in that it is owned and controlled by one central authority (the cloud provider).

Decentralized systems: A *decentralized* system, in contrast, is a type of distributed system whose composite devices are *not* under control of a single, central authority. Decentralizing a system across independent authorities in principle reduces the amount of trust we must place in each, and similarly limits the damage any one authority can cause if it is compromised or misbehaves in some way.

There are many different and often-conflicting ways in which a system may be *decentralized*, however, and just as many fiercely-debated criteria for deciding whether and how much a system is actually "decentralized." In particular, decentralization often refers to some combination of *role separation*, *trust dispersal*, and/or *threshold trust*, as we outline below.

Role separation: Perhaps the weakest, but useful and ubiquitous, form of "decentralization" is the division of a process into multiple qualitatively-different functions carried out by multiple authorities serving in different roles. The corporate accounting practice of requiring one person to write checks, and another person sign them, is classic example of role separation within an organization. The division between the roles that central banks and commercial banks play in classical economics – the former managing the national money supply and the latter managing customer relationships – is a large-scale example of role separation. The Bank of England's CBDC proposal to delegate the account management role to a commercial Payment Interface Provider (PIP) [10] is an example of (limited) decentralization via role separation in a CBDC design.

Trust dispersal: When one role in a distributed system may be played by many independent authorities, each serving only a small subset of the total user population, trust is *dispersed* among these authorities. The dispersal of a nation's governmental powers across many regional and local governments, each having jurisdiction mainly only over its own residents and territorial domains, is a classic pre-digital example of this form of decentralization.

In a CBDC design in which many different companies implement the PIP role on behalf of their customers and *only* their customers, only those customers of a given PIP in principle need to trust that PIP. The trust that the entire system collectively places in the PIP role, therefore, is dispersed among the many companies serving that role, limiting the damage any single (smaller) compromised PIP can cause. Each individual user must fully trust their chosen PIP, however, and if one is compromised then the damage to that PIP's unlucky customers – in terms of confidentiality, integrity, and availability – may be severe and difficult to limit. Further, if one or more of the authorities playing such a role becomes "too big to fail" – e.g., serving too much of the user population – then even the global protection ensured by trust dispersal may be limited.

Threshold trust: Finally, systems may be decentralized such that users need not "choose" and then fully trust a single authority. Users instead individually or collectively split their trust across several authorities independently serving in the same role, so that no single authority or small coalition have unlimited power or authority over *any* user. A board of directors or parliament is a classic organizational embodiment of threshold trust, for example, whose members are collectively trusted but no single member can act alone. A distributed ledger, spread across a consensus group of servers operated by independent companies in a federation, provides threshold trust at least in terms of the ledger's availability, so that the ledger remains available to serve *all* users even if *any* one or a limited number of member servers go offline.

Consensus or state-machine replication of this form does *not* necessarily protect users from integrity or confidentiality failures in these servers, however. Any single compromised server may be able to fake or rewrite history, unless the consensus algorithm is also designed to tolerate malicious or *Byzantine* failures – a property that most public cryptocurrencies strive to provide but which many "permissioned" blockchains fail to ensure. Similarly, any single compromised server may leak any confidential data that server may have had access to, unless the confidential information is separately protected via *threshold cryptography* mechanisms, for example, as we discuss in Section 6.3. Therefore, it is important to understand *which* properties of the C-I-A triad discussed above a given ledger design protects with threshold trust, and which properties remain potentially vulnerable to "weakest-link" failures.

3.3 State machine replication for distributed ledgers

The level of decentralization of the ledger state-machine-replication infrastructure can be roughly divided into three categories. On one end there is the centralized option (§3.3.1), where the central bank runs the system itself. This is arguably a good

	Centralized	Semi-Decentralized	Decentralized
Archetypal Example	Amazon Quantum Ledger Database [21]	Libra [20]	Bitcoin [17]
Performance	Excellent, full control of infrastructure	Good, SMR with many participants	Challenging, active area of research
Censorship	Easy: single operator	Possible: particularly if operators are in the same jurisdiction	Hard: operators might not even be identified
Rewind	Easy: can be done quickly by single operator	Hard: takes longer for operators to agree, implying longer history to revert and worse violation	Extremely hard: requires cooperation by majority of possibly unidentified operators

Table 1: Centralized vs. Decentralized Infrastructure

starting point, as it is the easiest to manage and builds on classical system design. However, this design choice misses central goals of a CBDC. As the centralized CBDC can unilaterally withhold transactions or even change the rules or revert history, the users are not getting the same guarantees as in cash. Indeed, these aspects are closer to those of payment application.

As a first step, history revision can be prevented by providing users with commitments that their transactions are irrevocably committed in centralized but verifiable ledgers (§3.3.2). But one can do better.

In a semi-decentralized, or *permissioned* option (§3.3.3), a consortium of entities collaboratively runs the system. Such a design denies full control of the system from small subsets of the operators, but still allows for centrally-coordinated changes, which can be an advantage for legal purposes and is often desirable by customers.

Finally, a fully decentralized option (§3.3.4) would implement a CBDC on an infrastructure similar to that used by cryptocurrencies like Bitcoin – a design that has demonstrated unprecedented robustness in the wild. The decentralized option implies a lack of centralized control; this is arguably the goal of a CBDC, in contrast to payment applications and digital bank accounts, but due to legal and regulatory considerations it is not trivially acceptable in the CBDC context.

We review these options below, and summarize them in Table 1.

3.3.1 Centralized ledgers

The most straightforward approach to implement a CBDC ledger infrastructure is in a centralized fashion. For resilience the system should still be distributed, replicating the state among multiple servers. Classical solutions to this challenge have been studied for decades [22], providing protocols that overcome crashes of a subset of the servers, as well as unexpected behavior due to bugs or an attack [23]–[26].

Crash faults imply a danger only to availability (in the C-I-A triad). State machine replication (SMR) algorithms that protect against crash faults have well-understood solutions with mature, efficient implementations already widely-used in cloud computing environments for example (e.g., [27], [28]).

Servers that behave arbitrarily for whatever reason – such as by being hacked or under control of a compromised insider – can violate not only a ledger's availability but also its integrity. It is important to understand that mere replication does not protect a ledger's integrity against arbitrary *Byzantine* failures such as hacking or insider attacks. Even in a widely-replicated ledger, a single compromised server behaving maliciously might be able to rewrite history illegally, "print money" without detection, or trick a targeted victim into thinking a transaction has been committed while everyone else sees a conflicting transaction, or no transaction, on the ledger. State machine replication algorithms that protect against Byzantine failures, or *Byzantine fault tolerance* (BFT) mechanisms, have also received a great deal of research attention (e.g., [25], [28]). Byzantine fault tolerance mechanisms tend to be more complex and less mature than SMR algorithms tolerating only crash faults, however, in part because their development has not been driven by the already-ubiquitous cloud computing paradigm.

Challenges remain to avoid *common mode failures* among multiple system nodes (servers), such as a power or network outage affecting several (or all) nodes at once. The typical solution for geographically-localized common mode failures – such as outages caused by lightning strikes, floods or other natural disasters, or long-distance roadside cables accidentally cut by backhoe operators – is *geo-replication*: locating nodes in different data centers widely spread geographically.

Another type of common mode failure affecting Byzantine replication mechanisms is software bugs. If all replicas run the same implementation of the same algorithm and this implementation has an exploitable vulnerability, then a hacker or malicious insider could compromise all (or a threshold of) the nodes at once to compromise the ledger's integrity, despite its nominal protection against Byzantine failures. Using diverse software implementations on each server is one solution to this common mode failure risk, albeit one incurring considerable development cost [29], [30].

With centralized or semi-centralized ledger designs, a small number of servers is usually sufficient. This small number is relatively easy to coordinate and achieve good performance, as the machines can quickly propagate messages among each other. However, in a fully-centralized design in which the choice of nodes and their operation are all under direct control of one authority such as the central bank itself, that authority – or a malicious insider – can potentially change the rules at will, roll back system state or rewrite history, and censor or delay transactions. These capabilities

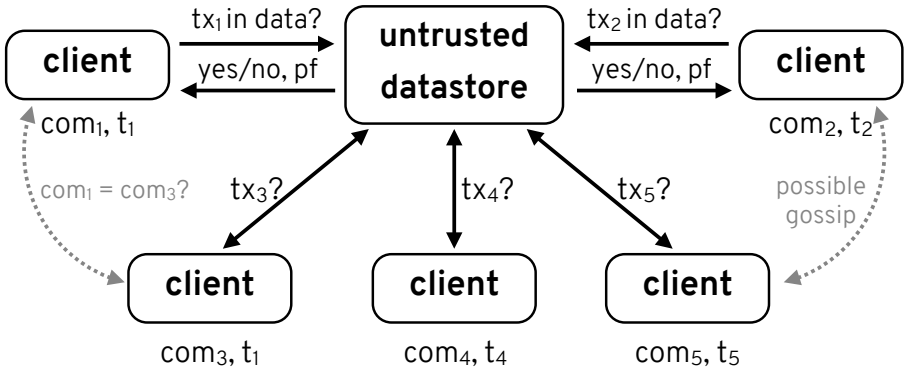

Figure 4: An overview of the interactions involved in an authenticated data structure. Clients do not trust the server in possession of the data. When the server tells a client whether or not a transaction is in the dataset, it provides a proof that this response is correct, which the client can check against a commitment to the data that it maintains and updates when needed. To ensure that the server is providing a consistent view of the dataset to all clients, a form of gossip is needed (which need not involve direct client-to-client communication). As one example (on the left), clients with commitments representing the same point in time (t_1) may want to check that these commitments are identical.

may be desirable from a management perspective, but may be undesirable from a perspective of robustness or public trust.

3.3.2 Centralized but verifiable ledgers

Even in a centralized setting, there are still ways to limit the trust that clients need to place in the servers maintaining the ledger. In particular, *authenticated data structures* [31]–[34] (ADSs) provide a method by which a server, in possession of some data, can prove things about that data to a client who does not necessarily trust the server to give accurate answers. For example, a client receiving money in a CBDC may want to ensure that the transaction in which they are being paid has gone through before providing any goods or service; i.e., they want to check for the inclusion of this transaction in the ledger. Rather than have the server simply tell the client if the transaction is included or not, in an ADS the server also provides a cryptographic proof of inclusion that the client can verify. The main guarantee of an ADS is that it should be impossible for a server to provide a valid proof for a response that does not accurately reflect the data it has stored.

To satisfy this requirement, clients must have some *commitment* to the data stored by the server, which the clients can then check the proof against. This commitment acts as a cryptographic fingerprint of the state of the entire dataset at a given point in time and is updated as the data changes; crucially, while it covers the entire dataset its size is a small constant that is independent of the size of the dataset. A central challenge of ADSs is that clients cannot be sure that their commitment is a canonical representation of the data stored by the server. In a *split-view attack*, for example, a

malicious server may attempt to present different clients with commitments to different data, and thus make them believe that different transactions are or aren't included in the ledger. In the extreme case, a server could store completely different data for each client, and thus provide each one with a different commitment; this would not prevent them from providing valid proofs, but the results would be meaningless as there would be no global and unique representation of the server's state.

To prevent—or at least detect—split-view attacks, clients must typically engage in a *gossip* protocol [35], by which they can learn about the commitments held by other clients and thus ensure that they have the same view of the data. Options for gossip protocols range from client-to-client communication [36] to having the server post the commitment on a public blockchain [37], [38]. An alternative protection against split-view attacks is to employ a *witness cosigning* protocol [39], in which the primary server maintaining the ADS first obtains cryptographic co-signatures on each fingerprint commitment from a threshold of independent witness servers before publishing or returning that fingerprint to clients. The witness servers need not repeat or thoroughly check the primary server's work, but merely attest that they have witnessed and cosigned at most one cryptographic view for a given state version. Witness cosigning can eliminate the bandwidth, energy, and potential privacy costs to clients participating in gossip protocols, and can protect clients that are disconnected or whose network connectivity is under an adversary's control [40], [41]. The main cost is that the witness servers must be deployed and managed, introducing limited decentralization, and clients must trust a threshold of witnesses to behave correctly, rather than trusting only in each other and the network as with gossip.

It is also possible to use ADSs to create a more distributed ecosystem, as is perhaps best exemplified by the Certificate Transparency (CT) project.[3] In the CT ecosystem, a variety of organizations log certificates signed by a Certificate Authority, and clients ensure using the interaction described above that all the certificates they see appear in at least one of these logs. This is crucial due to the central role that certificates play in providing trust in the Internet ecosystem, in the form of underpinning encrypted communication (HTTPS), and due to the known cases in which CAs have misissued certificates [42], [43]. A decentralized network of auditors and monitors are then free to interact with these log servers to check, respectively, that they are properly storing the data (e.g., they are not carrying out split-view attacks) and that the certificates they are storing are valid according to some global set of rules (e.g., that a Certificate Authority has not misissued a certificate). This type of architecture may accommodate a CBDC [44], in which certificates are replaced with transactions and the role of log servers is played by commercial banks or other entities with an existing stake in the ecosystem.

3.3.3 Semi-centralized ledgers

To reduce centralization, the CBDC can instead be operated by a larger set of independent parties chosen or approved by the central bank. Superficially, the solution

[3] https://certificate.transparency.dev/

is similar to the centralized approach – instead of directly operating the different nodes, the central bank chooses the entities that run them. In practice, however, this approach is quite different, as the central bank forgoes its total control of the system. With dozens or even hundreds of independent operators and adequate technical protection against Byzantine faults in the state machine replication scheme, no single party or group of parties (below a certain size) can change the rules, perform censorship or roll back the system state. Nevertheless, since the operators are all chosen by the central bank, that central bank can facilitate an agreement of all parties to perform arbitrary changes. This allows the operators to comply with regulatory and legal requirements, as well as revert illegal operations due to attacks. Indeed, the central bank can deploy complex governance mechanisms, where different subsets of the operators can force a decision or exert veto power.

Therefore, despite the distribution of operation, the semi-decentralized approach lacks the true decentralization of physical cash. Users of a semi-centralized CBDC have less control over their funds than with cash. However, this control is an advantage in certain perspectives: Law enforcement agencies can compel the operators to enforce monetary controls, censor transactions, etc., particularly if they are all within the same jurisdiction.

The basic design considerations of protocols for semi-centralized systems are similar to the centralized case. Indeed, the distributed systems literature puts them in the same category, implementing a replicated state machine by a set of predefined nodes. However, the large number of nodes make classical solutions problematic – the amount of communication they require is typically quadratic in the number of nodes, making them prohibitively slow among many independent parties.

Building on ideas from cryptocurrencies, recent work [28], [45]–[47] proposes state machine replication solutions that reduce most communication to be linear in the number of nodes, and hence practical even with a large number of operators. This is the approach chosen by Facebook's Libra (§11).

3.3.4 Decentralized ledgers with central-bank monetary control

The most decentralized approach available is to use a blockchain infrastructure similar to that of cryptocurrencies. Unlike those cryptocurrencies, the CBDC is under centralized monetary control, but anyone can join and operate the system. This approach is called *permissionless*, as operators do not need permission from the central bank to join. Although the system operators can still change the system rules (known in Cryptocurrency as a *fork*), in a permissionless system a fork requires a wide agreement among independent parties. This makes controversial changes unlikely, but also takes away the control of the central bank.

On the positive side, decentralized blockchains demonstrate unprecedented robustness – Bitcoin has been running continuously without interruption[4] for over a

[4]In two famous events [48], [49] there have been issues with Bitcoin's blockchain layer. But, if anything, they demonstrated its robustness: although some operators significantly deviated from the prescribed protocol, the impact on the users has been inconsequential.

decade. A decentralized implementation also prevents monetary controls and transaction censorship – the open membership implies that the different operators are not subject to the decision of any central entity. This is a desirable property of an instrument that strives to replace physical cash.

However, this decentralization also implies no central control even when it is necessary, e.g., no reversal of transactions in case of mistakes, and no prevention of operator misbehavior like front-running [50], [51]. The setup would therefore have to include means aimed directly at these scenarios, which are active areas of research.

Finally, it is not immediately clear how to implement the decentralized approach in the context of a CBDC. To ensure the security of such an open system, incentives are used. System operators, often called miners as in Bitcoin, should receive rewards to incentivize them to follow the desired protocol. Unlike cryptocurrencies, here the central bank determines the inflation rate, and the rewards for the miners. Although it does not choose the miners in the open system, the central bank in this setup is still uniquely powerful, as it determines the reward rules.

The common technique for securing decentralized blockchains is Proof of Work [52], [53], as used in Bitcoin, Ethereum, etc. The idea is that miners expend physical resources, typically electricity, to participate in the protocol and receive rewards. A coalition that controls less than a threshold of the resources will maximize its revenue by following the rules [54]–[58]. Moreover, if the rewards are sufficient [59], an attacker cannot perform a Denial of Service (DoS) attack (i.e., stop the system) with less than 50% of the mining power.

Other alternatives have also been explored (e.g., [60], [61]), with *Proof of Stake* (*PoS*) [62], [63] gaining the most traction. In PoS, the operators are the system participants, i.e., they have a stake in the system. While there remain challenges in understanding the incentive mechanism of PoS and its security, PoS avoids the energy expenditure necessary in PoW [64], which is unacceptable for CBDC.

A decentralized cryptocurrency's openness to a large number of permissionless participants does not imply absolute protection against ledger manipulation or censorship. In practice the distribution of mining power in PoW cryptocurrencies, or the distribution of stake in PoS currencies, has often proven to be so concentrated that only the top 2, 3, or 4 miners or stakeholders account for a majority of voting power, and thus could in principle collude (perhaps secretly) to censor or manipulate the ledger.[5] Further, any permissionless cryptocurrency allowing open participation is potentially vulnerable to attacks by any adversary sufficiently resourceful and motivated to deploy large-scale computational or financial resources in an attack, even just temporarily. An attacker might short-sell a currency in other markets in order to "bet against" its value, for example, just before launching an attack that deliberately violates the system's economic rationality assumptions [65]. Many flavors of such attacks have been explored [66]–[69], and actual 51% attacks on real permissionless cryptocurrencies have become increasingly common in practice [70].

The openness of a permissionless system also typically implies slower performance, as it takes longer for the participants to realize who is actually participating. Nev-

[5]See for example https://bitcoinera.app/arewedecentralizedyet/.

ertheless, even in an open system it is possible to achieve throughput limited only by network properties [46], [71]–[74], and good latency with advanced block topologies [73], [75].

In conclusion, we believe a fully-open, permissionless design option is not at this point a natural choice for a CBDC, due to its extreme lack of central control. For example, Libra has recently stated [20] that "...a key concern expressed by regulators in a number of jurisdictions, including the Swiss Financial Market Supervisory Authority (FINMA), is that it would be challenging for the Association to guarantee that the compliance provisions of the network would be maintained if it were to transition to a permissionless network where, for example, no due diligence is performed on validators." We therefore believe that there remain technical, legal and regulatory questions to answer before this permissionless approach can be adopted in a CBDC.

3.4 Scalability to large transaction volumes

Whichever ledger design approach is taken, another element to address is how to scale performance beyond the capacity of a single server. While even in a fully decentralized architecture there are protocols that allow for arbitrarily high transaction throughput, all those transactions must be processed, and the workload can grow beyond the capacity of any one server. The openness of permissionless systems like Bitcoin does not help the system handle increased transaction load, because each miner is replicating, and hence repeating, all of the work of processing *all* transactions, leaving the maximum processing rate fixed regardless of participation.

There are several complementary approaches to this question. The first [76], [77] is to split the state into multiple parallel ledgers. Each ledger is operated by different servers, allowing the system capacity to grow by the number of parallel chains. However, special care must be taken when using this approach to make sure the security of each chain does not deteriorate compared to a single chain. Additionally, a split into parallel chains implies that special, often slow, protocols must be used when a transaction spans the state of multiple chains, e.g., making a payment from an account in one chain to an account on another. This approach applies when the state can be cleanly split into chains with little interaction, and the number of participants is large enough to allow splitting their roles among the different chains.

Another approach [78] is to operate a ledger with some arbitrary protocol, but rather than using a single machine per node, replace it with several interconnected servers that operate as a single high-performance node. This approach maintains the security properties of the original protocol and allows for scaling according to the resources available to the different nodes. It applies when it is acceptable to rely on more resources per node operator, which is likely the case in the CBDC setting.

We discuss in Section 7.2 methods to reduce ledger load by offloading transactions to direct peer-to-peer channels. These rely on an efficient underlying ledger and are independent of its implementation.

4 Account and Identity Management

Since a CBDC cannot achieve its maximum usefulness unless ordinary individuals can hold and use the digital currency, this raises the key question of who should be responsible for managing accounts and satisfying associated responsibilities such as identity-checking for "know your customer" (KYC) compliance. This section first explores the question of who should be responsible for account and identity management, then briefly surveys current and emerging approaches to digital identity and how they may (or may not) be relevant to CBDCs. The important topic of identity privacy will be covered later in Section 6.

4.1 Who manages accounts?

Central banks traditionally do not maintain accounts for or enter into business relationships with individuals, only with banks. In this way, central banks effectively delegate the task of managing individual accounts and customer relationships to the commercial banks. Allowing individuals to open and use CBDC accounts directly with the central bank, therefore, would be a "new business" for most central banks, bringing with it many account- and identity-management challenges and potential risks. Providing individual accounts directly with the central banks may be a concern for citizens as well: for example, many potential customers may be more inclined to entrust their personal and financial information to a local business than to a remote government agency that they can at best hope to contact by phone or online.

Accounts in cryptocurrencies: Most decentralized cryptocurrencies such as Bitcoin are technically designed to address – or perhaps to *avoid* – the account and identity management problem in a different way: by defining "accounts" not in terms of human identities but in terms of pseudonymous cryptographic key pairs. Bitcoin or Ethereum accounts are simply random-looking strings that represent cryptographic *public keys*. Anyone who knows the corresponding, mathematically-related *private key* associated with that account can spend the currency it holds. This property is what makes decentralized cryptocurrencies a cash-like character, with corresponding advantages and disadvantages. The cryptocurrency *miners* primarily responsible for maintaining and securing the ledger can avoid managing or checking traditional individual identities.

The perceived privacy that decentralized cryptocurrency accounts provide is attractive to many cryptocurrency holders, although the use of pseudonymous key pairs alone offers only weak privacy, due to the many available de-anonymization attacks discussed later in Section 6.1. On the other hand, this disconnect between purely cryptographic accounts and human identities has in part given cryptocurrencies a shady, "underground" character, making it difficult for individuals and businesses to use and convert cryptocurrencies directly while ensuring legal compliance. The irreversible character of cryptocurrency transactions also gives individuals no clear recourse path if their cryptocurrency is stolen, due to a hacked wallet for example – an important usability and security issue discussed further in Section 5.

Cryptocurrency exchanges: This gap between cryptographic keys and human identities has in part driven the development of centralized *exchanges* and related businesses intended to bridge this gap. A cryptocurrency exchange typically allows customers to trade one or more traditional currencies for one or more cryptocurrencies. To make this possible, exchanges typically maintain a traditional account and business relationship with each individual customer, and they either take on directly or further delegate the identity-checking tasks required to ensure compliance with the prevailing legal and financial policies.

On the positive side, exchanges can make the use of cryptocurrency more convenient to customers more familiar with traditional banks, and potentially more legitimate and compliant in reality and/or perception. On the negative side, in practice most exchanges are centralized third parties that must be trusted with the custody of users' cryptocurrency balances. Such centralized, custodial exchanges can potentially lose much or all of their customers' funds if they fail, are successfully hacked, or are internally compromised. Centralized exchange hacks have historically occurred numerous times, and at an accelerating rate [79], [80]. Non-custodial decentralized exchanges are possible and starting to appear, but currently tend to be less mature and usable, and more complex in operation.

Delegated account management in CBDCs: Much like central banks traditionally delegate the task of account management and identity checking to commercial banks, and like cryptocurrency miners have come to delegate this task implicitly to exchanges and other cryptocurrency holding and investment businesses, it may be natural for a CBDC to delegate this task similarly. This is one of the key roles of the Payment Interface Providers (PIPs) in the Bank of England's CBDC proposal, for example [10]. We believe this delegation approach is reasonable and in-line with historically-proven role separations, as it would avoid the need for the central bank to enter the unfamiliar business role of individual account management, and it would allow the Payment Interface Providers to innovate competitively in the way they provide these individual-facing services. The Digital Yuan also appears to be adopting this model, as discussed later in Section 11.2.

A possible downside of this delegation approach, however, is that many potential innovation opportunities may be left "on the table" and undeveloped, if the commercial Payment Interface Providers do not find it profitable or in their business interests to compete on the basis of certain aspects of account and identity management. Because banks generally avoid trying to "compete on security" or distinguish themselves from their competitors on grounds of security for fear of shaking customers' perceived trust in banks in general, for example, delegation of account management to commercial entities may in practice be ineffective at driving innovation in security-related areas. Similarly, the competitive pressure most high-tech businesses currently feel to collect and monetize data is likely to undermine competitive incentives for Payment Interface Providers to innovate in privacy-related areas. Thus, for some aspects of account and identity management especially including security and privacy, strong regulation and standards-setting – whether directed by a central bank or other gov-

ernment agencies or independent voluntary federations – may be necessary to ensure quality and innovation in these areas.

4.2 Approaches to digital identity verification

Beyond the issue of who manages individual accounts and identities, another question is how a customer's identity is verified for accuracy and legal compliance. There is a rich body of both academic literature and practice on the complex and challenging problem of digital identity verification. This complexity boils down to the fundamental problem that no known technology in our digital ecosystem – whether a device, algorithm, protocol, or service – can identify a particular "real human" with complete security. Instead, what we have is a plethora of mechanisms for associating digital accounts with imperfect *proxies* for individuals. Such proxies include information tokens such as cryptographic private keys, private information about the individual concerned (e.g., mother's maiden name, childhood pet, favorite film), hardware devices such as two-factor authentication (2FA) or multi-factor authentication (MFA) tokens [81], [82], traditional physical-world credentials such as ID cards and passports, other digital identity proxies such as E-mail addresses and phone numbers, biometric templates, or presence and interconnections in a social trust network. All of these identity proxies may work sometimes, but all have important costs, limitations, and critical failure modes. We briefly review a few of these approaches here and discuss their promise, trade-offs, and potential relevance to identity management for CBDCs.

In-person identity checking: Traditional banking has typically relied on in-person identity verification, requiring the customer to present a government-issued ID or passport at a branch office, often together with other evidence such as utility bills, in order to open an account. If banks or other financial institutions with local branches prove willing to play the role of PIPs for a CBDC, then they will be able to continue relying on in-person verification for CBDC accounts just as they do for traditional accounts. With both banks and their customers rapidly shifting towards mostly- or all-digital relationships, however, the days of in-person verification being the dominant approach may be numbered.

Online identity checking: To adapt to the demands of the digital age, numerous companies have started offering online *video identification* services.[6] These services typically ask customers to present themselves and one or more suitable forms of traditional ID over a video chat session. Using machine-learning and video face-recognition techniques [83], these services attempt to verify that the presented ID "looks" genuine and appears to match the face of the person holding it. These algorithms must also address challenges such as distinguishing between an actual, live person and a static image or video recording of one that an identity thief might be using to pose as the victim. When the algorithm cannot verify the ID with sufficient certainty, it may forward the video session to a human operator.

[6]Examples include IDnow, Fully-Verified, eID, and WebID.

This approach is attractive from a cost perspective as long as the algorithm can decide most cases automatically without requiring human involvement. The use of such algorithms presents many poorly-understood and underappreciated risks, however. Any complex algorithm such as this involving machine learning is almost certain to have a non-negligible false-positive rate, incorrectly accepting false ID cards as real, or accepting the wrong person holding it, or accepting a recorded image or algorithmically-generated *deep fake* [84], [85] of the victim supposedly presenting their ID. A determined and sophisticated fraudster is likely to be able to exploit even a small false-positive rate, through many automated attempts, while suffering little risk of getting caught or effectively punished – especially if they are launching their attack anonymously from a foreign country via a network proxy.

An important risk that tends to be underappreciated by the proponents of machine learning algorithms for identity verification is that the bad guys have artificial intelligence and machine learning algorithms too [86]. Whether in playing chess [87], Jeopardy [88], Go [89], [90], solving CAPTCHAs [91], [92] or creating deepfakes [84], [85], our consistent historical experience is that when we set up games between increasingly-sophisticated machine-learning attackers and increasingly-sophisticated machine-learning defenders, we find ourselves in an arms race in which the machine attacker sooner or later wins consistently over the real person. It may thus not be long before machine-learning identity checkers consistently accept sophisticated deepfakes while rejecting most real people, and an ever-larger percentage of online society – and perhaps even much of its commerce – is fake [93]–[95].

Weak digital identity proxies: Many non-financial applications in the online ecosystem rely on weaker identity proxies, such as E-mail addresses, phone numbers, IP addresses, or simply asking users to solve CAPTCHA puzzles [91], [96]. These weak identity proxies are rightfully not usually considered adequate for financial purposes, in part because they only rate-limit, rather than reliably deter, abuses such as the creation of false identities. All of these weak identity proxies can be faked or purchased by a determined and resourceful abuser at varying costs. E-mail addresses are practically free, for example, especially now that many E-mail account services, including Apple, allow their users to create effectively unlimited disposable E-mail pseudonyms for privacy and spam control.[7] CAPTCHAs can be broken via machine learning [91], [92] or social engineering [97], [98], or can be outsourced to countries with inexpensive labor [99].

Mobile phone numbers can be slightly stronger identity proxies, since many countries require customers to show ID when signing up for a calling plan, but anonymous prepaid plans still exist nevertheless. In any case, sophisticated hackers and identity thieves can exploit the weakly-protected SS7 signaling protocol to hijack a victim's phone number, together with the many digital services that use SMS challenges to reset account passwords [100], [101]. In summary, most of the identity proxies popular in the online ecosystem merely increase the cost of abuse somewhat without reliably

[7] See for example Hide My Email for Sign in with Apple, Temp Mail, Guerrilla Mail, FakeMail, ThrowAwayMail.

deterring it, and thus are inadequate for financial applications such as CBDCs.

Biometric identity: Biometrics are often proposed as a strong technology-based solution to digital identity challenges. India's Aadhaar program is both a showcase and testbed of this approach, having registered over a billion people via iris and fingerprints [102]–[104]. The use of biometrics for digital identity presents many challenges and risks that should not be underestimated, however. Biometrics are effectively "passwords you can't change" after something goes wrong [105], [106]. Biometric algorithms provide only approximate matches against biological characteristics that can change over time, be obscured, be destroyed accidentally or intentionally, and can be faked either physically or digitally in various ways [107], [108]. Even in the best circumstances, biometric matching algorithms inevitably exhibit both false-positive errors (incorrect acceptance of non-matches) and false-negative errors (incorrect rejection of true matches). Error rate estimates in the case of Aadhaar imply that hundreds of thousands of records could be duplicates [104]. Even the most precise and hence apparently-secure biometric, namely DNA, is already subvertible by an identical twin – and may soon be readily subvertible via increasingly-efficient synthesis of organoids [109], from stolen stem cells – or eventually, perhaps, the DNA residue we leave constantly in our physical environments.

The use of biometrics also presents profound privacy issues, especially when used for biometric *identity* as in India's Aadhaar program, as opposed to the biometric *authentication* features that have become commonplace in mobile personal devices. With biometric authentication, a device records one or at most a few biometric templates of authorized users, then later performs "one-to-one" or "one-to-a-few" matches against those stored templates when the user wishes to authenticate. The stored templates generally need not and should not ever leave the device, mitigating the most severe privacy concerns, and the common practice of disabling biometric authentication after a few failed attempts mitigates the risk of an attacker abusing the false-positive error rate through many brute-force attempts or other experimentation-based fakery.

In biometric *identity* systems like Aadhaar, in contrast, a registration service must not only record biometric templates for later authentication, but must also *compare* the templates against those of all the other – potentially billions of – already-registered individuals. Aadhaar requires this *deduplication* process in order to detect attempts by one person to register multiple identities and multiply the benefits they can obtain from the state's social "safety-net" services, for example. The need for users to be able to register on one device at one office and then authenticate to a different device at a different office, together with the need for detection of duplicates, fundamentally means that the biometric templates *must* be exported from the registering devices and be collected in a (typically centralized) database for later authentication and deduplication queries. This use of biometrics for identity thus raises much more serious security and privacy concerns [110], [111], as the biometric template database becomes an incredibly attractive target for hackers and foreign state adversaries alike. The need for a one-to-billions comparison in deduplication amplifies the effective false-positive rate correspondingly: a seemingly tiny Aadhaar-compliant false-positive rate

of 0.01% for iris recognition actually means that each user might incorrectly match up to 100,000 others in the billion-user dataset. Finally, any single registration device that is hacked or under malicious control might be used to synthesize false biometric identities or impersonate real users. Most biometric templates previously thought to be irreversible have been proven otherwise [112], and cryptographic *secure sketch* algorithms have seen little adoption in part because they generally require the design of wholly new matching algorithms [113], [114].

In summary, while biometrics have legitimate uses when applied carefully in constrained applications such as mobile device authentication, we urge extreme caution in uses of biometrics for digital identity in security- and privacy-sensitive applications such as CBDCs.

Social trust networks: The basic human practice of "identifying" people – and deciding whether, how much, and for what to trust them – long predates modern government identity practices. Social or community trust remains an important identification factor in many parts of the world where government is weak or mistrusted, playing an important role in microfinance programs for example [115]. It should therefore be no surprise that social trust approaches to identity has long been of interest in the privacy-focused and often anti-government "cypherpunk" movement [116], [117], which first arose decades before but now overlaps heavily with the decentralized cryptocurrency community. The most well-known identity technology based on social trust is the "Web of Trust" concept introduced in the 1990s as part of the *Pretty Good Privacy* (PGP) encryption tool [118], [119]. Other social-trust identity technologies such as SPKI/SDSI [120], [121] were also proposed as public-key cryptography matured, but none proved usable enough to become widespread [122]. Web-based social media platforms such as Facebook and LinkedIn eventually popularized the social approach to identity [123], albeit with more emphasis on convenience and much less on privacy or strength of trust.

Identification based on social trust presents many practical issues. One principle challenge is that social identities tend to be easy and cheap to fake, especially for sophisticated automated algorithms, which has led to much of the social-identity ecosystem being essentially fraudulent [93]–[95]. While an important body of research has focused on algorithms for detecting or neutralizing false identities in social networks [124]–[127], it is not clear that actual trust networks have the properties needed for these algorithms to work [128], [129]. This is especially true when social media users are incentivized to inflate their "connectedness" or "follower counts" artificially through practices such as link farming [130], [131], and even to synthesize plausible content automatically [132]–[135]. For these and other reasons, social trust does not seem like a viable approach to identity for CBDCs – except perhaps in countries where existing government identification practices are weak or corrupt and *real-world* social trust may offer the only viable starting-point to identity.

Self-sovereign identity: One approach that has received significant attention recently, in the blockchain/cryptocurrency community especially, is the notion of *self-*

sovereign identity [136], [137]. In brief, the idea is to build a decentralized identity ecosystem allowing users to collect digital *attestations* of identity attributes from participating individuals or institutions, and then subsequently to reveal or *prove* those attributes selectively to other individuals or institutions demanding identification [138], [139]. Users might collect in their digital wallets attestations to attributes such as name, address, birthdate and other personal data, degrees and certificates earned, citizenships or memberships, etc. Institutions providing these attestations might include governments (for verifying government-issued identity attributes for example), academic institutions (for verifying grades and degrees earned), financial institutions (for verifying credit or other financial history), and so on. Ventures such as Sovrin are attempting to implement and build a self-sovereign identity ecosystem [140], [141], supported by some industry initiatives [142] and standardization efforts [143].

Self-sovereign identity is promising in that it allows many institutions beyond governments to assist with identity verification by issuing attestations, and because it in principle gives users control over how much and which specific attributes they're comfortable revealing to a particular relying party or to facilitate a particular transaction. The most commonly used illustration is that to enter a bar or nightclub, a user might need only to prove that they are above the legal drinking age, and not to reveal any other identity attributes. Self-sovereign identity is currently incomplete and immature, however, leaving many questions and concerns about how it will evolve in practice. For example, it remains unclear how many institutions (government or otherwise) will enter the business of providing attestations to users, how secure these attestations will be, or how widely they will be accepted. The privacy-enhancing "self-sovereign" principle could also be undermined if most relying parties demand that users reveal their real name and other personal information as a condition for doing business, as seems inevitable for financial use-cases. And if government-issued ID proves to be the most common or widely-trusted basis for users to obtain attestations to these personal attributes, then self-sovereign identity may prove to be mostly just a slightly-different technological mechanism for online identity-checking as discussed above. For these reasons, while self-sovereign identity approaches are worth watching, they currently appear to be neither mature enough, nor of sufficient value beyond that of standard identity-checking, to be a viable identity basis for CBDCs at the present time.

5 Digital Wallets

Digital wallets are the software applications through which users interact with a system of digital assets, such as currency. A digital wallet typically allows users to view their balance in one or more accounts, make payments, receive currency or digital assets from other users, and sometimes to trade assets or execute other financial transactions. The design and functionality of digital wallets are crucial not only for usability but also for security (users do not like to lose their money) and for privacy.

Popular money-transfer applications such as Venmo or Alipay may be viewed as digital wallets, and banking apps on mobile devices often include wallet-like functionality. These various applications, however, support user management of currency within existing financial infrastructure. They do not create or rely on fundamentally new representations of money such as CBDC. They do, however, provide a rough picture of the features and user experience that digital wallets for CBDCs would need to offer to see widespread adoption.

Software wallets for decentralized cryptocurrencies often resemble these money-transfer applications in terms of their user experience. Cryptocurrency wallets often differ in one essential respect, though: the cryptographic keys that authorize funds transfer may be stored *on the user's personal device itself* rather than being entrusted to a remote service. This property makes cryptocurrency wallets more cash-like, in principle eliminating the user's need to depend on and trust in a centralized financial provider. This dependence on the wallet device rather than a financial provider for key custody presents the similarly cash-like downside of exposing the user to the risk of unrecoverable financial loss if the wallet or device hosting it is compromised or lost. Motivated in part by the extreme security sensitivity of digital wallets, they also come in special-purpose hardware formats that function quite differently and take advantage of the hardware security practices discussed in Section 8.

In our discussion of digital wallets here, we assume for simplicity that the only digital asset the wallet manages is a CBDC, but note that digital wallets can in principle play a role in controlling a broad range of assets (cryptocurrency, digital tokens, smart contracts, digital cats [144], etc.). The capability of a digital wallet to hold many types of assets can itself present both significant opportunities and risks in its own right, as exemplified by the "ICO bubble" that was technologically enabled by Ethereum's ERC-20 standard [145], [146]. Thus, the question of whether a CBDC is typically held in and used via special-purpose wallets purpose-built to the CBDC, or generic wallets designed to manage a broader class of digital assets, presents important issues and questions to be considered carefully.

Digital wallets provide three main types of functionality: *user authentication*, *transaction authentication*, and a *user interface* for financial transactions. We explore each of these functionality areas in turn.

Figure fig. 5 shows the components and workflow of a digital wallet and can be referred to while reading this section.

5.1 User authentication

To ensure that access to digital wallet assets are limited to the authorized user or users, the wallet must *authenticate* a user. It must ensure that only the owner or a delegate acting on her behalf is able to operate the wallet or access the currency it controls.

Software wallets often use simple passwords or PINs for this purpose. With increasing reliance on mobile devices, though, it is common for wallets to authenticate users biometrically. Specifically, a user can gain access to her assets by possessing the particular device on which the wallet has been installed and successfully perform-

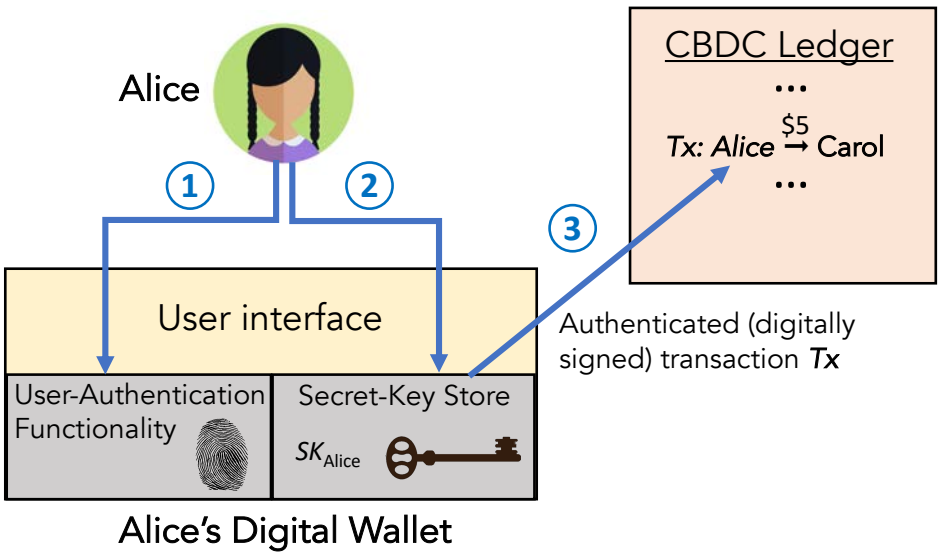

Figure 5: A digital wallet and example transaction workflow. In this example, a user Alice sends $5 to another user, Carol. Alice interacts with the wallet through its user interface. She first ① authenticates herself to the wallet. She then ② instructs the wallet to send $5 to Carol, whom she identifies by means of an account number or username. Her wallet ③ uses a secret key associated with Alice to digitally sign a transaction Tx specifying payment of the $5 and sends Tx to the CBDC system for inclusion on the CBDC ledger.
Note that this figure is strictly conceptual: To ensure privacy, users would be identified on the ledger by means of account numbers or pseudonyms, or for full anonymity might have their identities cryptographically concealed. (See section 6.1.) Additionally, Tx might be processed by a financial intermediary prior to transmission to the CBDC.

ing biometric authentication, e.g., fingerprint or face recognition.[8] Such biometric authentication is typically performed natively by mobile devices.

The goal of user authentication here overlaps with, but is slightly different than that of identity verification (see section 4.2), which generally aims to prove something about the the *real-world identity* of a user. Identity verification may be a precondition of wallet registration in some systems, but a wallet, once created, is in a sense agnostic to the user's identity: Securing the wallet requires only ensuring that the wallet is accessed by the original user (or a delegate).

In the loosest sense of the term, a "digital wallet" may be hosted by a *custody service*, meaning that it takes the form of an account with a service provider such as a financial institution, and need not be accessed through a particular device or instance of a software application. Custody services are used by many holders of cryptocurrency who wish to avoid the technical complexities – and potential security risks – of direct asset control.

[8]Such authentication schemes are sometimes called "two-factor" authentication: They involve a combination respectively of "something-you-have" and "something-you-know" factors [147].

5.2 Transaction authentication

Once authenticated, a user can cause her wallet to perform transactions on currency in her account(s), e.g., sending money to another person. Her wallet must then create and send a transaction message T to the CBDC system for processing—for example, for inclusion on the CBDC's digital ledger.

As a basic security requirement, the CBDC must ensure that transactions are issued authentically on behalf of the users upon whose assets they operate. If Alice issues a transaction T paying \$5 to Bob, the system must ensure that Alice herself authorized T.

In a typical online banking system, the processes of user authentication and transaction authentication are merged. If Alice logs into Piggy Bank, Ltd. (i.e., performs user authentication) and sends \$5 to Bob, the bank can identify the transaction as valid because it has already authenticated Alice. The setting here is simple: The entity that authenticates the user is identical with the one that manages the money she operates on.

A CBDC could operate in this way, requiring users to log onto a web platform to perform transactions. Alternatively, in a two-layer CBDC architecture in which financial intermediaries interface with users, an intermediary can submit transactions on behalf of users it has authenticated. Approaches of this kind where the CBDC operator and/or intermediaries vouch for the authenticity of users' transactions has the benefit of conceptual and design simplicity, but also has notable drawbacks.

The main drawback is the existence of a *single points of compromise* that is large and attractive to profit-motivated hackers. An adversary that compromises the infrastructure of any single financial intermediary can forge transactions from any of its users. Similarly, a financial institution could unilaterally freeze the funds of a user. Should the CBDC be able to confirm transactions unilaterally, then it would constitute a single point of compromise for the entire system.

Central to the design of cryptocurrencies is an alternative form of transaction authentication in which transactions are *digitally signed* in a cryptographic sense by the owners of the currency they transmit. In account-based cryptocurrencies (e.g., Ethereum), each account has an associated public key for verifying the validity of transactions signed by the account holder using a corresponding private key.[9] Only validly signed transactions are included on the blockchain / ledger.

If users manage their own private keys, then the use of digital signatures vests account control directly in the users' hands. Even if a financial intermediary is compromised, because it does not hold users' private keys, it cannot sign on their behalf, and thus cannot forge transactions from user accounts. Hybrid models are also possible (see, e.g., [148]) in which both a financial intermediary *and* a user must sign a transaction in order to authenticate it, helping ensure that compromise of either one does not enable transaction forgery.

In such a digital-signature based CBDC, digital wallets perform transaction authentication by digitally signing the transactions users initiate. They must also play

[9]See, e.g., the entry on Digital Signatures, for a primer on digital signatures and public-key cryptography.

the critical function of storing users' private keys for them. Secure and reliable storage of cryptographic keys has proven a serious challenge in cryptocurrencies, holding important lessons for similar CBDC designs. This challenge is often referred to as the problem of *key management*.

Key management: One statistic neatly sums up the intractability of the key management problem in practice: An estimated 4,000,000 Bitcoin, worth tens of billions of dollars at the time of writing, have disappeared forever because of lost private keys [149]. Key management is so daunting to users that many store their cryptocurrency with exchanges (custody services), such as Coinbase, paradoxically recentralizing systems whose main selling point is their decentralization.[10]

In an ideal world, a digital wallet might take the form of an app on a mobile device that secures a users' private keys effectively and makes them available for signing whenever the user needs them. But what happens if a user loses or breaks her phone? Or she wants to initiate a transaction from a different device? Or her phone is compromised by malicious software?

There is a fundamental tension between *security*, i.e., preventing theft of private keys, and *availability*, i.e., ensuring that keys aren't lost. Perfect security for a private key is easy: Just delete all copies of the key. So is perfect availability: Post the private key on a blockchain. Obviously neither of these solutions is useful. The challenge in building a workable system is striking a good balance between security and availability.

The cryptocurrency ecosystem has evolved various mechanisms in the quest for a good key management solution, with mixed success:

- *Secure hardware:* Most mobile devices contain what are often called *secure elements*, designed to store keys so that they can only be accessed upon successful user authentication.[11] A range of special purpose *hardware wallets* for cryptocurrency, usually in the form of USB devices, serve a similar purpose. Hardware wallets in principle reduce the risk of funds loss due to remote compromise (e.g., hacking) of the device holding the wallet, but they do not by themselves mitigate the risk of funds loss through the unavailability (loss or destruction) of the wallet device itself. The use of trusted hardware for wallets is further discussed in section 8.4.

- *Mnemonic seeds:* When a user initializes a software or hardware cryptocurrency wallet, she is typically presented with a list of words, known as a *mnemonic seed*, that encodes her private keys for the wallet. Users are encouraged to write down their mnemonic seeds and store them safely, e.g., in a safe deposit box, to enable recovery of lost private keys. This mechanism helps the user guard against loss of funds through the loss or destruction of the primary device containing the

[10] Coinbase alone reportedly holds some 10% of all Bitcoin in circulation [150].

[11] The iPhone has shown that this approach can be quite effective against even powerful adversaries, as shown for example by the difficulty that U.S. law enforcement authorities have encountered in recovering encrypted iPhone data [151].

(hardware or software) wallet, at the cost of introducing a new risk – namely that of the wrong person obtaining the mnemonic seed.

- *Threshold signing / multisig:* It is possible to split a private key into a set of n *shares*, as discussed in section 3.2, so that any k out of n shares (for $k \leq n$) can be used to reconstruct the private key. Advanced cryptographic protocols, e.g., [152], [153] enable generation of a signature from shares without explicit reconstruction. With this kind of setup, it is possible for n different entities (people or organizations) to exert joint control over a digital asset, with any k having signing authority.

  A popular feature of cryptocurrency wallets today is *multisig* (multiple signatures) transactions. Multisig transactions are similar in spirit to threshold signing and have the same goal of joint control. They require use of k-out-of-n distinct signing keys to validate a transaction, though, instead of shares. Both threshold signing and multisig transactions are different embodiments of the threshold trust approach to decentralization discussed earlier in Section 3.2.

  Multisig transactions are technically simpler to implement, because they do not require the signing keys to be generated cooperatively, or even to use the same cryptographic algorithm. They exhibit a subtle privacy-versus-transparency trade-off, however: a threshold signature does not reveal *which* particular k out of the n share-holders signed the transaction, whereas with multisig transactions the set of k signers authorizing the transaction is clearly visible on the ledger. Threshold signing may be preferable if stronger privacy or *group anonymity* of the signers is desired. Multisig may be preferable if it is considered important to make each signer individually accountable for the transactions they sign, and to deter attacks in which a threshold of k signers might secretly collude to authorize a transaction improperly.

  Both threshold signing and multisig mechanisms achieve a balance between security and availability that can be parameterized by varying k and n and various other enhancements [154], [155]. Its limitations, though, including a need for multiple participating users or devices, make it less readily suitable for the wallets of individual users. Threshold or multisig wallets might in principle enable individuals to recover their funds with the cooperation of some threshold of trusted friends – analogous to the "trusted friends" account recovery path that Facebook already supports – but such social recovery mechanisms have not yet seen widespread support in digital wallets.

Flexible key management in CBDCs: Key management is especially challenging in fully decentralized cryptocurrencies such as Bitcoin and Ethereum because there is no authority to intervene to remedy failures such as key loss. Erroneous transactions are irreversible; countless users have suffered losses as a result [50]. In CBDCs, however—or indeed, any permissioned currency system—more flexible key management regimes are possible.

One option is to empower the operator of the CBDC or some other authority to *revoke* the public key (and thus corresponding private key) associated with an account, and *authorize new keys*. Such a capability is analogous to the ability to rectify errors in the ledger, as discussed in section 10.4. In a digital-signature-based system, it will be equally essential. Loss and theft of keys is inevitable. A practically workable system must include mechanisms to remedy these eventualities.

This general key-registration capability amounts to near-total control of CBDC funds. By changing keys, it is possible to transfer control of accounts arbitrarily. Consequently, a key registration system would itself represent a critical point of system compromise. One way to mitigate the risk is to vest this capability in a *set* of authorities, of which an authorized subset must cooperate to make key registration changes. Cryptographic tools such as multisig or threshold signing can serve as a technical foundation for such joint control, another application of threshold trust.

A second and perhaps complementary risk mitigation approach relies on notification and time delays before transactions are executed or become irrevocable. Bitcoin vaults [154], for example, impose a time delay on moving funds out of "cold storage", giving the owner an opportunity to notice an unauthorized transaction and cancel it with a recovery key. Paralysis proofs [155] enable recovery of funds if too many signers in a threshold or multisig account become unavailable.

A key question that a CBDC design must answer is whether – and for how long – a transaction should be potentially reversible in some way if it is discovered and adequately proven to have been improper. The cash-like, irrevocable finality of cryptocurrency transfers may be attractive for small payments but undesirable for larger, high-stakes transactions in which certainty is more important than speed.

The policies governing conditions under which authorities can alter account access can be largely independent of the technical mechanisms for key registration. These policies can be designed to comply with legal frameworks for asset recovery and transfer, as discussed in section 10, and can involve a blend of automated tools as well as interventions dictated by conventional legal and regulatory institutions.

5.3 User interfaces

Just as important as the capabilities of a digital wallet is *how it presents these capabilities* to CBDC users. User interface design profoundly impacts not only the acceptance of a system by users, but also system security.

The history of Internet browsers abounds with cautionary tales. Hackers and researchers have shown repeatedly how misguided graphical design choices have caused users to misinterpret browser content. Users have as a result been vulnerable to deceptive attacks that cause them to navigate to unintended, malicious sites or click on malicious content even in the face of warnings [156].

USB-type hardware wallets for cryptocurrencies also underscore these challenges. Many include features to prevent malicious software from subverting digital wallets by displaying a valid-seeming cryptocurrency address (account number) to a user but instead generate transactions that send money to attackers. These devices include

built-in displays that show the true destination address for a transaction. User studies, however, show that users derive only limited benefit from these cues [157].

Designers of CBDC wallets will have to contend with similar challenges. Happily, where wallets are managed by financial intermediaries (or a consumer-oriented CBDC platform), it is possible to leverage the array of sophisticated fraud-control mechanisms already prevalent in consumer payment platforms [158], and similarly to benefit from user interface design experience. These fraud-control mechanisms typically depend on some form of manual and/or automatic surveillance of transactions, however, exacerbating the privacy challenges discussed below in Section 6.

Some CBDC proposals, and CBDC-like systems such as Libra, envisage the possibility of sovereign control of wallets by users. Unless the CBDC takes on the task of fraud detection and remediation of transaction errors currently assumed by commercial banks, sovereign wallets will need to adhere to strict security requirements like those in cryptocurrency platforms, and fraud and error will be hard to detect or remediate.

6 Privacy and Transparency

In permissionless and decentralized cryptocurrencies like Bitcoin [17], regulatory oversight and compliance are generally an explicit non-goal: such systems are specifically designed *not* to be controlled by a state, a bank or any other central authority. In contrast, a CBDC likely needs to support mechanisms to enforce regulatory and compliance rules, as states want to detect and prevent criminal activities and ensure financial stability [10], [159].

At one extreme, we could imagine a CBDC in which transactions were made using real-world identities and were fully visible to an authority like the central bank or law enforcement in the clear. Oversight of compliance rules in such a CBDC, in terms of the detection and prosecution of violations, would be easy. Even if this oversight were done with good intentions, however, it would lay the foundations for large-scale abuse and human rights violations, enabling the government (and potentially private operators like banks) to track individuals with an unprecedented level of granularity.

At the other extreme, a CBDC that offers *full* privacy may not reveal any information about transactions to the operator(s)—a digital cash of sorts. This in turn would facilitate large-scale money laundering and make it near-impossible for law enforcement to track financial flows. Hence, we expect that most CBDCs will prefer to operate in a middle ground that offers some privacy protections to consumers, while also offering some visibility to auditors and law enforcement.

How to choose such a middle ground will reflect differences in individual or cultural values, likely related to the factors that make some individuals and some societies still prefer the relative anonymity of physical cash for everyday commerce, while others embrace the convenience of credit cards and willingly entrust their personal details and transaction histories to the card issuers. For this reason, it is critical for the identity-management approach a CBDC adopts to fit the cultural values and expectations of its intended user population, and these values may differ from one

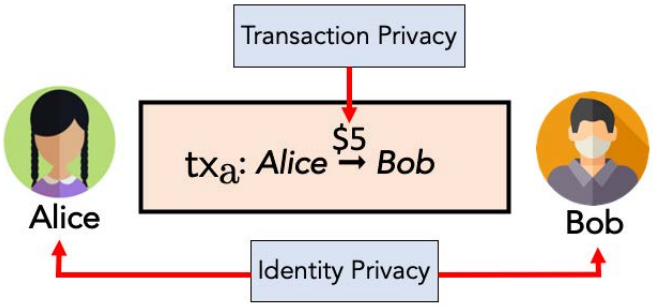

Figure 6: Two classes of privacy concerns arise: identity privacy, which concerns the participants in a transaction, and transaction privacy, which concerns the amount of the transaction (or the details of the contract in a smart contract).

country to another.

In addition to cultural considerations, choosing an appropriate middle ground will also depend on the technical limitations of existing tools. Indeed, there is a fundamental tension between transparency and privacy. On the one hand, transparency is essential to a digital currency because validators must be able to ascertain the correctness of transactions and their compliance with financial regulations. On the other hand, the more information we reveal to validators, the easier it is for those validators (and sometimes even outside observers) to learn information about individual transactions and the people executing them. This tension between privacy and transparency does not imply a "zero-sum" trade-off between the two, however: many privacy-enhancing technologies exist, outlined below, that can achieve privacy *together with* transparency in various forms.

We consider two types of privacy that a digital currency should consider (Figure 6). The first is identity privacy (Section 6.1), which describes the (in)ability to link transactions or activity to the sender or recipient of a given transaction. For example, we may wish to prevent an observer from learning that Alice sent money on Tuesday. The second is transaction privacy (Section 6.2), which describes the (in)ability to learn the nature of a given transaction. For example, we may wish to prevent an observer from learning that Alice paid $40 to the pharmacy. These categories of privacy require different tools, so we will separate them in the following discussion.

The trade-offs associated with identity and transaction privacy also have implications for decentralization of a CBDC (Section 6.3) and compliance (Section 6.4). Hence, we conclude the section by discussing some of these trade-offs.

6.1 Identity privacy

At the most basic level, a payment system can attempt to provide anonymity for its users by identifying them using *pseudonyms* rather than persistent identifiers. Such

a system might label transactions performed by Alice with a numerical user identifier (e.g., 1234) instead of her real-world identity ("Alice"). This offers a very basic form of anonymity, often called *pseudonymity*, but pseudonymity is fragile. If Alice, for example, reveals her name to a merchant in the course of a transaction, the merchant learns that 1234 = "Alice". If the merchant's customer database is breached, then Alice's pseudonym could leak more broadly. It improves anonymity if Alice uses different pseudonyms to identify herself to different merchants, or more generally if she goes by a different pseudonym with every user with whom she transacts. In Bitcoin, which is perhaps the most prominent example of a pseudonymous currency deployed today, a pseudonym, also called an *address*, is the hash of the public key for a digital signature scheme. This means that it does not inherently reveal any information about the user to whom it is tied, and that two pseudonyms cannot inherently be linked together.

In practice, however, pseudonyms can leak information about the participants in a transaction. Indeed, there exist blockchain analytics companies whose entire business model is centered around de-anonymizing users (e.g., Chainalysis [160]). In Bitcoin, it is possible to *cluster* pseudonyms together according to common transaction patterns [161]–[164]. For example, an exchange may combine the addresses associated with its different users in ways that make it possible to identify that they are all part of the same exchange. An entity storing the ledger of transactions can even perform this clustering retrospectively as they discover new patterns. If they can further *tag* addresses by performing their own transactions within the network (e.g., depositing coins into and withdrawing coins from an account they create at an exchange), they can identify the real-world owners of not only the addresses they directly interact with but also the broader clusters that their tagged addresses are part of [163]. Even if we consider a completely passive attacker who does not perform their own transactions, lists of tagged Bitcoin addresses are readily available online (e.g., at `https://www.walletexplorer.com/`). These attacks are best suited to de-anonymizing services like exchanges rather than individuals, but within most cryptocurrencies these exchanges represent a large part of the ecosystem, and a chokepoint in terms of how users can get their money into and out of it.

This simple technique of using pseudonyms has been adopted by most other cryptocurrencies, meaning the de-anonymization attack described above works equally well for them. So-called "privacy coins" such as Zcash and Monero, however, have different ways of forming transactions that use more advanced cryptographic techniques to protect anonymity. Even within these cryptocurrencies though, it is still possible to identify patterns of usage and de-anonymize users and services accordingly [165]–[167]. In Zcash, for example, the main privacy feature is optional, and the majority of transactions do not take advantage of it.

The previous attacks rely solely on analyzing transaction patterns within the ledger itself. However, pseudonymous identifiers can also be used to link accounts to users by exploiting communication network infrastructure; i.e., the Internet. When a user initiates a transaction involving digital currency, they must send a message over the Internet to communicate this transaction to the CBDC validators. For example, the user might send the transaction to their bank, which may act as a validator and

also forward the transaction to the remaining validators. A sophisticated attacker with access to Internet infrastructure (e.g., a nation-state or even a large Internet service provider) may be able to observe this message and identify its point of origination, i.e., IP address, even if the user encrypts their message. IP addresses can sometimes be linked to people, or at least to a person's hardware. If this same entity is also able to view the ledger, they may be able to link the IP address to an account. This again represents an attack on the anonymity of users, since an entity other than the user's bank can link financial accounts to a user's IP address.

Solutions: Prior experience suggests two main approaches for managing this class of attacks. The first approach is to eliminate pseudonymous identifiers. Cryptographic techniques can ensure that any two transactions by the same user cannot be linked together via the transaction contents, but can still be validated [168]. Although these techniques increase the complexity of implementation, they make it much harder for an adversary to link a user to their transactions.

Even without pseudonyms, it may still be able to link *individual* transactions to an IP address, using the previously described techniques. A common approach for addressing this concern is to alter the network relay dynamics. For example, instead of sending the transaction directly to a validator, the user may route her transaction through proxies using third-party services such as Tor [169]. However, this solution is not scalable to millions of people, nor is it usable by the average user due to high technical complexity. Additionally, it requires users to trust a third-party service, which may itself be vulnerable to attacks. Indeed, state-level adversaries have been known to break the anonymity of proxy services like Tor successfully [170], [171].

In general, there are no cryptocurrencies or privacy features today that make it impossible for attackers to learn information about the identity of senders and recipients in the network. This suggests that making the ledger of a CBDC globally visible would be undesirable, and that similarly it is not clear how to prevent validators (i.e., the organizations maintaining the ledger) from learning information. This further suggests that achieving similar anonymity guarantees to physical cash, at the same time as achieving forms of regulatory oversight (as we discuss in Section 6.4), is an open and challenging problem.

Active probing attacker: An average client does not typically meet the storage requirements needed to store something like the ledger of a cryptocurrency: e.g., the Bitcoin blockchain is currently on the order of hundreds of gigabytes. In a CBDC, clients may be unlikely to have direct access to the ledger anyway if it is centralized or permissioned, as discussed earlier in Section 3.3. As discussed above, this means that clients are likely to submit their transactions to the ledger through a validator, who will then be able to identify that client as the sender in the transaction.

Similarly, clients who receive a transaction may want to check with a validator if it is in the ledger or not. In Bitcoin, this is equivalent to a *lightweight* client, who stores only the headers of blocks, performing this check with a *full node*. Rather than completely trusting the full node to give the right answer, the client asks them for a

proof that the transaction is included in a block, which the client can verify against the block header. Clients may be more willing to trust validators in a CBDC (although they should not have to), but the privacy issue remains that asking about the transaction directly is a clear signal that the client was a recipient in that transaction: there is no reason why they would know about or be interested in it otherwise.

Furthermore, some version of this attack could be carried out by a remote attacker who just observes the network connection between the client and the validator [172]. If the data is not encrypted in transit, for example, then they have the exact same information as the validator. Even if the data is encrypted, a remote attacker can learn information – for example, whether or not the client was interested in a transaction – based on the timing of its network traffic. It is thus important to protect these network connections by encrypting all traffic, and for all participants to use constant-time algorithms to avoid revealing information via timing channels.

In addition to the network-level anonymization techniques discussed above, it may be possible to run a type of *private set intersection* or *oblivious transfer* protocol instead, so that the client learns only whether or not the transaction is included and the validator learns nothing. This is an open research area, however, and furthermore one that typically involves the usage of advanced cryptographic techniques. Some approaches have instead relied on trusted hardware, as discussed in Section 8.

A more concrete approach is to design the ledger so that a client does not require active privacy-sensitive communication with anyone in order to check if a transaction is included in the ledger. In the SkipChain structure [173], [174], for example, all committed blocks are collectively signed by a rotating committee of signing trustees, whose evolution is similarly validated by collectively-signed "forward links." This structure enables any client to provide any other client with a direct cryptographic proof of a transaction's commitment to the ledger, without needing to make privacy-sensitive queries to full nodes or other third parties. While there are complexity costs and security trade-offs to be considered in this approach, the Libra blockchain has adopted it [175] and some variant may be appropriate for CBDCs more broadly.

6.2 Transaction privacy

Beyond identity, there are other aspects of blockchain computations that users might wish to keep private. First, they may wish to mask the *input data*. In a digital currency, this might correspond to transaction amounts. In a smart contract platform, this could include other metadata, such as private customer data. Second, users may wish to mask the *nature* of the computation being executed by a smart contract. As one example, a group of participants may want to compute a private algorithm for predicting the viability of a loan, without revealing the structure of that algorithm to other CBDC participants. These two challenges of *data privacy* and *program privacy*, respectively, are generally addressed by different technical approaches in practice.

Data privacy: The simplest form of data privacy is hiding the amount of a transaction. This typically involves the use of more advanced cryptographic techniques, such as encryption or commitments. Here, however, it is crucial to keep in mind the

balance between privacy and verifiability: the amount that a user is sending can be hidden using encryption, but it is equally important that users cannot send more money than they actually possess. Validators must thus be able to check this property, at an absolute minimum, even without knowing the amount themselves. This is typically achieved using advanced cryptographic primitives known as *zero-knowledge proofs*. Such proofs allow the sender to prove that the amount they are sending does not exceed their current balance without revealing the amount or their balance.

To reap the benefits of zero-knowledge proofs, participants must be able to generate and validate transactions containing encrypted data. This design choice has trade-offs in terms of transaction efficiency. Zero-knowledge proofs typically take longer to generate and verify than a regular unencrypted transaction (on the order of seconds). While such a delay is negligible in blockchains with long confirmation times (e.g., Zcash), it could be a concern in a CBDC. A more significant concern is that validators likely want more than just this basic level of verifiability, in order to ensure compliance with financial regulations. This is a challenging problem that we discuss further in Section 6.4.

Another limitation of zero-knowledge proofs is that they can only prove facts about confidential data held *off* the ledger, but cannot ensure that the data is actually retained or disclosed when policy requires, as determined by a regulatory process or a smart contract for example. If policy allows transaction amounts and participant identities to be *normally* hidden but requires them to be disclosed if a relevant account comes under investigation, for example, then zero-knowledge proofs are inadequate by themselves because they can prove that the transaction amounts and identities are valid and *exist* but cannot ensure their retention and disclosure at investigation time. Some experimental ledger designs allow confidential data to be stored *on-chain* and collectively entrusted to the blockchain validators, which cooperate to control and record accesses, and to decrypt or transfer the confidential data when authorized [176], [177]. This design ensures that the ledger itself can retain and enforce selective disclosure or transfer of confidential transaction information on demand, at the cost of requiring users to place slightly higher privacy trust in the collective set of ledger validators.

Program privacy: Generalizing these ideas to smart contracts is challenging for two reasons. First, the inputs to a smart contract can be much more complex than inputs to a regular transaction. Second, the operations executed in smart contracts are also more complex. These issues make it difficult to extend zero-knowledge proof techniques for verifying encrypted transactions to arbitrary smart contracts. A common alternative is using *secure multiparty computation* (SMC). SMC is a class of techniques allowing multiple parties to compute a function of encrypted data without learning the function inputs. Although SMC is generally computationally inefficient, recent efforts have modified industrial blockchains like Hyperledger Fabric to support a curated class of SMC smart contracts optimized for performance, e.g., running private auctions [178].

Another possibility is for a subset of participants to perform the computation

locally and report only a commitment to its new state to the ledger. Again, this raises the question of verifiability: how can validators be sure these *compute nodes* are not lying? One approach is to provide strong disincentives for dishonest behavior, by allowing other participants to challenge the results of a computation and have the network punish compute nodes who are caught lying [179]. Another is to have compute nodes run the computation inside of a trusted execution environment (TEE), or *enclave*, which guarantees that their reported results are correct and that all data inside the enclave is kept private [180]. Finally, compute nodes can provide a succinct zero-knowledge proof that they executed the computation correctly [181]. All of these approaches have trade-offs in terms of their functionality, efficiency, and required levels of trust (for example, in the compute nodes or the TEE manufacturer) that must be taken into account when considering their use in a deployed application like a CBDC.

6.3 Privacy and decentralization

Considerations of centralization versus decentralization, introduced earlier in Section 3 from a ledger infrastructure security and trust perspective, create similarly complex privacy considerations and tradeoffs that we briefly outline here.

One of the central questions is what the CBDC's threat model should be for privacy purposes: i.e., *from whom* should the CBDC protect sensitive identity and/or transaction data? If it is acceptable from a social, legal, and risk perspective to require users to place complete trust in the central bank to protect their privacy, for example, then a fully centralized design may be both simplest and most effective at protecting clients *from each other*. In such a design, however, a single bulk data breach of just one replica of the permissioned ledger or an associated sensitive database can expose the identities and financial histories of millions of users at once, as the Equifax breach affecting nearly half the US adult population amply demonstrated [182].

Many privacy and cryptocurrency advocates, therefore, favor more decentralized approaches to privacy protection, which a CBDC design should consider. The three main forms of decentralization introduced earlier in Section 3.2 – role separation, trust dispersal, and threshold trust – are all potentially useful to varying degrees in protecting user privacy against the compromise of any one server or authority.

Role separation: The Bank of England's proposed delegation of account and identity management to a Payment Interface Provider (PIP) [10], for example, makes it simple to give users a limited form of *accountable anonymity* [183] for identity privacy. If the central bank normally treats accounts as pseudonymous public keys, and only the PIPs verify and record the associated identity information, then individuals are at least pseudonymous with respect to the central bank and the ledger transactions it processes – provided, of course, the PIP adequately protects this identity information. The PIP can disclose the identity associated with an account under suspicion to the central bank or an independent investigator as appropriate, however, to address regulatory compliance and anti-money laundering considerations. In cultures

where banking customers are more inclined to trust private companies than governments with their personal information, this form of role separation for privacy may be reasonable and useful, however limited.

Trust dispersal: The delegation of identity management to *multiple* commercial PIPs can similarly benefit privacy in terms of trust dispersal: the breach of a single PIP in principle affects only that PIP's customers, and not the CBDC's entire user population. While trust dispersal reduces the aggregate amount of trust placed in any one PIP, it does not necessarily do anything to improve the situation of – or perhaps to placate – each of those unlucky customers whose PIP has suffered a data breach. Further, the real possibility that the PIP service market could become dominated by a few large players – as in the case of credit rating services like Equifax in the US [182] – limits the privacy protection we can expect from decentralization via role separation and/or trust dispersal alone. Experience indicates that almost any centralized database of sensitive personal information inevitably becomes a prime target for hackers, identity thieves, industrial spies, or foreign adversaries. Therefore, while the two-level separation between central bank and multiple PIPs is a promising starting point already being embraced by multiple CBDC projects, including the digital yuan (Section 11.2) and the e-krona [5], we recommend that it be viewed merely as a starting point and not as a complete decentralized privacy solution.

Threshold trust: Analogous to the way that Byzantine state machine replication (SMR) protects the integrity of a ledger against any one compromised server (§3.3), *threshold cryptography* techniques such as Shamir secret sharing [184] protects the *privacy* of confidential information from any one compromised server holding a *share* of it. A threshold number of the parties holding shares must work together to decrypt or do anything with the threshold-encrypted data. With 3-of-5 secret sharing, for example, there are five independent trustees each holding one share, at least three of which must work together to decrypt the data. Threshold signing, discussed earlier in Section 5.2, is just one of numerous applications of threshold cryptography. Other applications include secure decentralized data deletion [185], or decentralized management of *on-chain secrets* entrusted to a ledger [176], [177]. Proper use of threshold cryptography could in principle address the tension between user identity privacy and the investigatory needs of law enforcement, as discussed below in Section 6.4.

To counter a common misunderstanding, a complete data breach in one of the five trustees in this example does not cause one-fifth of the data to be leaked. Instead, correctly-implemented threshold cryptography ensures that complete data breaches in one or even two of the trustees results in *no* data leakage. Only under a "full-threshold" breach, of three trustees simultaneously in this example, is any data leaked at all. At that point, *all* data may be leaked in bulk. Thus, in such systems it is critical that thresholds be chosen carefully to minimize the possibility of a full-threshold breach, while balancing this risk against data availability risks due to too many trustees failing or being affected simultaneously by network outages for example.

It is worth pointing out the cautionary note that a large number of projects and

technologies claim to "use blockchain" to "secure" sensitive private data, but do not actually implement decentralized privacy protection. In particular, most blockchain-based systems designed to manage personal information actually entrust the *privacy* of the data either to a mobile device, which may be lost or stolen; to a cloud service, which is a central authority that may suffer a data breach; or to a hardware security module (HSM), which may have bugs or side-channel leaks. The "blockchain" in such designs typically only *records* uses of the private data that the centralized data trustee device *claims* to have performed. If the trustee device (mobile device, cloud service, or HSM) is compromised, however, it can readily leak the private data to an adversary without recording this fact on the blockchain. Referring back to the C-I-A triad (§3.1), even a blockchain that perfectly protects its integrity and availability cannot protect the *confidentiality* of a user's data if that data is held off-chain by a centralized trustee that might leak it without even recording that fact on the ledger.

6.4 Privacy and compliance

Today, most countries have compliance rules whose goal is to protect the economy against malicious activities like money laundering or tax evasion. (Specifics, such as the United States requirement that cash transactions exceeding 10,000 USD must be reported to the government, are discussed in Section 10.2.) If a central bank were to deploy a CBDC, most likely it would want to have mechanisms in place that allow it to detect or prevent large transactions that exceed such limits (or series of small transactions that exceed the limit combined). That is, the CBDC implementation should the support the enforcement of *pre-existing* compliance rules. We discuss anti-money-laundering (AML) laws in more detail in Section 10.2.

Additionally, the introduction of a CBDC might create the need to introduce and enforce *new* compliance rules. One particular threat is the implications of a CBDC for financial stability. During a financial crisis people might get worried about a bank run and want to move bank deposits into CBDC (which would be free of such risks by definition). This could, in turn, make the risk of bank run seem even more probable and create a vicious cycle. The potentially resulting massive shifting of funds might threaten the stability of the entire financial system. This threat is brought up in the CBDC discussion paper from the Bank of England [10] and also mentioned in the working paper of the European Central Bank (ECB) [159]. As a potential solution, both documents suggest limits of how much CBDC any individual can hold at a given time. Such a holding limit is an example of a new compliance rule that might be needed, if a CBDC were to be introduced.

Challenges: Recent research efforts have explored the question of how to combine anonymous payments with compliance and oversight. In the context of token-based digital currencies, Camenisch et al. designed a solution that allows users to make payments such that they remain anonymous but the bank who issued the coins can enforce simple compliance policies such as per-user payment limits [186]. The main problem with such solutions is that they do not provide recipient anonymity or hide payment amounts. In the context of ledger-based digital currencies, Garman et al. studied

how similar compliance policies could be realized in payment systems like Zcash that provide strong privacy protection [187]. The drawback of such schemes is that they require expensive zero-knowledge proofs (SNARKs) and a trusted setup phase. Wüst et al. showed how simple payment limits can be combined with more lightweight anonymous payments that leverage cryptographic commitments [188]. One problem of such solutions is that transaction linking is still possible. Additionally, all above mentioned schemes require an enrollment phase where the identity of the user is verified by a trusted authority which may further complicate the adoption of a CBDC.

Recent work on privacy-preserving surveillance [189]–[194] may suggest CBDC designs that could provide strong privacy protection for transaction amounts and identities while ensuring not just proactive compliance (e.g., conformance with account balance or payment limits) but also retroactive compliance like retention and disclosure of confidential information in an investigation. An example privacy-preserving investigation process might require a "warrant" targeting a transaction or account of interest, even if the target's identity is as-yet unknown [190], [192]. Such a warrant, authorized by an independent judge and tallied in aggregate accountability mechanisms [193], [194], might authorize the transfer of encrypted on-chain secrets [176], [177] containing transaction amounts, identities, and other information to an investigator.

In general, the question of how to achieve payment privacy and enforcement of compliance rules with good performance, acceptable trust assumptions, and simple adoption is challenging. While the technological building blocks for such designs already exist, building them into operational, secure and privacy-preserving systems thus far remains an open challenge.

7 Smart Contracts

Many state-of-the-art digital assets feature a smart contract programming language. This is a way that independent third-party independent developers can extend the digital asset with new functionality. It is not necessary for a CBDC to provide smart contracts in order to fulfill its primary role as a digital currency, and some CBDCs (including the digital yuan, for example) are unlikely to do so. However, smart contracts can be an important way that a CBDC fosters innovation from other entities such as commercial banks and fintech providers.

There is a broad design space of smart contract languages for a CBDC to consider, and potentially many pitfalls. Notably, there have been many expensive losses in cryptocurrencies like Ethereum due to smart contract coding errors (known as "bugs") that have led to either to accidental losses or else made them vulnerable to deliberate attacks from opportunistic hackers. If a CBDC incorporates smart contract functionalities, then it will be important to consider safety and security when designing these.

Background: Smart contracts arise from the need to extend the spending limits and policies provided by digital assets, some of which are described in Sec-

tion 5. For example, to help secure high-value accounts, it is considered a best practice to attach restrictions such as `"Funds from account A may only spent with authorization from any two out of Alice, Bob, and Carol"` or `"Only $1000 can be spent per day"`, as well as many other possible variations or combinations. Smart contracts provide a flexible way for users to define and customize such policies.

More generally, smart contracts can behave like trusted third parties, realized using software. Smart contracts can include conditional statements, for example, in very high level pseudocode,

```
Alice may choose before time T whether to receive either $10 or
10 TOK from Bob
```

or

```
"If the price of TOK tokens exceeds P, then transfer $X from Alice
to Bob".
```

These policies are codified into machine-readable programs that can be executed by the system operators. Alice's choice in the above example, would be expressed through a digital signature, which is then checked by system operator according to the rules of the program.

In many applications, including the above examples, the correspondence with traditional legal contracts is fairly clear: The program code is available on-chain for both parties to see, hence it reflects the mutual agreement between them. The parties to the contract use digital signatures to express their intent to be bound by the contract and their acceptance of its terms. The contract may explicitly define the consideration to (automatically) transferred upon fulfilling the contract. Other applications of smart contracts do not as easily fit this pattern. For example, some of the most popular applications have included auctions, lottery games, and exchange services. Once funds are deposited into an account associated with a smart contract program, such funds are subject to those programmatic rules.

Smart contracts have emerged as an important tool for innovation in today's digital assets. Many of the most widely used applications, such as auctions exchanges, as well as safety features like multi-signature wallets, have been written by independent developer teams separate from the core developers responsible for the platform. Since the developers of smart contracts are not explicitly trusted by the platform, the design of the smart contract language is essential in defining the boundary for untrusted developers. As one example, regardless of how a user customizes their smart contract, they should not be able to inflate or counterfeit the underlying digital currency. This is achieved in Ethereum by defining the programming language, known as the Ethereum Virtual Machine (EVM) so that upon encountering any instruction that makes an account balance go negative the entire transaction is reverted. Thus smart contracts can be thought of as defining the ground rules for a sandbox, within which developers are allowed to innovate.

7.1 Striking a balance between safety and extensibility

The need for program verification: Even in Ethereum's first few years of operation, we have seen numerous expensive disasters caused in part by smart contract "bugs", errors introduced when developing the smart contract programs. These have underscored the importance of program verification and other verification tools that can help identify such errors or correct for them. A few are especially instructive for illustrating some potential solutions: the 2016 theft of $50M USD worth of Ether from "The DAO" smart contract, the July 2017 theft of $30M USD worth of Ether from the Parity Wallet smart contract,[12] and the November 2017 accidental loss of $30M USD worth of Ether from the same Parity Wallet smart contract. In all three of these incidents, the underlying programming errors were subtle and were not found during security audits and code reviews. In the first two incidents, the vulnerabilities were deliberately exploited by hackers who aimed to profit from the theft, while in the third case the loss was triggered by accident — the funds were not stolen, they were simply made inaccessible. A CBDC should take efforts to prevent such disasters with forethought.

Support for verification and analysis: The expensive disasters mentioned above, as well as many others, have led to an active research effort in designing and applying program analysis and verification tools to smart contracts [195]. Program analysis tools aim to identify known classes of bugs, and rule out certain kinds of misbehavior. Program verification aims to provide a guarantee that the software satisfies certain requirements (e.g., that an account balance cannot go negative), or implements a formal specification. These tools can be used to help developers avoid bugs in the first place, and can also be employed by users performing due diligence to inform their decisions about whether to use the contract. Most smart contract language compilers and editors, such as the Remix IDE in Ethereum, contain some degree of ad-hoc bug and hazard detection — for example, warning users if a contract has code that resembles bugs that have occurred in the past. Complementary tools like Mythril, Oyente and others can also be used to evaluate smart contracts. Besides smart-contract specific tools, there are also generic frameworks for program analysis that can be adapted to this use. In particular, the K-Framework has been used to analyze a large number of Ethereum smart contracts [196].

Some program analysis tools can be directly built into the smart contract language, effectively preventing classes of bugs from appearing in the first place. For example, the information flow language[13] allows the programmer to annotate the smart contract with trust relationships, i.e. "Alice should not be able to affect Bob's outcome in this contract." The information flow type checker can catch errors where the smart contract implementation fails to enforce this constraint. As another example, some important invariants such as that digital assets should not be counterfeited, can be enforced through so-called "linear types" [197], [198] in essence, quantities of currency are annotated as "linear", which means they must behave like conserved quantities that

[12] https://hackingdistributed.com/2017/07/22/deep-dive-parity-bug
[13] https://github.com/Neroysq/VyperFlow

are exchanged, but not created or destroyed. Some languages, such as Michelson and Plutus, are based on functional (rather than stack- or register-based) programming models, which may make it easier to adapt new program analysis tools.

A CBDC could help avoid smart contract programming errors by developing program analysis tools alongside the CBDC platform itself. To the extent a CBDC must choose which existing smart contract technologies to co-opt from, it should take into consideration the maturity and effectiveness of available program analysis tools. A CBDC may choose to require mandatory requirements for some kinds of program analysis, which could be enforced automatically by the platform.

Expressiveness, restrictiveness, and domain-specific languages: None of the promising applications for a CBDC strictly require a smart contract programming language at all. Instead, capabilities anticipated by the CBDC platform designers can hardwired in. This would prevent bugs introduced by new smart contracts developed by third-party developers, but would also foreclose on their potential innovations.

Many smart contract systems in use today offer some compromise in between a fully general purpose smart contract language and hardwired applications. Some smart contract systems like Bitcoin script are based on a general purpose programming language but with many control-flow structures (like `for`-loops) removed. This can make programs easier to analyze, but also rules out many applications. Another approach, called "domain specific" languages (DSLs), which includes DAML, BitML, offer some flexibility to generalize, but are designed primarily around particular classes of applications such as auctions or bilateral agreements.

Support for upgrades, reversibility, and redactions: Even a language selected with security in mind will inevitably encounter bugs and mistakes. These bugs may arise through user error, creative development, or malicious intent. A CBDC may anticipate needing to employ a variety of mitigation and remediation strategies, which may include overwriting account balances, changing the code of a smart contract, or potentially other modifications. It is true that blockchains are "immutable" in the sense that the historical record of past transactions, since they are copied widely between the system operators as well as the users, are likely to remain available and may be difficult to suppress. However, the rules of a blockchain can be changed by the system operators, and in fact this has been an important way that many mistakes and disasters have been remediated in existing digital assets. As an example, the earlier-mentioned theft of $60 million USD worth of Ethereum cryptocurrency affecting The DAO in 2016 was addressed in such a way. A large segment of the Ethereum community agreed to a "hard fork," which effectively transferred coins from the thief's account back to the original participants of The DAO. Although the remediation in this case was successful, this has not always been the case. It took over a month to coordinate on the hard fork change, but in most cases an attacker would have been able to escape with the funds in a matter of hours, making such a hard fork ineffective. There have been several proposals from the research community, known as "reparable" or "redactable" blockchains [199], [200], that aim to simplify

the process of applying such remediations so they can be applied more swiftly.

Even without relying on intervention from the platform itself, smart contract developers can build "upgradeability" features into their applications themselves. For example, OpenZeppelin, a popular library of smart contract templates, provides such a feature as a proxy layer. Essentially, the code for the outer layer, which indeed must be immutable, delegates its authority to a dynamically-named contract that can be updated by the original creator of the contract.[14] A CBDC may minimize the need to take platform-level remediations by encouraging smart contract administrators, such as fintech companies or commercial banks, to make use of such mechanisms.

Handling contention and concurrency in transactions: An important way in which smart contract languages have differed is in how they handle concurrent transactions and shared resources under contention. A benign example of a concurrent transactions is the following scenario, which is reminiscent of check floating in traditional (non-blockchain) banking.

1. Alice's account balance is initially \$10. She initiates a transaction T that sends \$20 to Bob - more than she has in her account.

2. Alice receives \$10 into her account because of a transfer on-chain, such as a withdrawal from an exchange. Alice's account now holds a \$20 credit.

3. Transaction T is committed on the blockchain, transferring her entire \$20 account balance to Bob.

Such "floating" transactions are supported in account-based cryptocurrencies such as Ethereum, just as they do with checks in the traditional banking system. It works because the identifier for Alice's account does not depend on its history of transfers. In contrast, with digital assets based on "Unspent Transaction Outputs" (UTXOs), such as Bitcoin, a transaction must refer not to the account identifier, but to the prior transfers that provide the source of funds. Hence in a UTXO-based digital asset, Alice could not initiate her transaction (Step 1) until after her withdrawal transaction is authorized (Step 2).

For other applications, such as auctions, this restriction on concurrent transactions makes the UTXO model far less flexible. Consider an auction, in which any member of the public can place a bid, and each bid is assigned a sequence number. The following example in pseudocode illustrates how such a mechanism could be implemented in an Ethereum-like digital asset:

```
memory cell BidCounter := 0;
    function PlaceBid() {
        ...
    BidCounter := BidCounter + 1;
...
}
```

[14] https://docs.openzeppelin.com/upgrades/2.8/writing-upgradeable

Suppose Alice and Bob each place a bid at roughly the same time — one or the other would be processed first, depending on network timing or potentially left to the choice of the system operator. However, this would be very difficult to implement in the UTXO model. One approach would be to represent the current value of `BidCounter` as a UTXO; a bid would need to "spend" the current value, and "create" a new UTXO for the updated value. If Alice and Bob place bids at roughly the same time, then only one of them would be committed, and the other would have to be resubmitted.

Besides account based and UTXO-based transaction models, other choices are possible as well. The Execute-Order-Validate model from Hyperledger Fabric lies somewhere in between. While account identifiers are used when initiating a transaction, the transactions are checked for "overlapping read/write sets" (such as the `BidCounter` in the above example), hence concurrently submitted bids may or may not need to be resubmitted, depending on the implementation of the system operator.

To summarize, the UTXO model can be seen as a restrictive case of the account model, which can avoid some potential hazards or delays (effectively preventing "check float"), but that also limit the ability to implement mechanisms like auctions through the smart contract system.

The potential of smart contracts to accelerate systemic risks: Even besides coding flaws, smart contracts may function exactly as intended by their creators, and yet when interacting together may lead to systemic hazards. Some set of smart contracts, such as stablecoins and other decentralized finance (DeFi) instruments, are highly interdependent, and rely on each other as collateral. It can be difficult to identify when the conditions are ripe for cascading "bank runs" on these instruments.

A fascinating recent example of systemic risks brought about by smart contracts has been the use of "flash loans" in price manipulation attacks. Flash loans are a particular kind of smart-contact enabled loan, for which there is no clear analog in traditional finance. In a flash loan, a user offers up their digital assets for a limited range of "zero counterparty risk" uses. Essentially, the rules of the smart contract guarantee that the funds that must be borrowed and repaid all within the timespan of a single transaction. This condition is automatically enforced by the smart contract code, which checks the account balance at the beginning and end of the transaction, and invalidates the entire transaction if these do not match. An example of a legitimate use of such a loan is an arbitrage opportunity across on-chain exchanges. If the currency can be traded in a cycle across multiple exchanges, all within a single atomically-executed (all-or-nothing) transaction, then a flash loan can enable both the arbitrageur as well as the lender to earn a profit. However, flash loans can also be used to manipulate market mechanisms whose "price estimate" can be temporarily affected by the movement of the flash-lent funds.

Limitations of enforcing policies through restrictions at the CBDC platform level: It may be tempting to think that a CBDC can limit the unsafe use of digital assets simply by restricting the expressiveness of the smart contract language. However, we have seen that even simple smart contract languages can be extended

to new features by sufficiently clever developers, even in ways that are not intentionally supported. For example, even basic digital assets with only a plain digital signature can be enhanced with "threshold signing" functionality (as described in Section 5) without any explicit support from the platform. A CBDC may collaborate with regulators to restrict such uses, but enforcing them may be outside of what can be automatically enforced within the system.

At an extreme, desirable smart contract features that are not provided by the CBDC platform itself, may be achieved by relying on third party custodians. This is already the case, for example, with stablecoins such as Coinbase's USDC. This is backed by deposits of dollars custodied by Coinbase itself as a third party, hence relies in trust in them as an administrator. It is possible that limiting the extensibility of the CBDC in an attempt to improve safety may have the unintended effect of driving more users to services outside the sphere of influence of the central bank. This may be an argument in favor of the CDBC providing smart contract features.

7.2 Off-chain protocols and advanced cryptography

While the focus of most program analysis and safety features for smart-contract based applications have focused on the smart contracts themselves, smart contracts are only one of several software components making up the entire application. Other software, such as the (often web-based) user interfaces can potentially contain bugs and lead to dangers as well. It will be important for the CBDC to consider safety features and analysis standards for these software components as well.

As one example, in order to support rapid micropayments, faster than what can be provided directly by the platform, smart contracts can enable "off-chain" payment channels, which involves a smart contract acting in concert with digitally signed messages exchanged between parties as well. These off-chain messages and digital signatures should be scrutinized to the same degree as the smart contract. In general, to improve performance and reduce execution (i.e., gas) costs, smart contract developers have found ways to make use of complex arrangements involving cryptographic evidence and minimally-trusted third parties. These includes "roll-ups", or verifiable computing, which can accelerate the task of validating a blockchain.

7.3 Smart contracts as a two-layer architecture

It may be most effective for a CBDC to function as a two-layered system, where the central bank issues digital tokens to commercial banks that in turn maintain the digital wallets and define the smart contract languages. The central bank could focus on defining a minimal set of features, just what is necessary to support the flow of funds between the layers, while the innovation and customization is developed by the commercial banks at the second layer.

One approach is to consider each of the commercial banks, as well as the central bank, to maintain its own instance of a blockchain. The challenge then is to provide a way to manage the transfer of assets from one chain to another. There has been significant research and development work on defining protocols for interoperability

across independent blockchains, such as HyperService, Aspen, Interledger, Cosmos, Protean, and sidechains [177], [201]–[203]. The basic idea of interoperability is that digital assets defined on Blockchain A can serve as a backing store for "shadow assets" defined on Blockchain B. The backing assets retain the security and monetary policy properties of Blockchain A, while the shadow assets can be traded and used in smart contracts according to the functionality defined by Blockchain B. This two-tier structure could function quite similar to how central banks currently provide reserve accounts that commercial banks must use as backing assets. The main challenge in interoperability is to ensure that the operators of the backing blockchain do not need to be aware of all the details of all the other blockchains, as otherwise this would undermine the extensibility and scalability benefits of point of independent operation. The minimal requirements to support such operation are that the backing Blockchain A can recognize a limited number of transactions (e.g., withdraw and deposit) on the shadow Blockchain B. In the simplest case, Blockchain B is identified simply by the public key of a fixed service provider responsible for it. The operators of Blockchain A can identify valid transactions on Blockchain B from these signatures. Other cases are more challenging. A research focus has been to define interoperability between public blockchains based on proof-of-work [53] and proof-of-stake [62], [204], which pose several challenges due to the design of their consensus protocols. If the CBDC is based on a permissioned blockchain architecture, it may be much simpler to define interoperability using simple digital signatures.

8 Secure Hardware

The role of secure hardware in digital currencies is a controversial and often misunderstood topic. The goal of this section is to provide non-expert readers a brief introduction to secure hardware technology and discuss both the main benefits and the limitations of the currently available secure hardware variants. We explain the problems of simple digital currency schemes that rely on secure hardware and discuss better ways to leverage secure hardware. Finally, we provide recommendations regarding the use of secure hardware for organizations like central banks that are currently investigating CBDC deployments.

8.1 Brief introduction to secure hardware

The term "secure hardware" commonly refers to a computing environment whose goal is to protect data and computation. Secure hardware can be used to execute security-critical applications such that their data and execution is protected and isolated from the rest of the computing platform that can be untrusted.

The use of secure hardware can enable significant security improvements compared application execution on standard computing platforms such as PCs and smartphones. In standard computing platforms, the security of any executed application relies on the trustworthiness of a very large software stack that includes the entire operating system (OS) and thus millions of lines of code, and also the trustworthiness of the

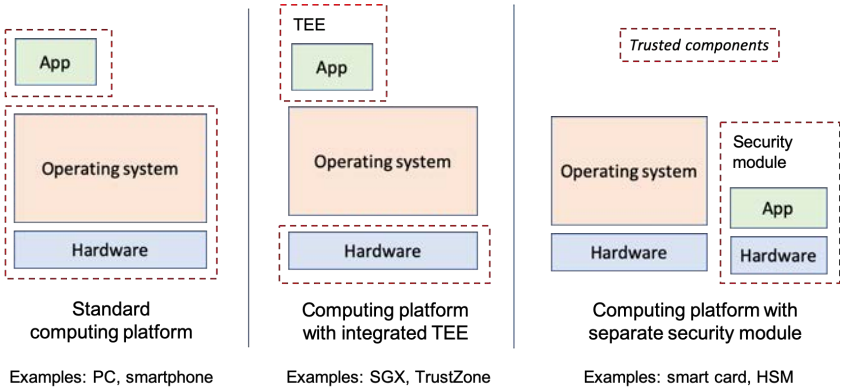

Figure 7: In standard computing platforms, shown on the left, the entire OS must be trusted for secure execution of security-critical applications. In computing platforms with TEE capabilities, shown in the middle, it is sufficient to trust the application's code and the underlying hardware. In systems with dedicated security modules are used, shown on the right, the additional hardware and the application code must be trusted. The trusted components in all three models are shown surrounded by a dashed red border.

entity who operates the computing platform like an administrator. This standard trust model is illustrated in Figure 7, on the left. In comparison, when the same application is executed inside secure hardware, its security relies only on the correctness of the application code itself (e.g., few hundred lines), and the assumption that the protections of the secure hardware technology itself cannot be circumvented.

Secure hardware variants: Secure hardware can be realized in two main ways. The first common approach is a separate module or a chip that is connected to the rest of the computing platform over an interconnect. This approach is shown in Figure 7, on the right. The second common option, and one that has gained popularity recently, is to enhance the general-purpose CPU such that it provides a "trusted execution environment", also known as TEE. The code that is executed inside the TEE is protected, with hardware and software enforcement, from other untrusted code that runs on the same platform. This approach is shown in Figure 7, in the middle.

On commodity computing platforms, such as PCs, the prime example of currently widely available secure hardware technology is Intel's SGX [205]. SGX creates a TEE where applications, called *enclaves*, are protected from other code running on the same platform. SGX enclaves are supported by practically all latest Intel CPUs and the SGX architecture is open in the sense that it allows third parties to develop their own enclaves.

Most modern smartphones support two types of secure hardware. First, smartphones are typically equipped with a smart card (SIM card). Most smart cards are closed systems, as installing new code inside them requires permission from its issuer (e.g., mobile network operator). Second, many smartphones support the ARM

TrustZone [206] architecture which realizes a TEE on the main CPU of the mobile device. Permission from the mobile device manufacturer is typically needed to run new applications inside a TrustZone TEE, and thus TrustZone can be considered a closed platform as well.

Security-critical server platforms often use external hardware secure hardware modules (HSMs). Compared to the previously listed commodity secure hardware variants, modern HSMs are specialized and expensive hardware equipment that typically provide various protections against physical tampering and other attacks [207].

Other secure hardware variants exist too. For example, portable standalone devices can be used to store cryptographic keys for use cases such as two-factor authentication [82] and cryptocurrency client credential storage [208]. For the latter, see more details from Section 5 where digital wallets are discussed.

Security properties: Most secure hardware designs, whether realized as a separate chip or integrated in the main CPU, have two main security properties. The first is *data confidentiality*, which means that secrets like cryptographic keys that are stored and processed inside the secure hardware should be protected from unauthorized access. The second is *execution integrity*, which means that external parties that communicate with the secure hardware can be guaranteed that the intended code was executed correctly inside the secure hardware.

One simple approach to realize execution integrity is to rely on a closed secure hardware platform where only pre-vetted code can be executed. In such systems the code correctness is based on *whitelisting* by a trusted authority like the secure hardware vendor. Another, typically more flexible approach is to allow execution of any code inside the secure hardware, and to let other parties verify the correctness of the code through a process called *remote attestation* [209]. Remote attestation is an interactive protocol that can be executed between the secure hardware and a remote verifier who wants to examine which code is run inside the secure hardware. The secure hardware is trusted to execute the examined (attested) code faithfully, and thus ensure its execution integrity.

8.2 Limitations of secure hardware technology

Secure hardware can enable significant security benefits, as long as the protections of the secure hardware itself hold. In this section we discuss possible ways that adversaries can attempt to break secure hardware protections and the estimated feasibility of such attacks.

Software-based side channels: One common way to attack secure hardware is through side channels. TEEs like Intel SGX and ARM TrustZone share their hardware with the rest of the computing platform, which creates a potential susceptibility to side-channel attacks, where malicious software on the same CPU attempts to infer secrets that are processed inside the TEE based on utilization of shared physical resources like caches. Researchers have recently demonstrated the possibility of such

information leakage from SGX enclaves [210], [211]. In addition, subtle side effects of transient execution can leak secrets to untrusted software on the same platform [212], [213].

One possible defense against side-channel attacks is to write code that includes defensive measures such as elimination of secret-dependent branching. Automated tools that help the developer to harden enclave code against side-channel attacks are an active area of research [214], [215]. While perfect elimination of all possible sources of leakage is difficult, code hardening can be an effective defense against most known software-based side-channels.

Physical tampering: In scenarios where the adversary has physical access to the secure hardware, physical attacks become another relevant concern. Modern TEE architectures like SGX and TrustZone store and process secrets inside the main CPU package. Physical attacks that attempt to extract secrets like cryptographic keys from the CPU package are generally considered difficult and expensive. Physical side-channels (e.g., ones that monitor the power consumption of the secure hardware during its processing of secrets and thus attempt to infer the value of the secret) are another and less invasive potential attack vector. Full elimination of all possible forms of physical side-channel leakage is considered difficult [216].

Most smart card manufacturers consider simple physical key extraction and side-channel attacks as part of their threat model and include protections against them. Researchers have shown that well-resourced adversaries (e.g., ones equipped with sophisticated laboratory equipment) may be able to extract secrets from smart cards [217]–[219]. Such attacks are likely to be infeasible to many adversaries. HSMs include sophisticated and expensive physical protections in their manufacturing process and are considered difficult to tamper with.

Trust concerns: Besides the above discussed attack vectors, another common concern for the use of secure hardware technology is the trust model, where the hardware vendor must be implicitly trusted. The TEE architectures that are widely available in commodity computing platforms today come from few vendors (mainly Intel and ARM). Thus, system designers have little choice in terms or which secure hardware to use and which vendor to trust. Such trust issues are emphasized by the fact that the designs details of TEEs are not entirely public. For example, many aspects of the Intel SGX's design remain undisclosed.

This situation may improve. The computing platform and chip manufacturing landscape is getting more diverse, and in near future system designers may have more hardware and chip vendors to choose from. Also open-source initiatives such as the Keystone TEE [220] architecture on the RISC-V platform may enable secure hardware variants where the entire design is public.

Another trust concern is related to the supply chain. If the adversary manages to tamper with the secure hardware after its manufacturing and before it reaches the customer, the protections provided by the secure hardware may be weakened or circumvented. Such supply chain attacks require sophisticated hardware tampering

skills and access to the logistics process, and thus such attacks are considered feasible only for the most resourceful adversaries (e.g., nation states).

8.3 Problems of simple use

Now, that we have explained the main benefits and limitations of secure hardware technology, we proceed to discussing its use in digital currencies. We start with a simple but commonly suggested example that is also one of the digital currency implementation approaches mentioned in a recent patent by People's Bank of China [221], as discussed in more detail in Section 11.3.

Simple design: Consider a token-based digital currency system where a trusted bank issues signed coins to users. One user, say Alice, can perform a payment by passing the sufficient number of coins (tokens) to another user, Bob. Bob accepts the coins by checking the bank's signature on each coin, but Bob does not contact the bank. The attractive aspect of such a design is that payments can take place entirely offline, as they require no communication with the bank, and payments enjoy strong privacy guarantees, as no third party is involved in the payment.

The main issue with any token-based design is the possibility of *double spending*. If Alice is able to spend the same coins twice, the whole digital currency system is obviously broken. The problem of double spending could, in principle, be addressed by using secure hardware on each user's device. During payments, the coins could be transferred from one (attested) secure environment to another, and the secure environments could enforce that each coin is spent at most once.

Main problems: Unfortunately, such simple use of secure hardware technology has serious problems. The first major problem is that it creates *economic incentives* for users to attack their own secure hardware. When the adversary is the owner of the computing platform, the adversary typically has significant control over other software that is running on the same platform and permanent physical access to the secure hardware. Such aspects may make the above discussed attack vectors easier to exploit. Attacks against secure hardware become economically viable if the adversary gains a larger benefit (in terms of double spent coins) compared to the time and equipment investment needed to attack the secure hardware.

The second problem with such simple design is that it does not support *graceful degradation*. Even if only one secure hardware environment gets compromised, the adversary is able to spend the same tokens an unlimited number of times and thus break the digital currency system completely. Alternatively, if the secure hardware vendor turns out to be malicious, if the attestation mechanism gets compromised, or if an attacker can infiltrate a link on the supply chain and change the hardware, double spending of coins without restrictions is possible.

8.4 Better ways to leverage secure hardware

Motivated by the problems of simple use of secure hardware, in this section we discuss better ways to leverage secure hardware in digital currencies. In particular, we present examples where the security of the entire digital currency system does not rely on the use of secure hardware (graceful degradation) and platform owners do not want to attack their own hardware (economic incentives).

Infrastructure hardening: Our first example is using secure hardware to harden the backend infrastructure of a digital currency system as a complementary *defense-in-depth* security mechanism. Assume a digital currency where a bank maintains an account balance for each user. Such critical data repositories are typically replicated among multiple servers (consensus nodes) using a Byzantine-fault tolerant (BFT) protocol (recall Section 3). Each node could store its credentials and execute the consensus protocol within secure hardware. Such protection mechanism raises the bar for successful attacks, as now the adversary not only has to obtain software control of a large fraction of the consensus nodes, but he *additionally* has to break the secure hardware protections in them. The owners of the computing platforms do not have an incentive to attack their own secure hardware and secure hardware failure alone is insufficient to compromise the currency system.

Simplified and efficient privacy: Many digital currency systems use cryptography for user privacy protection, as discussed in Section 6. One of the first examples is Chaum's *e-cash* [222] that is based on a bank that issues coins to user. In this system a cryptographic construct called *blind signatures* enables anonymity protection for the payer (but not payment recipient). Another, more recent example is Zcash [168], which is based on a distributed ledger where all transactions are stored in encrypted format that make use of zero-knowledge proofs to show the correctness of a transaction without revealing its details. The main drawback is that the required zero-knowledge proofs make such systems complicated to deploy and have high resource requirements which often makes their use on e.g. mobile devices infeasible. For example, clients in Zcash are required to download and process the whole chain to benefit from Zcash's privacy guarantees and transaction creation is relatively slow (several seconds on recent desktop PCs).

Such cryptographic protections could be *replaced* with transaction processing that takes place inside secure hardware. In a simple solution, the secure hardware maintains a balance for each user and users send encrypted transactions to the secure hardware where they are processed. Compared to centralized schemes like Chaum's e-cash, such a solution achieves better privacy (also recipient anonymity and value privacy). Compared to ledger-based anonymous payment systems like Zcash, such solution provide greatly simplified deployment and better performance.

Complementary privacy: Another option is to *complement* cryptographic privacy protections with the use of secure hardware for increased privacy protection. For

example, transactions that use cryptographic commitments [223] offer payer and recipient privacy and value privacy, but no "transaction graph privacy" which prevents adversaries from linking transactions to each other, as explained in Section 6. The users could send commitment-based transactions to secure hardware that is operated by a bank and the secure hardware would process the transactions. Such a deployment would provide improved privacy (added transaction graph privacy protection) and graceful degradation: if secure hardware is compromised, only the additional gained privacy protection is lost and all the other privacy protections still apply due to the use of cryptography.

Compliance rules: CBDC deployments should, most likely, protect user privacy and at the same time enforce compliance policies and audit mechanisms, such as ones that are commonly used for anti money laundering and tax evasion detection purposes. As explained in Section 6, supporting both user privacy and compliance simultaneously is technically challenging. One possible way to address this challenge is to leverage sophisticated cryptographic constructs such as zero-knowledge proofs that allow users to show that their payments are compatible with compliance rules without revealing their identity or other transaction details. However, such schemes have drawbacks like complicated deployment and poor performance.

The use of secure hardware could allow greatly simplified systems where compliance policies could be enforced without violating the privacy of the users. The users could send their transactions in an encrypted format to secure hardware that would process them and at the same time enforce that each transaction complies with regulatory rules. Such policy enforcement would be easy to implement, because the necessary details like how much each user has spent within a given time period could be visible in plaintext inside the secure hardware.

Lightweight clients: Digital currency systems that store all transactions on a public ledger have benefits like *public verifiability*, which means that any third party can verify that all transactions are correctly formed. The downside of such a solution is heavy download requirements. Especially mobile clients, such as smartphones, would not want to download the entire ledger that can easily be multiple gigabytes in size. Selectively downloading only transactions that are relevant for the user, might reveal the identity of the user, even in systems with strong cryptographic transaction protections.

In such a setting, mobile clients could send encrypted queries to secure hardware on an infrastructure server that has access to the entire ledger. The secure hardware could then return the relevant transactions back to the client, without revealing the identity of the client to the operator of that infrastructure server. Such schemes have been developed for permissionless cryptocurrencies in recent research [224], [225] and the same principle could be applied to centralized CBDC deployment as well, if the CBDC system uses a public ledger.

Hardware wallets: Similar to infrastructure hardening, secure hardware could also be used to harden the client devices. Systems where client credentials are protected by some form of secure hardware are typically called "hardware wallets" and such hardware solutions are already widely used in permissionless cryptocurrencies, such as the hardware wallets produced by Ledger [226] or Trezor [227]. Hardware wallets protect the user's private keys from external adversaries that may be able to install malicious software on the computing platform of the user (PC or smartphone). To create a transaction, the user connects the hardware wallet to his PC or phone, where the transaction gets created and then sent to the hardware wallet. The user is expected to confirm the transaction amount and recipient from a small screen on the hardware wallet before the hardware wallet authorizes the payment by signing the transaction.

While hardware wallets can help to keep a user's credentials safe, they do not solve all problems of key management and payment safety. For example, users still need to be careful when confirming transactions details and they should ensure that they have a backup in case they lose their hardware wallet.

Smart contracts: Besides payments, CBDCs might support more complicated financial applications implemented as smart contracts, as discussed in Section 7. One challenge with the current smart contract solutions is that all contract data is typically recorded on a public ledger. This prevents deployment of a large class of financial applications that require confidential contract data. Recent research has explored techniques that could allow smart contracts to support confidential data on-chain [176], [181], [228]. However, such solutions require expensive and complicated cryptographic techniques like zero-knowledge proofs that can be difficult to deploy and suffer from poor performance. Secure hardware could be used to enable an easier way to execute smart contracts with confidential data, as demonstrated by recent research [180].

8.5 Summary and recommendations

Secure hardware is clearly not a panacea for any digital currency including CBDC. The currently available secure hardware technologies offer significant security benefits compared to standard program, but this technology also has its limitations. Thus, our recommendation is to be cautious of solutions where the integrity and security of the currency depends solely on secure hardware.

However, this does not mean that the use of secure hardware cannot offer benefits for CBDCs. When deployed in carefully chosen ways, secure hardware can either replace expensive cryptographic protections or function as a complementary security mechanism in addition to cryptographic protections. Therefore, our recommendation is that institutions currently investigating the possibility of CBDC deployments should understand the potential benefits of secure hardware technology.

9 Opportunities for Novel Financial Technology

As discussed in the introduction to this paper, a retail CBDC offers several commonly identified benefits: (i) gains in transactional efficiency: higher speed, lower cost, and finality, (ii) broader tax base, reduced tax evasion, (iii) backstop to private sector managed payment systems, and (iv) enhanced financial inclusion.

In principle, however, even more important than these benefits are the implications of CBDC for monetary policy and financial stability. CBDC can enable mechanisms for implementing monetary policy that are analogous to those available today, but novel in terms of their practically achievable parameters. Additionally, CBDC offers opportunities to implement a range of innovative monetary policies that operate at finer granularity, with greater transparency, and with more sophisticated programmatic logic than is technically possible in existing financial systems. These opportunities have implications for both the implementation and transmission of monetary policies. There are also a few risks, which we discuss below.

9.1 Implementing monetary policy

The basic mechanics of monetary policy implementation will not be affected by a switch from physical currency to CBDCs. Given that a relatively modest share of the supply of broad money is in physical form, this should not be surprising. However, other technological changes that are likely to affect financial markets and institutions could have significant effects on monetary policy implementation and transmission.

Retail CBDC disseminated through electronic wallets would make it easier to implement monetary policy more effectively in two ways. First, the nominal zero lower bound, which became a binding constraint for traditional monetary policy in advanced economies during the worst of the global financial crisis, would no longer apply. The central bank could institute a negative nominal interest rate simply by reducing balances on these electronic wallets at a pre-announced rate.

In an economy with physical cash, this should in principle not be possible since consumers (and firms) have the alternative of holding physical currency banknotes, a zero nominal interest rate instrument. In a scenario where there was no zero-interest central bank-issued alternative such as cash, it would be easier to implement a negative nominal interest rate on CBDC. If a CDBC co-existed with cash, there would be a limit (determined by cash storage costs, frictions in the use of cash etc.) to how low the central bank could drive the interest rate on the CBDC. In principle, negative nominal interest rates that would become feasible with certain forms of CBDC should encourage consumption by making it expensive for households to maintain cash positions.

Monetary policy could also be implemented through "helicopter drops" of money, once seen as just a theoretical possibility of increasing cash holdings in an economy in a non-distortionary fashion by making lump sum transfers to all eligible individuals or households. This would be easy to implement, at least in principle, if all citizens in an economy had official electronic wallets and the government could transfer central bank money into (or out of) those wallets. Channels for injecting outside money

into an economy quickly and efficiently become important in circumstances of weak economic activity or looming crises, when banks might slow down or even terminate the creation of outside money.[15]

One challenge, if a CBDC is issued through a two-layer approach in which the digital wallets are maintained by commercial banks, is the possibility that an individual or household might maintain multiple wallets at different institutions. Some coordination would be required to avoid double-dipping or multiple-dipping from helicopter drops of money. This concern would be obviated if the central bank directly managed identities and accounts. In both cases, however, there would be adverse implications for privacy.

Thus, a central bank could substantially reduce deflationary risks by resorting to such measures in order to escape the liquidity trap that results when it runs out of room to use traditional monetary policy tools in a physical cash-based economy.

There is an important asymmetry in this context that could become even more consequential if outside money were to have only a small role in the overall money supply. In that case, if banks were expanding outside money rapidly at a time of strong economic activity with rising inflationary risks, the central bank's ability to shrink electronic wallets holding CBDC might not do much to control the overall money supply. Although most advanced economy central banks now use price-based monetary policy measures (policy interest rates) rather than quantity-based monetary policy measures, this might be another reason for central banks to issue CBDCs rather than letting central bank money wither away if households were to use less and less cash.

There is a flip-side to the ease with which a central bank can increase or decrease the supply of outside money. The ability to impose a haircut on CBDC holdings, or to increase them rapidly in case the government were to apply pressure on a central bank to monetize its budget deficit, could lead to substitution away from the CBDC. The reduction in nominal balances and the erosion in the real purchasing power of nominal balances through monetary injections would have similar effects—decreasing confidence in the currency as a safe asset that can hold its value, at least in nominal terms. This could pose potential risks to monetary stability.

9.1.1 Transparency

With broad adoption, retail CBDC may capture a significant segment of economic activity at a national level and even, for CBDCs that serve as global reserves, at an international level. Data harvested from the resulting panoramic view of monetary flows on the underlying ledger can in principle provide policymakers with unprecedented data and insights. One consequence, as noted above, is an ability to limit tax

[15]There is a precedent for this rooted in Silvio Gesell's accelerated "free money" idea [229], physically implemented as stamp scrip, which showed promise to jump-start local economies in some historical experiments [230]. Also, there are a number of non-governmental cryptocurrency proposals and projects for permissionless cryptocurrencies with some form of "universal basic income" built-in: see, for example, [231]–[234]. A government implementing a CBDC is better-positioned to implement policies like this at large scale, of course.

evasion, but there are many others.

The ability to monitor monetary expenditures at the level of individual consumers is already available to credit-card issuers and provides predictive power at the level of the individual consumer, as well as offering microeconomic and macroeconomic insights. Goldman Sachs, for example, advertises a service called Quantinomics that leverages credit-card data to forecast corporate earnings growth, purportedly more accurately than with traditional methods [235]. Lenders have long exploited data on individuals' expenditures to make accurate predictions about their likelihood of divorce, patterns of travel, and creditworthiness [236], [237]. A CBDC that has disintermediated or overlaps with private payment systems can potentially relay the result of detailed analytics in real or near-real time to monetary policymakers.

Naturally, there is a tension here between transparency and privacy similar to those discussed in section 6. Given the potential scale and scope of CBDCs, however, an analytics system with global view of transaction data—or even metadata—could be repurposed as or support mass surveillance tools. Consequently, it is of vital importance that the transparency benefits of CBDCs be balanced against and leveraged with an eye to privacy requirements.

9.1.2 Non-fungible money

Certain forms of CBDC permit the implementation of monetary policy that affects accounts or units of money *selectively* and *conditionally*, that is, in a non-fungible manner.

Money transmitted in "helicopter drops," for instance, can carry spending conditions that permit only certain classes of expenditure. To amplify their effect in stimulating economic activity, lump sum transfers might for example carry the requirement that they be spent on, e.g., durable goods, as such spending has been shown to demonstrate limited responsiveness to economic stimulus during recessions [238]. As discussed in section 11, the People's Bank of China has filed a patent application—perhaps meant for use with the DCEP (Digital Yuan)—that suggests the idea of a central bank issuing currency for loans that carry central-bank-set interest rates and borrower qualifications.

Spending policies can alternatively be linked to aggregate data by analogy with inflation-adjusted financial instruments or even data about the holder of the funds (e.g., for the implementation of age limits on retirement-account withdrawals). As another example, the USDA's existing Supplemental Nutrition Assistance Program (SNAP) (formerly "food stamps") could be implemented in a CBDC through the distribution of dollars that are only eligible for transfer to accounts held by authorized food retailers. The policy attached to such dollars can be modified in real time, enabling immediate grants and revocation of food retailer authorization. With certain technical enhancements, such a policy could also stipulate the types of goods eligible for purchase using SNAP funds or make additional funds available in order to incentivize the purchase of particular types of food.

CBDC can in principle permit all money to take the form of such financial instruments, with individual policies that determine nominal value and spending conditions

based on nearly any desired form of data. At the extreme limit, *it may ultimately be feasible for every penny to be its own smart contract.*

There are potential downsides to non-fungibility, and fungibility is in fact treated as a design goal in many cryptocurrencies. This is because non-fungibility can be at odds with privacy and choice for currency holders. The ability to differentiate among units of currency based on serial numbers or transaction histories facilitates tracing, and indeed the distinctive transaction histories of Bitcoin enable blacklisting of units tainted by criminal activity [239] and transaction tracing by companies specializing in that activity, e.g., Chainalysis [160]. Similarly, the SNAP program places limitations on currency holders' purchasing behavior. Non-fungible currency would offer new mechanisms for government control of citizens' spending behavior that could catalyze new classes of "nanny state" interventions that may be unduly heavy-handed or micromanaging, and/or infringe on consumers' civil liberties.

9.2 Monetary policy transmission

A central bank endeavors to use the policy tools at its disposal to deliver objectives such as low and stable inflation, low unemployment, and financial stability. The transmission of monetary policy to economic variables such as GDP growth, unemployment, and inflation occurs mainly through the banking system but also through other financial channels.

A number of banks and consortia of banks are exploring the use of distributed ledger technologies (DLT) for bilateral settlement of clearing balances without going through a trusted intermediary such as the central bank. DLTs, as discussed earlier, make it easier to track and verify transactions. If all participants in a closed pool can monitor such activities and if there is a permanent indelible transaction record that is tamper-proof, they may be able to use group monitoring as an alternative for a trusted central counterparty.

Will such developments dilute the ability of the central bank to affect interest rates in the economy through its control of very short-term policy interest rates (such as the discount rate and the Fed funds rate in the U.S.)? This gets to the crux of the question about whether central banks can maintain their influence over aggregate demand and inflation even if they are sidelined from some of their traditional roles—issuing (outside) money and providing payment and settlement services for major financial institutions.

If banks and other major financial institutions do create such payments and settlement mechanisms among themselves (both bilaterally and across members in the group), and are also able to manage their liquidity positions and overnight balances more effectively, then settlement and liquidity management through the central bank might play a less important role. Still, competitive forces might limit the use of DLTs as an alternative for a trusted third party such as a central bank to provide settlement services while maintaining the confidentiality of those transactions. If these challenges are overcome, one possibility is that the central bank eventually becomes a liquidity provider of last resort in times of crises but, otherwise, commercial banks route their settlement and liquidity management operations through direct channels

among themselves.

A related issue is whether nonbank and informal financial institutions, such as Fintech lending platforms, are less sensitive to policy interest rate changes than traditional commercial banks. If these institutions do not rely on wholesale funding and have other ways of intermediating between savers and borrowers, then the central bank might face significant challenges to the effectiveness of monetary policy transmission. The relative sensitivity of the nonbank financial sector to changes in policy interest rates and other operational tools of monetary policy needs further study as the structures of financial systems undergo changes that could significantly affect the implementation and transmission of monetary policy.

9.3 Selective review of academic literature

The academic literature has only recently begun to grapple with the implications of CBDC as well as Fintech more broadly for monetary policy. Some authors argue that a CBDC will not in any material way affect the implementation of monetary policy, although there could be other macroeconomic effects. The conclusions, as indicated by the limited and selective survey below, depend to a great extent on the model structure and the manner in which the CBDC is introduced into the economy.

Barrdear and Kumhof [240] develop a DSGE model with multiple sectors and several nominal and real rigidities to understand the effect of introduction of CBDC. These authors suggest that infusing CBDC into an economy could result in substantial steady state output gains of nearly 30 percent. This effect persists if the central bank issues a large amount of CBDC against government bonds.

Andolfatto [241] studies the implications of CBDC in an overlapping generation model with a monopolistic banking sector. In this model, the introduction of interest-bearing CBDC increases the market deposit rate, leads to an expansion of the deposit base, and reduces bank profits. This is because competition from the CBDC causes banks to raise deposit rates. However, the CBDC has no effect in terms of bank lending activity and lending rates. Although the introduction of the interest-bearing CBDC increases financial inclusion, diminishing the demand for physical cash, it does not disintermediate banks.

Bordo and Levin [242] consider how digital cash could bolster the effectiveness of monetary policy. They lay out some steps for implementing digital cash via public-private partnerships between the central bank and supervised financial institutions. They conclude that digital cash could significantly enhance the stability of the financial system.

9.4 Smart contracts: realizing other novel capabilities

Beyond non-fungible units of currency, CBDC platforms that incorporate or serve as a substrate for smart-contract functionality can realize a range of monetary policy tools and novel financial instruments. Such instruments could serve as new conduits for monetary policy, but could also fundamentally impact monetary policy transmission. Experience with smart contract platforms such as Ethereum illuminates some of the

possibilities for new financial instruments in CBDC platforms. It also highlights the fundamental questions and challenges these instruments could surface around platform control and governance.

We encourage readers to refer to also section 7 for a more in-depth treatment of smart contract mechanics.

Atomic and instantaneous execution: Most smart contract platform architectures today permit multi-step transactions to execute *atomically*, that is, in an all-or-nothing manner. In a serialized DLT, that is, one in which transactions are fully ordered, and not processed in parallel, it is additionally possible for transactions to be executed *instantaneously*, in the sense that no changes to platform state from other transactions can occur at the same time.

These properties enable new capabilities without parallel or precedent in existing financial systems. One example is known as a *flash loan*. A flash loan is a loan that is initiated and repaid *within a single transaction*. Between the steps of borrowing and repayment, the transaction can execute any desired logic supported by the underlying DLT—for example, arbitrage on DLT-resident currency exchanges. If at the end of the transaction the loan is not repaid (with the requisite interest), the transaction aborts and, thanks to atomicity, has no persistent effect on platform state. If the borrower defaults on the loan, in other words, the loan is retroactively unwound, and in effect never took place. Because the lender assumes no risk, flash loans require no identification of or collateral from borrowers. They also typically carry very low interest rates, e.g., 0.1% [260], as the duration of the loan is nearly instantaneous.

Similar properties could provide a central bank with new mechanisms for implementing monetary policy. For example, it would be possible to effect systemic changes across a platform instantaneously. This could prove useful in preventing financial intermediaries from exploiting arbitrage opportunities resulting from temporally inconsistent implementation of policy changes—even when a central bank publicizes these changes in advance.

Distributed Autonomous Organizations (DAOs) and Decentralized Finance (DeFi): In the predominant model of smart contract execution, smart contracts run autonomously. They take the form of unmodifiable code that may be called by any user.[16] As discussed in section 7, this execution model is critical if smart contracts are to realize their intended role as virtual trusted third parties, but also carries risks, as historical incidents discussed in section 7 have shown.

A side-effect of smart contracts' autonomy is their ability to realize financial instruments or markets that run programmatically outside the control of the contract creator or any other single entity, resulting in efficiencies in execution and enforcement of terms unavailable in conventional contracts.

One class of such smart contracts is known as a *Distributed Autonomous Organization* (DAO). The best known example, called *The DAO*, was launched early in the

[16]Smart contracts can be instrumented to permit code updates and enforce access controls, but such features are only available if they are hard-wired into a contract's original code.

history of the Ethereum blockchain. It implemented a form of crowdsourced venture fund, allowing users to invest money in the contract and vote on the allocation of the resulting pool of money. The DAO accumulated some 15% of all the cryptocurrency in the ecosystem. (See section 7 for a discussion of a vulnerability that caused the failure of The DAO.)

DAOs are one design pattern for *decentralized finance (DeFi)*, a broad label that applies to smart contracts that lend money, support stablecoins, i.e., tokens that aim to maintain parity with fiat currencies, or run marketplaces where trades execute on a blockchain. DeFi instruments at the time of writing make up a small but rapidly growing $1+ billion market [261]. Many DeFi instruments lack exact counterparts in traditional financial systems, making the DeFi ecosystem a crucible of financial innovation. (See also section 7 for discussion.)

Support for smart contracts within or on top of a CBDC could give rise to similar innovations in a setting that is more tightly aligned with existing regulatory and legal frameworks and financial controls than cryptocurrency ecosystems are today.

Challenges and questions: Such incidents as The DAO hack and the weaponization of flash loans, both discussed in section 7, raise a number of questions that will inevitable surface in a CBDC platform with smart-contract functionality. While a cryptocurrency-based smart-contract system Ethereum is decentralized in a degree that a CBDC platform is unlikely to be, smart contracts nonetheless make it easier to create financial instruments whose control is shared or ambiguous. The rules and regulations governing the platform as a whole then become critical in assuring its integrity.

These questions include the following:

- *Accountability:* Should all smart contracts be required to have owners who assume responsibility and liability for their effects in the system? How will these owners be identified? (See section 4.)
- *Oversight:* Should smart contracts be instrumented by design with reporting functionality for regulators? What privacy considerations then come into play? (See section 6.4.)
- *Functionality:* Should smart contracts have the richest possible functionality supported by the underlying DLT, or should their functionality be constrained, e.g., through imposition of a domain-specific programming language?
- *Intervention:* Under what circumstances would a central bank administering a CBDC intervene should code running on the underlying DLT prove to be buggy, fraudulent, criminal [262], or otherwise problematic? Techniques such as those in, e.g., [199], [200], [263] may be worth considering.

Such questions arise even in CBDC designs where smart-contract functionality is overlaid on a DLT by third parties, one option considered in, e.g., [10], as application-layer functionality can have a systemic impact on the underlying currency.

9.5 Summary

The opportunities CBDCs offer for financial innovation may be summarized as follows:

- *Implementing monetary policy:* CBDCs offer the possibility of creating currency with *nominal negative interest rates*, as a means of stimulating consumption. They also can in principle allow the creation of various types of *non-fungible currency* with particular constraints on or incentives for their expenditure. CBDCs potentially panoramic view of national economies could *yield deeper insight for regulators and policymakers* into historical and ongoing economic activity than today's monetary systems, thus enabling better informed implementation of monetary policies.

- *Monetary policy transmission:* Should banks and other major financial institutions be able to create payment and settlement mechanisms among themselves with suitably strong transaction confidentiality, they could assume many of the settlement and liquidity management operations that are traditionally the province of central banks. CBDCs could thereby potentially blunt central banks' ability to transmit monetary policy.

- *Smart contracts for other novel capabilities:* By acting as a substrate for smart contracts, CBDCs could provide a way to translate a variety of financial innovations arising in cryptocurrency-based platforms into a setting more closely aligned with existing regulatory and legal systems.

With such capabilities, of course, come risks. Chief among them are:

- *Privacy concerns:* In a richly featured platform—e.g., one with a global analytics capability—the ability of platform operators to gather information about end users could resemble or extend even beyond those discussed in section 6.

- *Micromanagement:* The rich range of policies realizable by highly targeted capabilities of non-fungible currency and smart contracts could tempt policymakers into interventions that are unduly complex and representative of special interests (like the U.S. tax code [264]) and/or influence consumer behaviors in ways that infringe on civil liberties or are otherwise harmful.

10 Legal Considerations

The specific legal requirements for a CBDC depend on the jurisdiction and often can be modified by the jurisdiction establishing the CBDC. Thus, rather than discussing specific doctrines, it is more helpful to consider a few high-level issues involved in the design of a CBDC and its governing legal framework. The specific examples in this section are primarily drawn from the United States, but the same general issues will arise in most legal systems.

10.1 Jurisdiction

A CBDC is a national institution by definition, and will need to interface with its nation's laws and legal system. There will usually be no question of which nation's laws and legal system take priority in case of international disagreement: its own. Other legal systems may resolve disputes about CBDC assets, sometimes under their own law, and they may enter orders binding CBDC users. But to the extent that parties to a foreign dispute want the CBDC's own institutions to take any action to enforce those orders, it is reasonable to expect that they must first domesticate that judgment in a court within the CBDC's own national legal system.

That said, in a *federal* system such as the United States or the European Union, the CBDC will still be exposed to many subnational jurisdictions with varying laws. Relatedly, it may need to deal with a diversity of national and local courts. Some thought should be given as to how to authenticate orders and verify the authority of the courts issuing them, given that forgery of court orders is not unknown. Two options to simplify this task are to centralize all orders affecting CBDC assets in a single national tribunal, or to require all such orders to proceed through a common set of procedures before CBDC administrators are expected to act on them.

10.2 Compliance

Money laundering is the use of financial transactions to conceal the source of funds. Governments prohibit money laundering not because they care about funds as such, but because money laundering makes it easier for criminals to conceal their crimes, evade taxes, and profit from their ill-gotten gains. Relatedly, governments increasingly prohibit the use of financial transactions to support terrorist organizations. Roughly speaking, anti-money-laundering and countering the funding of terrorism (AML/CFT) laws come in three layers:

1. General prohibitions, such as the Money Laundering Control Act, that directly target money laundering itself by prohibiting the use of financial transactions to conceal the source of proceeds of criminal activity. Examples include "spending" cash received from drug dealing at a front business, or making wire transfers to make kickbacks look like legitimate business receipts. Similarly, the Antiterrorism and Effective Death Penalty Act makes it illegal to "knowingly provide[] material support or resources to a foreign terrorist organization."

2. Reporting requirements, such as the Bank Secrecy Act (BSA), which requires that financial institutions report to the government all transactions in cash of 10,000 USD or more. Reporting requirements also include Know Your Customer (KYC) rules, which require institutions to verify the identities of their clients, as well as more open-ended standards requiring institutions to report suspicious transactions. These rules are designed to help regulators find money laundering by enlisting surveillance at the financial institution level.

3. Anti-evasion (or "structuring") rules that prohibit attempts to circumvent reporting requirements, e.g., by breaking a larger transaction down into smaller

ones under the reporting threshold.

Collectively, these are the specific enforcement rules that governments currently thinks they need to achieve the broad functional goals of preventing money laundering. With a CBDC, it is useful to ask (a) whether these existing rules are enforceable against a proposed CBDC, (b) whether they are sufficient for the functional goals, and (c) if the answer to either previous question is "no," what other rules might be enforceable and sufficient. In general, if a payment or deposit system has any significant potential to facilitate transactions not meeting these goals, the financial regulatory system will attempt to strictly regulate and monitor transfers in and out of it. Thus, for example, if a CBDC itself offers strong anonymity (thus making KYC impossible at this layer), regulators may demand that exchanges which convert between the CBDC and other currencies implement KYC rules for all customers.

A hybrid two-tier CBDC in which the core digital currency is traceable can meet the requirements of existing AML laws. The institutions which provide customer accounts would be required to implement all of the reporting requirements under BSA, KYC, etc. Regulators could then monitor transactions entered into the system to trace funds as needed. In fact, the centralization of a single transactional ledger might help with monitoring, making the CBDC comparatively unattractive for money laundering.

Other CBDC designs raise serious AML issues. Designs that allow for the mixing of transactional inputs or with weak identity verification make some aspects of AML enforcement much more difficult. Designs with strong untraceability are almost certainly incompatible with AML regulation. This may be considered a virtue from their designers' privacy-oriented perspective, but is a deficit from the perspective of a financial regulator considering introducing a CBDC.

10.3 Privacy

Privacy *law* (as contrasted with privacy norms and privacy goals) interacts with CBDC design in two ways. The first is that existing legal rules about financial privacy already reflect considered balances between customer privacy and law-enforcement needs. In the United States, these rules are primarily statutory, as the Fourth Amendment generally does not apply to financial transaction records. (Under the "third party doctrine," a financial services customer "takes the risk, in revealing his affairs to another, that the information will be conveyed by that person to the Government." United States v. Miller, 425 U.S. 435, 443 (1976)). Instead, statutes like the Right to Financial Privacy Act (RFPA) specify the procedures law enforcement must follow to obtain such records and the legal standards of relevance they must meet. These rules apply in money laundering investigations, so the AML rules described above are compromises that already try to preserve some measure of privacy. But these statutory privacy rules also apply more broadly in any investigations, for whatever type of crime. So, for example, the RFPA standards apply when law enforcement seeks to examine the transaction records of a business it suspects of credit-card fraud, bribery, tax evasion, or hiding money from creditors.

A CBDC design could depart from this baseline by being either more or less technically protective of privacy. A design that is *less* protective — e.g., a public blockchain with identities easily linked to specific users — effectively provides law enforcement with freer access to transaction information. (One federal appeals court has held that under the third-party doctrine, there is no privacy interest in transaction information on the public Bitcoin blockchain. United States v. Gratkowski, No. 19-50492 (5th Cir. June 30, 2020.)) A design that is *more* protective — e.g. a public blockchain with strong untraceability and strong anonymity — raises questions about whether law enforcement access is sufficient. A hybrid two-tier CBDC design comes close to the status quo: federal law enforcement would need to proceed under the RFPA standard to obtain account information from financial institutions about specific customers. It would be appropriate to make clear (which might require statutory amendments) that a similar standard should apply for obtaining transaction records from the transaction ledger itself.

The second way in which privacy law pushes on CBDC designs is that it also imposes requirements *for* privacy against other parties. That is, privacy law sets a floor which the existing banking system is legally required to meet. In the United States, the RFPA, the Financial Services Modernization Act (a/k/a Gramm-Leach-Bliley or GLB), the identity-theft rules of the Fair and Accurate Credit Transactions Act, and the Electronic Fund Transfer Act all place limits on whom a financial institution is allowed to disclose customer information to, and when. These rules reflect a broad consensus that some degree of financial privacy is appropriate, not just against governments but also against private parties. In the European Union, the General Data Protection Regulation (GDPR) establishes a broad and high privacy baseline.

Again, a CBDC could depart from this baseline by being either more or less private. A design that is *less* privacy protective is not necessarily legally problematic: for example, a blockchain that makes transaction details public often reflects a decision by users to make their transactions public, rather than a privacy violation by the blockchain or by any institutions dealing with it. That said, it might be a tough sell for a payment system to come with lower privacy safeguards than users expect from existing ones. (Imagine the reaction from businesses if a central bank decided to standardize on Venmo, in which transactions are visible to all users by default.) A design that is *more* privacy protective is not necessarily legally problematic either: these statutes generally set a floor and not a ceiling. Again, a hybrid two-tier design comes closest to the status quo. In general, a CBDC is unlikely to perfectly replicate the balance struck by existing privacy law. Some aspects will likely be either more private or more public.

Unless a CBDC's designers secure legislative changes, therefore, a CBDC must be capable of complying with these privacy mandates. A few such requirements of note include:

- **Purpose Limitation:** Under the GDPR, personal information may only be used for the purposes for which it was collected; new uses require fresh consent.
- **Disclosure Limitation:** Under GLB, consumers have a right to opt out of having their personal information disclosed to unaffiliated third parties. Under

the GDPR, such disclosures generally may not take place without affirmative opt-in consent.

- **Access and Portability:** The GDPR gives each data subject a right to obtain any information "concerning him or her" in a structured digital format so that they may give that data to other services.
- **Rectification:** The GDPR gives each data subject a right to correct any erroneous data "concerning him or her."
- **Security Breach Notices:** The GDPR and numerous American state laws give each data subject the right to be notified promptly if their data is the subject of a security breach.

Some of these privacy restrictions significantly constrain the design space for a CBDC. It is not clear, for example, that a CBDC built around a public blockchain can ever comply with the GDPR's rules on disclosure limitation and rectification. Others require particular features in a CBDC rollout: any system that handles accounts for individuals, for example, will need to be designed such that it is possible to generate and send breach notifications if needed.

10.4 Fraud and mistake

While a well-designed CBDC, like any well-designed banking or payment infrastructure, can potentially reduce the incidence of fraud and mistake (or exacerbate them if poorly designed and implemented), it is not feasible to eliminate them entirely. Thus, a CBDC and associated legal regime must consider and balance two related concerns: *preventing* and *correcting* incorrect transactions. These transactions fall into a few recurring patterns, including:

1. *Disloyal Agents*: People frequently authorize others to act on their behalf. Sometimes this is informal: spouses ask each other to do their online banking. Sometimes it is legally necessary: guardians must have this power to do their job of protecting incapacitated or underage principals. And sometimes it is unavoidable: entities like corporations only act by and through human agents. Whenever an agent is authorized to take some action affecting a CBDC, the agent may exceed its authority and engage in unauthorized transactions.
2. *Impersonation* of an authorized user by an unauthorized one. Any credentials associated with an authorized user are themselves attack targets. 2FA and other authentication protocols are designed to furnish additional evidence that a user is who they say they are, but are necessarily less than completely reliable. (The design trade-offs involved are discussed further in Section 4.2.)
3. *Mistakes*: Phishing and related attacks induce users to engage in intended transactions with unintended parties, and sometimes parties make incorrect transactions even when there is no malicious intent (e.g., using the wrong recipient

account number in an electronic transfer). Some of these issues can be mitigated with good UI designs and identity management (discussed in Section 4.2), but some mistakes, such as making an incorrectly large transfer by fat-fingering an extra zero, can never be entirely eliminated.

4. *Fraud in the Factum*: Parties can be presented with the authorization for a transaction and misled into believing they are taking some other action when they authorize it. Attacks of this sort range from swapping the pages of a paper contract to simulating UI elements.

5. *Fraud in the Inducement*: Parties are sometimes tricked into entering into transactions under false pretenses. For example, a fraudster might pass off a cheap fake as a $10,000 watch, or "sell" a watch stolen from another party. In both cases, the buyer's payment to the seller is intentional, but the overall deal is fraudulent.

Importantly, *none of these patterns* can be reliably detected by any mechanism internal to the CBDC, because the legal validity of the transaction is determined by external facts that are not directly observable by the CBDC's participants.

In all of these cases, existing law typically gives the victim a right to recover from a purposeful wrongdoer. But the law is more complicated on the questions of (a) whether the victim has a right to rescind a transaction not involving fraud, (b) whether this right extends to recover assets from third parties to whom they have been transferred in the interim. These issues often cannot be disentangled from the specifics of the payment system used. Thus, for example, in the United States cash is both fungible and negotiable: if B steals $10,000 from A, A is entitled to recover $10,000 but not the specific bills stolen, and if B then buys and destroys a watch from C with the $10,000, A cannot recover from C at all.

Any CBDC must consider, therefore, not just how it will be designed to limit the opportunities for these incorrect transactions, but how it will recover when such transactions take place. Many decentralized blockchains have taken an extreme "user beware" attitude: all transactions are effectively irrevocable once entered on the blockchain and assets cannot be recovered from an unwilling recipient. This option is not viable for a CBDC intended for mass adoption, and it is worth noting that no existing non-blockchain banking system or payment system receives such treatment under existing law. Nor is it feasible to plan for recovery with the legally compelled cooperation of the recipient: an identity thief could well be an unknown overseas hacker not subject to compulsion from the legal system of the CBDC's jurisdiction.

Instead, the CBDC's administrators must be capable of modifying its state in accordance with property, contract, payment, and banking law. This raises several important subsidiary design questions.

First, the CBDC needs an appropriate governing legal regime. Cash, checks, debit cards, credit cards, wire transfers, private systems such as PayPal and Venmo, and other payment systems all have their own governing laws (typically a mixture of public regulations and private contracts). The specifics of liability for unauthorized transfers and the circumstances under which transactions can be halted, modified,

or reversed will need to be worked out by analogy and with careful attention to the technical details of the CBDC.

Second, the CBDC needs an appropriate management infrastructure. Someone will need to verify when a transaction should be modified in accordance with the jurisdiction's law and then make the appropriate changes. This requires a dispute resolution system within the CBDC's administration, an interface to the judicial system capable of authenticating legitimate orders, an interface to other agents capable of giving trusted instructions for transaction modification (such as banks), or some combination of all of the above. This infrastructure will require its own security, auditing, and compliance processes. In a hybrid two-tier design, some of these procedures can be handled by the institutions that layer customer services on top of the payment layer, but some of them will need to be implemented directly against the underlying ledger.

Third, the CBDC needs systems for reporting out its state to appropriate authorities. This may be more than just the state of the transactions that have been entered. The process of determining whether a transaction was properly authorized, for example, may depend on information such as the IP address from which it was requested and on the history of similar attempted transactions. Such information is relevant in litigation and dispute resolution for existing payment systems, and there is no reason to think that a CBDC will be any different. With a blockchain or other public ledger, the authorities can directly inspect its state (although this may fail to capture some details of how users interacted with it). In a hybrid two-tier system, existing reporting systems would largely suffice at the customer layer; the transaction layer would require one if the ledger is not already public.

10.5 Liens

Most kinds of property can be used as collateral for debts. Legally, the creditor is said to hold a *lien* in the property. The debtor remains the owner, but under appropriate circumstances (typically default), the creditor can seize the property and force a sale to help satisfy the debt. Some liens are *possessory*, e.g., a margin loan is collateralized by securities deposited with a broker. Other liens, such as mortgages, are *nonposessory*: the debtor retains possession and control of the property. Some liens are created explicitly by the parties as part of a loan transaction. But others arise by operation of law. For example, unpaid taxes on property may give the government a tax lien against the property, or a person who wins a lawsuit for damages can obtain a judgment lien over the defendant's assets.

Creditors collecting a debt or judgment can typically seize the defendant's property, subject to detailed and jurisdiction-specific rules on what property can be seized and how. Liens give secured creditors priority over other creditors, come with expedited procedures for a creditor to proceed quickly against the property subject to the security interest (e.g., repossessing and foreclosing on a car subject to an unpaid loan is faster and less regulated than seizing a car to pay off an unrelated debt), and are subject to fewer restrictions on what property can be seized (e.g. some states protect a debtor's primary residence from seizure for unsecured debt, but not from

mortgage foreclosure) [265]. A closely related concept is *garnishment*, in which a creditor obtains payment of a debt by seizing the debtor's assets from a third party. Garnishment orders can be particularly effective because the third party can hold the property as soon as the debtor acquires it, e.g., when a bank garnishes a parent's wages for payment of a child support order.

The widespread use of liens impacts CBDC design in two important and related ways. First, it raises doctrinal questions of how CBDC accounts and related assets should be treated as collateral. How can liens in them be created, and how can they be repossessed and foreclosed on? (See [266]–[269].) Existing law on security interests varies greatly by type of property, even within a jurisdiction, which means that categorization questions matter greatly. If the default legal regime that would govern CBDC assets is unclear or undesirable, then legislation to establish more appropriate treatment will be necessary. Second, the CBDC's design should facilitate the use of security interests in a manner that integrates smoothly with the rest of the financial and legal system.

It is important to note that while excluding CBDC assets from the lien system is doctrinally possible (almost anything is with appropriate legislation), this likely would be a complete non-starter for many financial institutions. For one thing, CBDC assets which can never be pledged as collateral for loans are worth less to borrowers; businesses which would otherwise be willing to deal in CBDC accounts might avoid them for that reason. Second, the security of secured lending frequently depends on the ability to trace collateral through changes in form. A creditor who holds a lien in the inventory of a farm-equipment dealer would be shocked and outraged if its lien failed to attach to the money the dealer receives in exchange when it sells a tractor out of its inventory. Categorically excluding CBDC assets from being encumbered by liens would give debtors a loophole capable of destroying almost any lien.

Assuming, therefore, that CBDC assets will need to be part of a secured credit system, the following are a few of the regulatory and design considerations involved.

10.5.1 Collection

The CBDC design should provide a mechanism for the satisfaction of debts from CBDC assets. The most basic considerations here are similar to those above: the CBDC should have a management infrastructure that can accept and authenticate instructions from the legal system directing one user's assets to be paid over to another in satisfaction of a debt. In some cases, such as judgment debts, these orders could come directly from a government official: e.g., from a court or from a sheriff levying upon the debtor's property. In others, legal systems provide mechanisms for private repossession: e.g., brokers can unilaterally sell securities held on margin if the debtor fails a margin call. The CBDC should consider whether and how to support these private repossession mechanisms. Whatever mechanisms are in place, the CBDC must be capable of providing properly authenticated documentation that levy or repossession has been made. This is most straightforward in a hybrid two-tier system: the financial institutions that manage customer accounts can freeze and seize accounts as required. In a centralized one-tier ledger, the administrator can do so directly. The

absence of an appropriate mechanism in a decentralized blockchain architecture is an argument against such architectures for a CBDC.

Once a creditor repossesses an asset subject to a lien, there are questions about what happens next. Some such assets, such as houses subject to mortgages, are required to be sold at a public auction; in other cases, as under the Uniform Commercial Code, the creditor may follow any "commercially reasonable" procedure to sell the assets. To the extent that a CBDC is fungible and has clearly-defined exchange rates, few such issues are likely to arise in CBDC design. The important part is that the legal system should be able to clearly determine how much of a debt has been satisfied by the sale of collateral.

10.5.2 Locking

One common use case for assets subject to a lien is to prevent the assets from being transferred without the consent of the creditor. The creditor's fear, of course, is that transfer will place the assets beyond its reach, either legally or practically. A CBDC's designers should consider whether this is a use case the CBDC should support. If so, then it requires an appropriate interface to receive and authenticate locking orders. Again, in a hybrid two-tier system, the financial institutions that manage customer accounts have experience dealing with such orders and infrastructure to handle them. A one-tier centralized ledger would require building an analogous infrastructure, while a decentralized ledger would require that creditors have an interface they can use to register their claims in a way that enables them to lock assets under appropriate circumstances.

Forward-looking remedies provide further challenges. Garnishment, for example, can attach to wages *as soon as the debtor becomes entitled to them.* This requires that the appropriate locking be attached to a user's CBDC account, not just to specific CBDC assets. As assets are deposited in the account, they are immediately garnished, before the creditor can deal in them. Note that since garnishment orders are typically not all-or-nothing (e.g. a fixed percentage of wages up to a set total), a CBDC design supporting this use case will need to implement the appropriate logic. In a hybrid two-tier system this logic could be added to existing processes in financial institutions; in a centralized design, it would need to be reimplemented; in a decentralized design, it might need to be hard-wired into the CBDC design in a deep (and technically challenging) way.

10.5.3 Notice

Liens are invisible and intangible. But people considering buying or lending against property want to make sure that there are no liens they don't know about. This raises the question of how to provide effective notice to third parties. (If a CBDC asset is locked, this issue does not arise: the locking itself is an effective form of notice.) Under United States law, the validity of security interests against third parties often depends on "perfection," which roughly requires giving specific statutorily required forms of notice.

A CBDC's designers should give thought to how liens in CBDC assets should be communicated to third parties. In the United States, there are, very roughly, three techniques in widespread use. The first is debtor-specific recording: the creditor files a form, indexed by the name of the debtor, with an appropriate office in the debtor's state. This, for example, is what the Uniform Commercial Code requires for the catch-all category of "general intangibles." The second is asset-specific recording: the creditor files a form with an appropriate office, indexed by a unique identifier for the asset subject to the lien. This, for example, is commonly used for cars (which have unique alphanumeric Vehicle Identification Numbers) and for registered copyrights (which have unique registration numbers). The third, and in some way the most complex, involves possession of the asset by the creditor or an appropriate custodian. For example, a lien on a bank account can be perfected by transferring the account into the name of the creditor. Debtor-specific recording requires the least investment to set up, as the infrastructure is already in place, but does not provide easily searchable information on whether a given CBDC asset is subject to a lien. Involvement of CBDC asset custodians, as in a two-layer or centralized design, requires the most supporting infrastructure, but it also allows these assets to be locked, preventing transactions without appropriate creditor consent [266].

10.6 Tracing

Some kinds of property, such as cash, are regarded as completely fungible. Other kinds, such as ancient art, are regarded as completely unique. (See Section 9.1.2). One important difference along this spectrum is the degree to which the legal system attempts to trace ownership of specific property through multiple hands. Tracing is a way of asserting that one particular aspect of an asset's identity – its transaction history – renders it less than completely fungible. Tracing may be necessary, for example, when the property has been stolen and the original owner claims it from a downstream transferee. It may also be necessary when the property was subject to a lien and the creditor seeks to obtain it from a downstream transferee.

Different CBDC technical designs can facilitate different degrees of traceability, with important implications for their legal treatment. If all assets in the system are globally unique, for example, then perfect tracing is possible and property can in theory be recovered from a remote transferee many steps down the line. In other systems, the interchangeability of assets prevents such perfect tracing. Bitcoin, for example, does not attempt to tag specific transaction outputs with specific transaction inputs. After a series of transactions, it is possible to say that that *some* of this BTC came from *some* of that BTC, but not that *these specific* Bitcoins are exactly the same as *those specific* Bitcoins. Some anonymity-preserving cryptocurrencies, such as Zcash and Monero, are designed to eliminate traceability entirely: assets cannot be identified across transactions. (See Section 6.1.)

Traceability has advantages and disadvantages. It provides simplicity and clarity when unwinding tainted transactions. It also obviates the need for the legal system to apply crude approximations, such as identifying stolen funds *deposited* in an account with the last funds *withdrawn* from that account following the deposit. On the

other hand, the impossibility of tracing paradoxically provides clarity for transferees. It means they do not need to investigate the remote provenance of the assets they are receiving for fear that some transaction somewhere far back in the chain of title was tainted. This is one of the chief advantages of cash: except in the most blatant cases, cash is cash and it provides a reliable payment mechanism from the recipient's perspective. CBDC design should consider the advantages and disadvantages of traceability in light of their legal consequences.

10.7 Taxation

The general principles of taxation apply without great difficulty to CBDCs. For example, in computing a person's income from the sale of property, they are typically allowed to deduct the purchase price (or "basis") of an asset from the sale price in computing their gain or loss. Nothing in a CBDC design is likely to disrupt such bedrock principles. Two broad overarching themes, however, are worth consideration by a CBDC designers.

First, tax authorities have confronted the question of how to categorize various digital assets for tax purposes. In the United States, for example, cryptocurrencies like Bitcoin have been treated as "property" rather than as "foreign currency" for income tax purposes [270]. Given the importance of a CBDC, its tax categorization should be made as explicit as possible.

Second, income-based taxation does not attempt to continually mark to market the value of all assets, i.e., impose tax on them based on their current valuation, regardless of whether the owner has actually realized that value by selling them. This would be administratively difficult, could lead to significant errors for assets that do not trade in thick liquid markets, and would be unfair to taxpayers who hold illiquid appreciated assets. Instead, tax is assessed on gain or loss from the change in value of an asset only when it is sold or some other "realization event" occurs (such as the release of a debt). The design of a CBDC should be sensitive to which events and transactions will be regarded as realization events. To the extent that a CBDC is highly liquid and is denominated in, easily interchangeable with, identical to, or replaces a fiat currency, there is no great difficulty of or injustice in frequent realization events. Such events might include the payment of interest on CBDC accounts, the transfer of CBDC from one account to another, or the mining of CBDC by blockchain participants. To the extent that the CBDC is illiquid, only conversions between the CBDC and other property should be deemed realization events.

10.8 Conclusion

Existing legal requirements are easiest to meet in a hybrid two-tier CBDC design, in which banks manage digital wallets for individuals and entities. These financial institutions already have the infrastructure to support oversight and reporting, to freeze and transfer assets when required by law, and to perform extensive customer service. A design in which users directly interact with a CBDC managed by a central bank requires it to take on these functions and to interface extensively with the

legal system. A decentralized design in which financial institutions or central bank administrators cannot directly modify the state of CBDC assets would likely require extensive and controversial legal changes.

11 Overview of Libra and Digital Yuan

Two projects have played a central role in catalyzing the global CBDC discussion: Libra and the digital yuan. By examining the designs of these two digital currencies, we see some ways in which the needs for a highly scalable digital currency can be met. With the scale and resources of a powerful group of participants, the Libra Association has set lofty goals for its influence. A stated goal of Libra's digital currency is to give access to the financial system to the world's 1.7 billion unbanked. It hopes to make financial transactions as easy as "sending a message." Although not backed by a central bank, the Libra Association has the resources to disseminate its technology across national borders and influence how digital money is spent globally. With the backing of the Chinese government, the People's Bank of China (PBoC) is poised to be the first large economy to issue a digital currency. It envisions its digital currency as a cash equivalent with the potential to be the first CBDC to gain global traction.

11.1 Libra

Financial model: In 2019, Facebook's announced its intention to issue a cryptocurrency called Libra. While Facebook created Libra, Facebook will not control it. Libra will instead be controlled by the Libra Association, a nonprofit entity based in Switzerland with a governing board of 27 leading corporations in tech, finance, and nonprofit. The Libra Association has identified banking the 1.7 billion globally unbanked, people without access to traditional bank accounts, as a fundamental goal. The Libra Association released its initial white paper in June 2019 and a second, updated white paper in May 2020 detailing their plans for the Libra cryptocurrencies [20].

Libra plans to produce single-currency stablecoins, meaning cryptocurrencies that will exactly track the value of existing fiat currencies. These stablecoins will be tied to reserves of major global currencies (e.g., LibraUSD or ≈USD, LibraEUR or ≈EUR, LibraGBP or ≈GBP, LibraSGD or ≈SGD). Each stablecoin will be fully backed 1:1 by reserves of cash, cash-equivalents, and short term government securities denominated in that currency. This means that to create a new stablecoin, for example 1 new LibraUSD, the association will acquire $1 for their reserve. In order to remove 1 LibraUSD from circulation, the reserve will release $1. With these guidelines for generation and removal, Libra stablecoins will not generate value, rather they will be digital representations of existing fiat currencies held in the Libra reserve.

As central banks begin to issue CBDCs, Libra hopes to integrate them directly into the Libra network. Libra will also maintain a capital buffer, in addition to the reserve of circulating Libra stablecoins, in order to ensure solvency. Libra stablecoins will be minted and burned by the reserve based on demand, with supply expanding

and contracting based on the market for each Libra stablecoin.

The Libra Association will also create a platform-specific cryptocurrency called the ≋LBR. It will be a digital composite of the single-currency stablecoins offered, set at a fixed ratio. The ≋LBR will be administered by a smart contract. LBR is intended for use in efficient cross-border settlement and as a low-volatility option for those in nations that do not yet have a single currency Libra stablecoin available. Because it is made up of single-currency Libra stablecoins, each fully backed by reserves, ≋LBR will also be fully backed by reserves.

Conversation from ≋LBR and Libra stablecoins into fiat currency will be handled by third party financial institutions called VASPs (Virtual Asset Service Providers), who will interact with end users. Wallets belonging to users other than known financial institutions and VASPs are referred to as "unhosted wallets." Unhosted wallets are supported, but with tight controls (transaction limits, maximum balance enforced by the protocol). Their aim is to facilitate Libra participation for users who may be unable to interact with the system via a VASP.

Single currency stablecoins predate Libra. Tether (USDT), a stablecoin pegged to the US dollar, currently trades on over 100 cryptocurrency exchanges. Like Libra, it is controlled by a central party that pegs its value to USD and backs each USDT with $1 reserve. Tether's reserves are different from Libra's in that they (controversially) allow inclusion of short term loans from third-parties in addition to cash and cash equivalents. Tether has also expanded beyond US dollar stablecoins, now offering stablecoins for additional assets including gold, the Chinese Yuan, and the Euro [271].

Technical foundations: The Libra system will use a fully permissioned, Byzantine Fault Tolerant (BFT) ledger / blockchain that relies on open-source software. Libra does not plan to transition to a permissionless system. The stated design goal for the system is to "serve as a foundation for financial services, including a new global payment system that meets the daily financial needs of billions of people." To achieve this goal, Libra prioritizes flexibility, security, and scalability to billions of accounts in its design. Libra will support programs written in its bespoke Move programming language.

The Move programming language is designed to implement custom transaction logic and smart contracts on the Libra blockchain. Its goals are safety and security, its design explicitly informed by past security incidents involving smart contracts, including those discussed in section 7. Move includes first-class support for operations on currency and tokens and regulatory compliance features.

Initially, only the Libra Association will be able to publish smart contracts that interact directly with the payment system. Over time, third parties will be able publish smart contracts. Libra's (BFT) consensus protocol, called LibraBFT, a variant of Hotstuff [28], is designed to facilitate high transaction throughput and low latency. As a permissioned system, it will require relatively little energy—far less than Proof of Work. Its security relies on the standard BFT assumption that fewer than 1/3 of validator nodes, the machines that maintain the ledger, are compromised.

Validators will be approved / permissioned by the Libra Association. Transactions

are finalized when approved by a quorum of validators, and confirmed transactions are final and visible on the ledger. Thanks to its use of a publicly verifiable ledger, Libra will be fully auditable by law enforcement, regulators, and users, meaning that anyone can view an authoritative sequence of all processed transactions. Rather than grouping transactions into blocks like previous blockchains (Bitcoin, Ethereum, etc.), the Libra blockchain will be structured as one continuous data structure.

11.2 The digital yuan

China's central bank, the People's Bank of China (PBoC) has pursued creation of a CBDC more actively than any other global economic power. They are the first major economy to pilot a sovereign digital currency. They have publicly released very little about their digital currency, and limited technical information is available since the PBoC has not publicly released a whitepaper. Therefore, the information below is based on a combination of news reports, official statements, and analysis of Chinese patents filed for technologies that we conjecture are related to CBDC planning and may yield insight into the design choices under consideration.

Background: The PBoC has named their CBDC the Digital Currency for Electronic Payments (DC/EP). As implied by the name, this digital currency will be used for some payments in lieu of traditional fiat. The PBoC formed a research team to explore how to issue a "legal digital tender" in 2014 [272]. One of the goals of this digital currency was to internationalize China's fiat currency, the RMB [273]. This research group was expanded and formalized into the Digital Currency Research Institute in 2017, with the objective of conducting research and technical trials for a digital currency. The Chinese Agricultural bank began testing a wallet application for the digital currency internally in April 2020 [2]. The mobile payment platform Alipay has also filed patents related to its role as a likely secondary issuer of DC/EP [274].

System overview: The DC/EP is expected to be centrally issued and widely available. The PBoC envisions the DC/EP as a replacement for cash and with equal status as a legal tender. The former head of the PBoC's Digital Currency Research Institute described that the DC/EP would be based on the model: "one coin, two repositories, three centers." "One coin" is the DC/EP, the "two repositories" lie in the central bank as well as the commercial banks who will distribute the digital currency and individual wallets; "three centers" refers to the data centers which will perform authentication, registration, and big data analysis [272].

According to Fan Yifei, the Deputy Governor of the PBoC, the system should operate in two tiers with two distinct layers of functionality: interaction with commercial banks and token-based interactions [275]. The PBoC is expected to issue and redeem DC/EP via large, commercial banks. The DC/EP would will be token based, with commercial banks and financial institutions circulating the tokens. This structure is similar to the way fiat is currently handled by central banks, with a two-tiered system in which central banks issue currency and distribute it to financial

institutions who manage user interactions [276]. By leaving user-facing activities to banks, the DC/EP will avoid disintermediating the financial system and increasing the responsibilities and risk exposure of the central bank [275].

DC/EP will earn interest only if it is moved from the digital wallet into a deposit account, where it can be used for payment only through a bank card linked to that particular deposit account. Its two-tier structure will permit application of existing monetary policy tools [272]. DC/EP will make use of non-fungible tokens: Each coin will have an individual denomination and serial number [277].

To store DC/EP, users will hold digital wallets with digital ledgers, protected by cryptography and consensus protocols [278]. The DC/EP wallet currently undergoing trials is available as a smartphone application for individual users. It offers user-to-user payments by QR code; payments can also be initiated by tapping smartphones with another user [2].

11.3 What patent filings reveal about the digital yuan

As of early 2020, the PBoC has filed more than 80 patent applications related to digital currency. These patent applications may be viewed as falling into four categories: "digital currency management, circulation and interbank settlement; digital currency wallets; processing payments and deposits; and distributed ledger transactions and technology" [279]. On the whole, these patents suggest a system under very tight control by the central bank, more so than is consistent with the banking system in Western nations. They also place a strong emphasis on compatibility with existing banking infrastructure; for example, some PBoC patent applications describe technical mechanisms for users to make deposits with their existing banks and then exchange deposited money for DC/EP.

Alipay, a mobile payment platform established in China by Alibaba, has also filed several patent applications explicitly related to the DC/EP. These filings include interesting capabilities and architectural nuances relating to the financial institutions managing the second tier of the DC/EP system [274], and are not discussed in previous treatments of DC/EP of which we're aware, e.g., [279].

Initially, the PBoC was interested in embracing innovative financial tools, particularly smart contracts and a distributed ledger, as expressed by the former head of the Digital Currency Research Institute, Yao Qian [280]. However, according to Fan Yifei, the Deputy Governor of the PBoC, that interest is tempered by concerns about undermining its cash-like status by supporting smart contracts [281]. Given this tension, the portfolio of patent applications we review here should be viewed as a spectrum of technologies that the PBoC may someday choose to develop rather than as a preview of the DC/EP at its release.

Rather than providing a comprehensive survey, we highlight some of the most interesting and salient features of the relevant PBoC and Alipay patent filings.

Anonymity: PBoC patent applications support a system design in which person-to-person or person-to-business transfers can be anonymous at the user level. Commercial banks in the first layer, however, would collect identifying information about

transacting parties that the central bank could also access. This tiered anonymity was referred to as "controllable anonymity," by Mu Changchun, the head of the PBoC Institute for Digital Currency. Users enjoy some degree of anonymity with respect to other users; banks, however, have a mechanism to deanonymize suspicious transactions, in order to combat money laundering and the financing of terrorism [282]. A patent application filed by Alipay, which uses the same phrase, "controllable anonymity," describes a mechanism for anonymity in user-to-user transactions, but does not provide any support for anonymity for the user from her financial institution. User-to-user anonymity could possibly extend to transactions between users in different banks [283].

Two other patent applications [284], [285] filed by PBoC in 2017 describe a cryptographic scheme—similar to Greg Maxwell's confidential transactions [223]—that hides the amount of currency in individual accounts as well as the amount of currency transferred in a given transaction from all parties but the participants in a transaction.

None of these patent applications, however, describes technology to hide transaction graphs, i.e., the pseudonyms of participants in transactions are recorded on the ledger. This means that the ledger *must be private to the authorities managing the system*. Otherwise users can deanonymize other users using the information revealed in the transaction graphs, as discussed in section 6.1.

Account control: A recent patent application filed by Alipay describes a command-and-control architecture in which regulators can directly, instantaneously, and unilaterally freeze users' funds. This account control could change the type of account belonging to a particular user, stop the flow of money in or out of a particular account, or freeze part or all of the DC/EP in the account [286]. As envisaged in this patent application, accounts are categorized in four levels, with the level of the account determined based on the amount and anticipated kinds of use, as well as the type of identifying information a user provides to open the account. Higher-level accounts offer more flexibility to their holders. For example, a user can apply for either an "anonymous" account or an account associated with his or her real name. "Anonymous" accounts (which in fact require some identifying information, like an email address or phone number, upon registration) are "low-level" in the sense that they provide only minimal functionality [277], such as strict balance limits.

Novel tools: PBoC patent propose technology to adjust the supply of DC/EP using an algorithm tracking certain triggers, like loan interest rates [287]. They also lay the groundwork for digital currency smartcards and digital wallets that a user can link directly to conventional bank accounts [288]. Additionally, the PBoC filed a patent application which specifies means for the central bank to activate tokens it distributes to banks with designated interest rates based on market conditions; these interest rates functionally tag coins with repayment conditions when they are used in loans [280].

Uses of secure hardware: As discussed in section 8, the role of secure hardware in digital currencies is a controversial and often misunderstood topic. However, recent patent applications filed by Alipay reveal a potential interest in embracing secure hardware, in particular Trusted Execution Environments (TEEs), in critical operations such as currency issuance. [289], [290] describes a TEE-based implementation of the two-layer issuance architecture. The central bank may deploy a "front-end encryption machine" (FEM), potentially realized by a TEE, at second-layer operators. The TEE functions in effect as a delegate of the central bank. The FEM stores the central bank's secret keys and is invoked by operators to issue and exchange digital currencies. To provide more detail, the second-layer issuance works as follows. First, an operator deposits a 100% reserve at the central bank, in exchange for a receipt in the form of an "encrypted string." The operator then feeds the FEM with the receipt and receives digital currency tokens of equivalent value. Another use of the FEM is to split a digital currency token of a large value into multiple ones with smaller denominations.

More commonplace applications of TEEs are also mentioned in these patent applications. One patent application [221] describes a digital wallet design that uses TEEs to protect users' private keys and perform *offline transactions* when the sender and/or receiver is not connected to the ledger. The patent implicitly assumes a perfectly secure TEE, however, and does not address the possibility of TEE compromise or failure. For example, since private keys are only accessible to the TEE, availability failures (e.g., permanent malfunctions) could lead to loss of funds. TEE can support flexible access control policies (e.g., [155]) capable of remedying such problems, but the patent filing does not consider this important direction.

12 Summary Position

Our explorations in this paper suggest a number of topics and issues that deserve special consideration by CBDC designers.

Monetary policy considerations: The issuance of CBDC will not in any way mask underlying weaknesses in central bank credibility or other issues such as fiscal dominance that affect the value of cash. In other words, digital central bank money is only as strong and credible as the central bank that issues it. In considering a shift to digital forms of retail central bank money, it is important to keep in mind that the transitional risks could be higher in the absence of stable macroeconomic and structural policies, including sound regulatory frameworks that are agile enough to be able to recognize and deal with financial risks created by new types of financial intermediaries.

It should also be recognized, notwithstanding the potential benefits, there are many unanswered questions about how the new financial technologies could affect the structure of financial institutions and markets. Questions also abound about whether retail CBDC will in any significant way affect monetary policy implementation and transmission. These uncertainties suggest a cautious approach to embracing

the concept of CBDC.

Ledger infrastructure: We discuss a range of architectural options for the digital ledger underpinning a CBDC. Our expectation is that central banks will wish to retain tight control over currency issuance and transaction processing, including the ability to alter or reverse transactions.

Such control is especially important given the historically demonstrated risks of catastrophic error in ledger-based systems, and existing legal requirements for handling error and fraud. In principle, tight ledger control is possible for any type of ledger, even public ("permissionless") ones, as central bank privileges can be hard-wired in, but in practice permissioned ledger systems, i.e., those limited to pre-designated entities, or centralized systems are more suitable choices. As multi-entity permissioned systems have not seen extensive deployment yet—their planned use in Libra constituting a significant technical experiment—our expectation is that central banks will opt for centralized CBDCs, and indeed the digital yuan appears to be embracing such an approach.

Central banks may wish to consider use of *authenticated data structures* (ADSs) as an extension for centralized CBDC deployments. An ADS may be thought of as a highly compressed version ("digest") R of the ledger at a given time. The central bank can distribute this digest R publicly without revealing ledger contents. It can prove the inclusion of particular transactions in the ledger with reference to R alone. It can also use R to demonstrate that it is not "forking," that is, showing different ledger contents / balances to different entities. Forking could occur as a result of operator malfeasance or a breach, so the ability to prove that forking has not taken place can strengthen users' confidence in the system.

Wallets and funds / key custody: One of the major challenges in successful democratization of cryptocurrency has arisen around the usability of wallets, and in particular the problem of *key management*. To authenticate users' transactions, that is, create strong evidence that they are submitted legitimately by holders of the relevant funds, it is necessary to *digitally sign* them. Digital signatures are a powerful cryptographic tool used in all modern computing infrastructure, but require the use of a *secret key*. Cryptocurrency users have found protecting and backing up keys to be unduly burdensome, and the result has been a heavy reliance on service providers that hold users' assets and act effectively like financial intermediaries. While CBDCs are likely to rely primarily on financial intermediaries[17], it is unclear how CBDCs can significantly advance the explicitly stated goal for CBDCs of financial inclusion should consumers need to engage with financial institutions.[18] Workable approaches to custody of funds and/or secret keys will be of pivotal importance in a CBDC.

[17]It appears that some designs, such as the digital yuan, may offer limited support for user-administered accounts.

[18]CBDCs would, however, make it easy to prepopulate individual accounts with funds, which would be an important first step in enrolling consumers in the financial system.

Privacy: Should a CBDC maintain the account balances of individuals on the ledger, which would seem to be a prerequisite for a retail CBDC, then *privacy* will become an issue of major importance. (The same is true for alternative representations of value, such as digital banknotes.) While there are cryptographic systems for maintaining transactional privacy in such settings, they are complex and costly, and unlikely to scale to meet the requirements of a CBDC in the short-to-medium term. One critical observation is that *pseudonymous* accounts, i.e., accounts in which account holders' names are kept secret, *offer only weak privacy.* Under many circumstances, as the history of cryptocurrencies shows, it would be possible to *deanonymize* accounts. In a practical sense, therefore, a CBDC will *reveal significantly more information about individuals' transactions to central banks than existing systems do.* This observation strongly motivates considered technical and legal confidentiality protections for ledger contents.

Opportunities for innovation: On the positive side, we believe there is a rich range of opportunities for innovation in a CBDC beyond mere reduction of frictions in transaction processing. Some derive from the unprecedented transparency a CBDC would afford regulators, including an ability to obtain a panoramic yet fine-grained view of global spending in an economy. These opportunities would also include new monetary policy levers, such as the ability of central banks to institute negative nominal interest rates, create currency with time-limits or other spending conditions (e.g., required spending on durable goods) in order to create highly targeted monetary interventions in a national economy.

Opportunities for novel financial technologies may be best captured with CBDC support for smart contracts, which would offer a flexible means of defining policies. Smart contracts would also offer opportunities for the creation of new types of financial instruments; in cryptocurrency systems, they have led to the creation of instruments so novel (e.g., "flash loans") that they have no direct analogs in the existing financial system. Given the historically demonstrated hazards of smart contract bugs (e.g., The DAO), however, software assurance and oversight will be of paramount importance.

Secure hardware: We also believe that central banks should explore the use of secure hardware to strengthen elements of a CBDC system. While vulnerabilities have surfaced in recently produced secure hardware, suggesting that it should not be used in mission-critical subsystems, there are a number of places in which it can serve as an adjunct to strengthen or harden systems in which it is deployed. We describe several such opportunities, such as improving privacy through defense-in-depth, i.e., as an added protective layer, improving compliance enforcement by constraining system use according to regulatory rules, and protecting the wallets of individual users.

Two-layer architectures: The publicly revealed plans or explorations of central banks to date focus on two-layer CBDC architectures. In such architectures, existing non-governmental financial institutions or payment application providers—dubbed

"Payment Interface Providers" (PIPs) in [10]—constitute a second layer on top of the CBDC, serving as the main interface between users and the CBDC. Two-layer architectures align closely with current customer service delivery models and compliance mechanisms for anti-money-laundering and countering the funding of terrorism (AML/CFT) laws, and would also appear to have the merit of avoiding disruptive disintermediation of the existing banking system.

It is important to note that a two-layer system would not remedy the privacy concerns associated with representation of individuals' accounts (or banknotes) in the CBDC. It could also introduce additional complications. For example, should smart contracts be deployed by by PIPs, rather than directly on the CBDC, systemic risks could escape the observation and control of regulators, and different PIPs' deployments could be mutually incompatible, creating patchwork interfaces to the CBDC. Conversely, tight control of smart contract environments by a central bank could stifle innovation. Establishment of basic technical and operating standards by the central bank could prove fruitful middle ground.

In summary, the benefits and risks of CBDC are complex, encompassing an interplay among financial, legal, and technical considerations. Each country will have to take into account its specific circumstances and initial conditions before deciding whether the potential benefits of introducing a CBDC outweigh the possible costs.

Acknowledgments

The authors wish to thank IC3's industry partners for their support of this work. Thanks also to Jim Ballingall for helpful comments. Ittay Eyal was partially funded by ISF (1641/18) and BSF grants. Giulia Fanti was partially funded by NSF grant CIF-1705007 and ARO grant W911NF17-S-0002. Bryan Ford was partially funded by ONR grant N00014-19-1-2361 and the AXA Research Fund. Ari Juels was partially funded by NSF grants CNS-1704615 and CNS-1933655. Sarah Meiklejohn was partially funded by EPSRC Grant EP/N028104/1. Andrew Miller was partially funded by NSF grant CNS-1943499.

References

[1] C. Boar, H. Holden, and A. Wadsworth, *Impending arrival – a sequel to the survey on central bank digital currency*, https://www.bis.org/publ/bppdf/bispap107.pdf, BIS Publication 107, 2020.

[2] W. Zhao, *Chinese state-owned bank offers test interface for pboc central bank digital currency*, Apr. 2020. [Online]. Available: https://www.coindesk.com/chinese-state-owned-bank-offers-test-interface-for-pboc-central-bank-digital-currency.

[3] V. Bharathan, "Digital dollar project in light of recent congressional hearings", *Forbes*, Jun 29, 2020.

[4] Y. Mersch, "An ECB digital currency – a flight of fancy?", Speech at Consensus 2020 virtual conference, 11 May 2020. [Online]. Available: https://www.ecb.europa.eu/press/key/date/2020/html/ecb.sp200511~01209cb324.en.html.

[5] Sveriges Riksbank, *The Riksbank to test technical solution for the e-krona*, Riksbank press release, 20 Feb. 2020.

[6] J. Light, B. Bain, and O. Kharif, "Facebook weighs Libra revamp to address regulatory concerns", *Bloomberg News*, 3 March 2020.

[7] V. Mislos, "CBDC 'not a reaction' to Libra despite study confirming consumer benefits of stablecoins", *International Business Times*, 26 June 2020.

[8] M. Ricks, J. Crawford, and L. Menand, "Central banking for all: A public option for bank accounts", *The Great Democracy Initiative Report*, June 2018.

[9] H. Jones, "Pandemic pushes central bank digital currencies into top gear", *Reuters Technology News*, 11 June 2020.

[10] "Central Bank Digital Currency: Opportunities, challenges and design", 12 March 2020. [Online]. Available: https://www.bankofengland.co.uk/paper/2020/central-bank-digital-currency-opportunities-challenges-and-design-discussion-paper.

[11] M. Kumhof and C. Noone, "Central bank digital currencies—design principles and balance sheet implications", Bank of England working paper No. 725, 2018.

[12] Q. Yao, "A systematic framework to understand central bank digital currency", *Science China Information Sciences*, vol. 61, no. 3. Article 033101, 2018.

[13] O. Bjerg, "Designing new money-the policy trilemma of central bank digital currency", CBS Working Paper, June 2017.

[14] M. D. Bordo and A. T. Levin, "Central bank digital currency and the future of monetary policy", National Bureau of Economic Research Working Paper No. 23711, 2017.

[15] K. S. Rogoff, *The curse of cash: How large-denomination bills aid crime and tax evasion and constrain monetary policy*. Princeton University Press, 2017.

[16] B. Mishra and E. Prasad, "A simple model of a central bank digital currency", Manuscript, Cornell University, 2019.

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

[18] V. Buterin, *A next generation smart contract & decentralized application platform*, 2013. [Online]. Available: https://www.ethereum.org/%20pdfs/EthereumWhitePaper.pdf/.

[19] M. Lokhava, G. Losa, D. Mazières, G. Hoare, N. Barry, E. Gafni, J. Jove, R. Malinowsky, and J. McCaleb, "Fast and secure global payments with stellar", in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 2019, pp. 80–96.

[20] *Libra white paper: Blockchain, association, reserve*, Apr. 2020. [Online]. Available: https://libra.org/en-US/white-paper/.

[21] Amazon AWS, *Amazon quantum ledger database*. [Online]. Available: https://aws.amazon.com/qldb/.

[22] L. Lamport, "Time, clocks, and the ordering of events in a distributed system", in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 179–196.

[23] ——, "The part-time parliament", *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[24] ——, "Fast Paxos", *Distributed Computing*, vol. 19, no. 2, pp. 79–103, 2006.

[25] M. Castro and B. Liskov, "Practical byzantine fault tolerance", in *OSDI*, vol. 99, 1999, pp. 173–186.

[26] I. Abraham, G. Chockler, I. Keidar, and D. Malkhi, "Byzantine disk Paxos: Optimal resilience with Byzantine shared memory", *Distributed Computing*, vol. 18, no. 5, pp. 387–408, 2006.

[27] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm", in *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, 2014, pp. 305–319.

[28] M. Yin, D. Malkhi, M. K. Reiter, G. Golan-Gueta, and I. Abraham, "Hotstuff: BFT consensus with linearity and responsiveness", in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019*, P. Robinson and F. Ellen, Eds., ACM, 2019, pp. 347–356. [Online]. Available: https://doi.org/10.1145/3293611.3331591.

[29] M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro, "Analysis of operating system diversity for intrusion tolerance", *Software: Practice and Experience*, vol. 44, no. 6, pp. 735–770, 2014.

[30] L. Breidenbach, P. Daian, F. Tramer, and A. Juels, "Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts", in *27th USENIX Security Symposium*, Aug. 2018.

[31] R. Tamassia, "Authenticated data structures", in *Proceedings of the 11th Annual European Symposium on Algorithms*, vol. 2832, Springer, 2003, pp. 2–5.

[32] S. Crosby and D. Wallach, "Efficient data structures for tamper-evident logging", in *Proceedings of the 18th USENIX Security Symposium*, 2009.

[33] A. Eijdenberg, B. Laurie, and A. Cutter, *Verifiable data structures*, github.com/google/trillian/blob/master/docs/VerifiableDataStructures.pdf, 2015.

[34] L. Reyzin, D. Meshkov, A. Chepurnoy, and S. Ivanov, "Improving authenticated dynamic dictionaries, with applications to cryptocurrencies", in *Proceedings of Financial Cryptography and Data Security*, 2017.

[35] L. Chuat, P. Szalachowski, A. Perrig, B. Laurie, and E. Messeri, "Efficient gossip protocols for verifying the consistency of certificate logs", in *Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS)*, 2015.

[36] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman, "CONIKS: Bringing key transparency to end users", in *Proceedings of the 24th USENIX Security Symposium*, 2015.

[37] A. Tomescu and S. Devadas, "Catena: Efficient non-equivocation via Bitcoin", in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2017.

[38] M. Al-Bassam and S. Meiklejohn, "Contour: A practical system for binary transparency", in *Proceedings of the 2nd International Workshop on Cryptocurrencies and Blockchain Technology (CBT)*, 2018.

[39] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford, "Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning", in *37th IEEE Symposium on Security and Privacy*, May 2016.

[40] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking Bitcoin: Large-scale Network Attacks on Cryptocurrencies", *38th IEEE Symposium on Security and Privacy*, May 2017.

[41] B. Ford, "Apple, FBI, and Software Transparency", *Freedom to Tinker*, Mar. 2016.

[42] P. Bright, *Independent Iranian hacker claims responsibility for Comodo hack*, Mar. 2011. [Online]. Available: www.wired.com/2011/03/comodo_hack/.

[43] J. Menn, *Key Internet operator VeriSign hit by hackers*, Feb. 2012. [Online]. Available: www.reuters.com/article/2012/02/02/us-hacking-verisign-idUSTRE8110Z820120202.

[44] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies", in *Proceedings of NDSS*, 2016.

[45] J. Kwon, "Tendermint: Consensus without mining", *Draft v. 0.6, fall*, vol. 1, no. 11, 2014.

[46] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing", in *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[47] M. Baudet, A. Ching, A. Chursin, G. Danezis, F. Garillot, Z. Li, D. Malkhi, O. Naor, D. Perelman, and A. Sonnino, "State machine replication in the Libra blockchain", *The Libra Assn., Tech. Rep*, 2019.

[48] G. Andresen, *March 2013 chain fork post-mortem*. [Online]. Available: https://en.bitcoin.it/wiki/BIP_0050.

[49] B. Community, *2015 BIP66 blockchain fork*. [Online]. Available: https://en.bitcoin.it/wiki/Softfork#2015_BIP66_Blockchain_Fork.

[50] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability", pp. 566–583, 2020.

[51] Y. Doweck and I. Eyal, "Multi-party timed commitments", *arXiv preprint arXiv:2005.04883*, 2020. [Online]. Available: https://arxiv.org/abs/2005.04883v2.

[52] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail", in *Annual International Cryptology Conference*, Springer, 1992, pp. 139–147.

[53] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols", in *Secure information networks*, Springer, 1999, pp. 258–272.

[54] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable", in *Financial Cryptography and Data Security*, 2014.

[55] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in Bitcoin", in *Financial Cryptography and Data Security*, 2016.

[56] R. Pass and E. Shi, "Fruitchains: A fair blockchain", in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2017, pp. 315–324.

[57] C. Hou, M. Zhou, Y. Ji, P. Daian, F. Tramer, G. Fanti, and A. Juels, "Squirrl: Automating attack discovery on blockchain incentive mechanisms with deep reinforcement learning", *arXiv preprint arXiv:1912.01798*, 2019.

[58] R. B. Zur, I. Eyal, and A. Tamar, "Efficient MDP analysis for selfish-mining in blockchains", *arXiv preprint arXiv:2007.05614*, 2020. [Online]. Available: https://arxiv.org/abs/2007.05614.

[59] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, and A. Juels, "BDoS: Blockchain denial of service", in *Proceedings of the 2020 ACM SIGSAC conference on Computer and Communications Security*, 2020.

[60] A. Miller, E. Shi, A. Juels, B. Parno, and J. Katz, "Permacoin: Repurposing Bitcoin work for data preservation", in *Proceedings of the IEEE Symposium on Security and Privacy*, San Jose, CA, USA: IEEE, 2014. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=217984.

[61] F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. Van Renesse, "REM: Resource-efficient mining for blockchains", in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1427–1444.

[62] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol", in *Annual International Cryptology Conference*, Springer, 2017, pp. 357–388.

[63] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine agreements for cryptocurrencies", in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 51–68.

[64] I. Tsabary, A. Spiegelman, and I. Eyal, "Heb: Hybrid expenditure blockchain", *arXiv*, arXiv–1911, 2019.

[65] B. Ford and R. Böhme, *Rationality is Self-Defeating in Permissionless Systems*, Sep. 2019.

[66] J. Kroll, I. Davey, and E. Felten, "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries", in *Workshop on Economics and Information Security (WEIS)*, Washington, DC, 2013.

[67] J. Becker, D. Breuker, T. Heide, J. Holler, H. P. Rauer, and R. Böhme, "Can we afford integrity by proof-of-work? Scenarios inspired by the Bitcoin currency", in *The Economics of Information Security and Privacy*, R. Böhme, Ed., Springer, 2013, pp. 135–156.

[68] J. Bonneau, "Hostile blockchain takeovers (short paper)", in *Financial Cryptography and Data Security Workshops*, A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, and M. Sala, Eds., ser. Lecture Notes in Computer Science, vol. 10958, Springer, 2018, pp. 92–100.

[69] A. Judmayer, N. Stifter, A. Zamyatin, I. Tsabary, I. Eyal, P. Gazi, S. Meiklejohn, and E. Weippl, *Pay-to-win: Incentive attacks on proof-of-work cryptocurrencies*, Cryptology ePrint Archive, Report 2019/775, 2019.

[70] E. Attah, *Five most prolific 51.1667em% attacks in crypto: Verge, Ethereum Classic, Bitcoin Gold, Feathercoin, Vertcoin*, Cryptoslate.com, https://tinyurl.com/yyrvxyoh, Apr. 2019.

[71] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-NG: A scalable blockchain protocol.", in *NSDI*, 2016.

[72] R. Pass and E. Shi, "Thunderella: Blockchains with optimistic instant confirmation", in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2018, pp. 3–33.

[73] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits", in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 585–602.

[74] H. Yu, I. Nikolic, R. Hou, and P. Saxena, "Ohie: Blockchain scaling made simple", in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 112–127.

[75] Y. Sompolinsky and A. Zohar, "PHANTOM and GHOSTDAG: A scalable generalization of nakamoto consensus", *IACR Cryptology ePrint Archive, Report 2018/104*, 2018. [Online]. Available: https://eprint.iacr.org/2018/104.

[76] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding", in *IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, pp. 19–34.

[77] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: A fast blockchain protocol via full sharding.", *IACR Cryptology ePrint Archive*, vol. 2018, p. 460, 2018.

[78] A. Manuskin, M. Mirkin, and I. Eyal, "Ostraka: Secure blockchain scaling by node sharding", in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&B)*, IEEE, 2020.

[79] P. Thompson, "Most significant hacks of 2019 — new record of twelve in one year", Jan. 2020. [Online]. Available: https://cointelegraph.com/news/most-significant-hacks-of-2019-new-record-of-twelve-in-one-year.

[80] SelfKey, *A comprehensive list of cryptocurrency exchange hacks*, Feb. 2020. [Online]. Available: https://selfkey.org/list-of-cryptocurrency-exchange-hacks/.

[81] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey", *Cryptography*, vol. 2, no. 1, Jan. 2018.

[82] S. Srinivas, D. Balfanz, E. Tiffany, A. Czeskis, and F. Alliance, "Universal 2nd factor (u2f) overview", *FIDO Alliance Proposed Standard*, pp. 1–5, 2015.

[83] N. Poh, C. H. Chan, J. Kittler, S. Marcel, C. M. Cool, E. A. Rúa, J. L. A. Castro, M. Villegas, R. Paredes, V. Štruc, N. Pavešić, A. A. Salah, H. Fang, and N. Costen, "An evaluation of video-to-video face verification", *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, Dec. 2010.

[84] R. Chesney and D. K. Citron, "Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security", *California Law Review*, Jul. 2018.

[85] D. Mack, "This PSA About Fake News From Barack Obama Is Not What It Appears", *Buzzfeed News*, Apr. 2018.

[86] T. C. King, N. Aggarwal, M. Taddeo, and L. Floridi, "Artificial intelligence crime: An interdisciplinary analysis of foreseeable threats and solutions", *Science and Engineering Ethics*, vol. 26, pp. 89–120, Feb. 2020.

[87] M. R. Anderson, "Twenty years on from Deep Blue vs Kasparov: How a chess match started the big data revolution", *The Conversation*, May 2017. [Online]. Available: https://theconversation.com/twenty-years-on-from-deep-blue-vs-kasparov-how-a-chess-match-started-the-big-data-revolution-76882.

[88] J. Markoff, "Computer wins on 'Jeopardy!': Trivial, it's not", *The New York Times*, Feb. 2011. [Online]. Available: https://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html.

[89] S. Borowiec, "Alphago seals 4-1 victory over Go grandmaster Lee Sedol", *The Guardian*, Mar. 2016. [Online]. Available: https://www.theguardian.com/technology/2016/mar/15/googles-alphago-seals-4-1-victory-over-grandmaster-lee-sedol.

[90] F.-Y. Wang, J. J. Zhang, X. Zheng, X. Wang, Y. Yuan, X. Dai, J. Zhang, and L. Yang, "Where does AlphaGo go: From Church-Turing thesis to AlphaGo thesis and beyond", *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 2, Apr. 2016.

[91] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen, and Z. Wang, "Yet another text Captcha solver: A generative adversarial network based approach", in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Jan. 2018, pp. 332–348.

[92] J. Dzieza, "Why CAPTCHAs have gotten so difficult", *The Verge*, Feb. 2019. [Online]. Available: https://www.theverge.com/2019/2/1/18205610/google-captcha-ai-robot-human-difficult-artificial-intelligence.

[93] M. Read, "How Much of the Internet Is Fake? Turns Out, a Lot of It, Actually.", *New York Magazine*, Dec. 2018.

[94] A. Berger, *Bot vs. Bot: Will the Internet Soon Be a Place Without Humans?*, Singularity Hub, Jul. 2018.

[95] M. Latah, *The art of social bots: A review and a refined taxonomy*, May 2019. [Online]. Available: https://arxiv.org/pdf/1905.03240.pdf.

[96] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: using hard AI problems for security", in *Eurocrypt*, 2003.

[97] C. Doctorow, "Solving and creating CAPTCHAs with free porn", *Boing Boing*, Jan. 2004. [Online]. Available: http://www.boingboing.net/2004/01/27/solving_and_creating.html.

[98] L. Kang and J. Xiang, "CAPTCHA phishing: A practical attack on human interaction proofing", in *Information Security and Cryptology (Inscrypt)*, Dec. 2009.

[99] B. Krebs, *Virtual sweatshops defeat bot-or-not tests*, Krebs on Security, Jan. 2012. [Online]. Available: https://krebsonsecurity.com/2012/01/virtual-sweatshops-defeat-bot-or-not-tests/.

[100] R. Brandom, "This is why you shouldn't use texts for two-factor authentication", *The Verge*, Sep. 2017. [Online]. Available: https://www.theverge.com/2017/9/18/16328172/sms-two-factor-authentication-hack-password-bitcoin.

[101] G. Tu, C. Li, C. Peng, Y. Li, and S. Lu, "New security threats caused by IMS-based SMS service in 4G LTE networks", in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Oct. 2016, pp. 1118–1130.

[102] A. Bhatia and J. Bhabha, "India's Aadhaar scheme and the promise of inclusive social protection", *Oxford Development Studies*, vol. 45, no. 1, pp. 64–79, Jan. 2017.

[103] B. Chaudhuri and L. König, "The Aadhaar scheme: a cornerstone of a new citizenship regime in India?", *Contemporary South Asia*, vol. 26, no. 2, pp. 127–142, Sep. 2017.

[104] R. Abraham, E. S. Bennett, R. Bhusal, S. Dubey, Q. (Li, A. Pattanayak, and N. B. Shah, "State of Aadhaar Report 2017-18", IDinsight, Tech. Rep., May 2018.

[105] B. Schneier, "Tigers use scent, birds use calls – biometrics are just animal instinct", *The Guardian*, Jan. 2009.

[106] A. Chanthadavong, "Biometrics: The password you cannot change", *ZDNet*, Aug. 2015.

[107] S. Venugopalan and M. Savvides, "How to generate spoofed irises from an iris code template", *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, Jun. 2011.

[108] Q. Zhao, A. K. Jain, N. G. Paulter, and M. Taylor, "Fingerprint image synthesis based on statistical feature models", in *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Arlington, VA, USA, Sep. 2012.

[109] N. Brandenberg, S. Hoehnel, F. Kuttler, K. Homicsko, C. Ceroni, T. Ringel, N. Gjorevski, G. Schwank, G. Coukos, G. Turcatti, and M. P. Lutolf, "High-throughput automated organoid culture via stem-cell aggregation in microcavity arrays", *Nature Biomedical Engineering*, Jun. 2020.

[110] P. Dixon, "A Failure to "Do No Harm" – India's Aadhaar biometric ID program and its inability to protect privacy in relation to measures in Europe and the U.S.", *Health and Technology*, vol. 7, no. 4, pp. 539–567, Dec. 2017.

[111] J. Srinivasan, S. Bailur, E. Schoemaker, and S. Seshagiri, "The Poverty of Privacy: Understanding Privacy Trade-Offs From Identity Infrastructure Users in India", *International Journal of Communication*, vol. 12, pp. 1228–1247, Mar. 2018.

[112] M. Gomez-Barrero and J. Galbally, "Reversing the irreversible: A survey on inverse biometrics", *Computers & Security*, vol. 90, Mar. 2020.

[113] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data", in *International conference on the theory and applications of cryptographic techniques*, Springer, 2004, pp. 523–540.

[114] R. Chatterjee, M. S. Riazi, T. Chowdhury, E. Marasco, F. Koushanfar, and A. Juels, "Multisketches: Practical secure sketches using off-the-shelf biometric matching algorithms", in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1171–1186.

[115] R. Aggarwal, J. W. Goodell, and L. J. Selleck, "Lending to women in microfinance: Role of social trust", *International Business Review*, vol. 24, no. 1, pp. 55–65, Feb. 2015.

[116] E. Hughes, *A cypherpunk's manifesto*, Mar. 1993. [Online]. Available: https://www.activism.net/cypherpunk/manifesto.html.

[117] J. Assange and J. Appelbaum, *Cypherpunks: Freedom and the Future of the Internet*. OR Books, Oct. 2016, ISBN: 978-1944869083.

[118] W. Stallings, "The PGP Web of Trust", *BYTE Magazine*, vol. 20, no. 2, pp. 161–164, Feb. 1995.

[119] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA, USA: MIT Press, 1995, ISBN: 0-262-74017-6.

[120] R. Rivest and B. Lampson, *SDSI: A Simple Distributed Security Infrastructure*, Apr. 1996.

[121] C. Ellison *et al.*, *SPKI Certificate Theory*, RFC 2693, Sep. 1999.

[122] A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0.", in *USENIX Security Symposium*, vol. 348, 1999, pp. 169–184.

[123] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: Somebody you know", in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 168–178.

[124] A. Mislove, A. Post, P. Druschel, and K. P. Gummadi, "Ostra: Leveraging trust to thwart unwanted communication", in *5th USENIX Symposium on Networked Systems Design and Implementation*, Apr. 2008, pp. 15–30.

[125] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks", in *29th IEEE Symposium on Security and Privacy*, Oakland, CA, May 2008. [Online]. Available: https://www.iscs.nus.edu.sg/~yuhf/sybillimit-tr.pdf.

[126] N. Tran, B. Min, J. Li, and L. Submaranian, "Sybil-resilient online content voting", in *6th Symposium on Networked System Design and Implementation (NSDI)*, Apr. 2009, pp. 15–28.

[127] B. Viswanath, M. Mondal, K. P. Gummadi, A. Mislove, and A. Post, "Canal: Scaling social network-based sybil tolerance schemes", in *EuroSys Workshop on Social Network Systems (SNS)*, Apr. 2012.

[128] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks", in *Internet Measurement Conference (IMC)*, San Diego, USA, Oct. 2007.

[129] B. Viswanath and A. Post, "An Analysis of Social Network-Based Sybil Defenses", in *ACM SIGCOMM*, New Delhi, India, Aug. 2010.

[130] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi, "Understanding and Combating Link Farming in the Twitter Social Network", in *21st International Conference on World Wide Web (WWW)*, Lyon, France, Apr. 2012.

[131] J. Messias, L. Schmidt, R. Oliveira, and F. Benevenuto, "You followed my bot! Transforming robots into influential users in Twitter", vol. 18, no. 7, Jul. 2013.

[132] C. A. Freitas, F. Benevenuto, S. Ghosh, and A. Veloso, "Reverse Engineering Socialbot Infiltration Strategies in Twitter", in *Advances in Social Networks Analysis and Mining (ASONAM)*, Paris, France, Aug. 2015, pp. 25–32.

[133] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The Rise of Social Bots", *Communications of the ACM*, vol. 59, no. 7, Jul. 2016.

[134] A. Bessi and E. Ferrara, "Social bots distort the 2016 U.S. Presidential election online discussion", *First Monday*, vol. 21, no. 11, Nov. 2016.

[135] D. A. Broniatowski, A. M. Jamison, S. Qi, L. AlKulaib, T. Chen, A. Benton, and S. C. Q. M. Dredze, "Weaponized Health Communication: Twitter Bots and Russian Trolls Amplify the Vaccine Debate", *American Journal of Public Health*, Sep. 2018.

[136] C. Allen, *The path to self-sovereign identity*, Apr. 2016. [Online]. Available: `http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html`.

[137] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity", *Computer Science Review*, vol. 30, pp. 80–86, Nov. 2018.

[138] J. S. Martin Schanzenbach Georg Bramm, "reclaimID: Secure, Self-Sovereign Identities using Name Systems and Attribute-Based Encryption", in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, New York, NY, USA, Aug. 2018.

[139] Q. Stokkink and J. Pouwelse, *Deployment of a blockchain-based self-sovereign identity*, Aug. 2018.

[140] A. Abraham, *Self-sovereign identity*, Oct. 2017. [Online]. Available: `http://www.egiz.gv.at/files/download/Self-Sovereign-Identity-Whitepaper.pdf`.

[141] A. Satybaldy, M. Nowostawski, and J. Ellingsen, *Self-sovereign identity systems: Evaluation framework*, Apr. 2020. [Online]. Available: https://www.researchgate.net/publication/339836401_Self-Sovereign_Identity_Systems_Evaluation_Framework.

[142] Decentralized Identity Foundation, *DIF website*, https://identity.foundation/, 2020.

[143] W3C, *Peer DID method specification*, https://openssi.github.io/peer-did-method-spec/index.html#privacy-considerations, 2020.

[144] O. Kharif, "Cryptokitties mania overwhelms ethereum network's processing", *Bloomberg (4 Dec. 2017)*, 2017.

[145] G. Fenu, L. Marchesi, M. Marchesi, and R. Tonelli, "The ICO phenomenon and its relationships with Ethereum smart contract environment", in *2018 IEEE 1st International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Mar. 2018.

[146] D. A. Zetzsche, R. P. Buckley, D. W. Arner, and L. Föhr, "The ICO gold rush: It's a scam, it's a bubble, it's a super challenge for regulators", *Harvard International Law Journal*, vol. 60, no. 2, 2019.

[147] Wikipedia contributors, *Multi-factor authentication — Wikipedia, the free encyclopedia*, [Online; accessed July 2020], 2020. [Online]. Available: https://en.wikipedia.org/wiki/Multi-factor_authentication.

[148] E. Cecchetti, F. Zhang, Y. Ji, A. Kosba, A. Juels, and E. Shi, "Solidus: Confidential distributed ledger transactions via PVORM", in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 701–717.

[149] J. J. J. Roberts and N. Rapp, "Nearly 4 million Bitcoins lost forever, new study says", *Fortune*, 25 Nov. 2017.

[150] B. Armstrong, *Coinbase is not a wallet*, 25 Feb. 2016.

[151] F. Wu, "No easy answers in the fight over iPhone decryption", *Communications of the ACM*, vol. 59, no. 9, pp. 20–22, 2016.

[152] R. Gennaro and S. Goldfeder, "Fast multiparty threshold ECDSA with fast trustless setup", in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1179–1194.

[153] V. Shoup, "Practical threshold signatures", in *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2000, pp. 207–220.

[154] M. Möser, I. Eyal, and E. G. Sirer, "Bitcoin covenants", in *International Conference on Financial Cryptography and Data Security*, Springer, 2016, pp. 126–141.

[155] F. Zhang, P. Daian, I. Bentov, I. Miers, and A. Juels, "Paralysis proofs: Secure dynamic access structures for cryptocurrency custody and more", in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, ser. AFT '19, Zurich, Switzerland: Association for Computing Machinery, 2019, pp. 1–15. [Online]. Available: https://doi.org/10.1145/3318041.3355459.

[156] D. Akhawe and A. P. Felt, "Alice in Warningland: A large-scale field study of browser security warning effectiveness", in *22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 2013, pp. 257–272.

[157] E. Almutairi and S. Al-Megren, "Usability and security analysis of the KeepKey wallet", in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2019, pp. 149–153.

[158] I. Sakharova, "Payment card fraud: Challenges and solutions", in *2012 IEEE international conference on intelligence and security informatics*, IEEE, 2012, pp. 227–234.

[159] Ulrich Bindseil, *Tiered CBDC and the financial system*, https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2351~c8c18bbd60.en.pdf, 2020.

[160] *Chainalysis*, Referenced July 2020. [Online]. Available: chainalysis.com.

[161] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in Bitcoin", in *International Conference on Financial Cryptography and Data Security*, Springer, 2013, pp. 34–51.

[162] D. Ron and A. Shamir, "Quantitative analysis of the full Bitcoin transaction graph", in *Proceedings of the 17th International Conference on Financial Cryptography & Data Security*, 2013, pp. 6–24.

[163] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of Bitcoins: Characterizing payments among men with no names", in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 127–140.

[164] M. Spagnuolo, F. Maggi, and S. Zanero, "BitIodine: Extracting intelligence from the Bitcoin network", in *Proceedings of the 18th International Conference on Financial Cryptography & Data Security*, 2014, pp. 457–468.

[165] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in Zcash", in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 463–477.

[166] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A traceability analysis of Monero's blockchain", in *ESORICS 2017*, 2017, pp. 153–173.

[167] M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin, "An empirical analysis of linkability in the Monero blockchain", *Proceedings on Privacy Enhancing Technologies*, pp. 143–163, 3 2018.

[168] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from Bitcoin", in *Security and Privacy (SP), 2014 IEEE Symposium on*, IEEE, 2014, pp. 459–474.

[169] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router", in *12th USENIX Security Symposium*, Aug. 2004.

[170] K. Hill, "How did the FBI break Tor?", *Forbes*, Nov. 2014.

[171] C. Farivar, "Judge confirms what many suspected: Feds hired cmu to break tor", *Ars Technica*, Feb. 2016.

[172] F. Tramèr, D. Boneh, and K. G. Paterson, "Remote side-channel attacks on anonymous transactions.", *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 220, 2020.

[173] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, N. Gailly, L. Gasser, I. Khoffi, J. Cappos, and B. Ford, "CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds", in *26th USENIX Security Symposium*, 2017, pp. 1271–1287.

[174] E. Kokoris-Kogias, "Secure, Confidential Blockchains Providing High Throughput and Low Latency", PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), May 2019.

[175] Z. Amsden, R. Arora, S. Bano, M. Baudet, S. Blackshear, A. Bothra, G. Cabrera, C. Catalini, K. Chalkias, E. Cheng, A. Ching, A. Chursin, G. Danezis, G. D. Giacomo, D. L. Dill, H. Ding, N. Doudchenko, V. Gao, Z. Gao, F. Garillot, M. Gorven, P. Hayes, J. M. Hou, Y. Hu, K. Hurley, K. Lewi, C. Li, Z. Li, D. Malkhi, S. Margulis, B. Maurer, P. Mohassel, L. de Naurois, V. Nikolaenko, T. Nowacki, O. Orlov, D. Perelman, A. Pott, B. Proctor, S. Qadeer, Rain, D. Russi, B. Schwab, S. Sezer, A. Sonnino, H. Venter, L. Wei, N. Wernerfelt, B. Williams, Q. Wu, X. Yan, T. Zakian, and R. Zhou, *The Libra blockchain*, May 2020. [Online]. Available: `https://developers.libra.org/docs/assets/papers/the-libra-blockchain/2020-05-26.pdf`.

[176] E. Kokoris-Kogias, E. C. Alp, S. D. Siby, N. Gailly, L. Gasser, P. Jovanovic, E. Syta, and B. Ford, *Verifiable Management of Private Data under Byzantine Failures*, Cryptology ePrint Archive, Report 2018/209, 2018.

[177] E. C. Alp, E. Kokoris-Kogias, G. Fragkouli, and B. Ford, "Rethinking General-Purpose Decentralized Computing", in *17th Workshop on Hot Topics in Operating Systems (HotOS XVII)*, Bertinoro, Italy, May 2019.

[178] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on Hyperledger Fabric with secure multiparty computation", *IBM Journal of Research and Development*, vol. 63, no. 2/3, pp. 3–1, 2019.

[179] H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts", in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1353–1370.

[180] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts", in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2019.

[181] S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra, and H. Wu, "Zexe: Enabling decentralized private computation", in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020.

[182] H. Berghel, "Equifax and the Latest Round of Identity Theft Roulette", *IEEE Computer*, vol. 50, no. 12, Dec. 2017.

[183] J. Feigenbaum, J. A. Hendler, A. D. Jaggard, D. J. Weitzner, and R. N. Wright, "Accountability and deterrence in online life", in *International Conference on Web Science (ICWS)*, 2011.

[184] A. Shamir, "How to Share a Secret", *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[185] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data.", in *USENIX Security Symposium*, 2009, pp. 299–316.

[186] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Balancing accountability and privacy using e-cash", in *International Conference on Security and Cryptography for Networks*, Springer, 2006, pp. 141–155.

[187] C. Garman, M. Green, and I. Miers, "Accountable privacy for decentralized anonymous payments", in *International Conference on Financial Cryptography and Data Security*, Springer, 2016, pp. 81–98.

[188] K. Wüst, K. Kostiainen, V. Čapkun, and S. Čapkun, "Prcash: Fast, private and regulated transactions for digital currencies", in *International Conference on Financial Cryptography and Data Security*, Springer, 2019, pp. 158–178.

[189] J. A. Kroll, E. W. Felten, and D. Boneh, *Secure protocols for accountable warrant execution*, Apr. 2014.

[190] A. Segal, B. Ford, and J. Feigenbaum, "Catching bandits and only bandits: Privacy-preserving intersection warrants for lawful surveillance", in *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI'14)*, Aug. 2014.

[191] A. Segal, J. Feigenbaum, and B. Ford, "Privacy-Preserving Lawful Contact Chaining", in *Workshop on Privacy in the Electronic Society (WPES)*, Oct. 2016.

[192] J. Feigenbaum, "Multiple Objectives of Lawful-Surveillance Protocols (Transcript of Discussion)", in *Cambridge International Workshop on Security Protocols*, Springer, 2017, pp. 9–17.

[193] J. Frankle, S. Park, D. Shaar, S. Goldwasser, and D. J. Weitzner, "Practical Accountability of Secret Processes", in *27th USENIX Security Symposium*, Aug. 2018.

[194] G. Panwar, R. Vishwanathan, S. Misra, and A. Bos, "Sampl: Scalable auditability of monitoring processes using public ledgers", in *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*, Nov. 2019.

[195] A. Miller, Z. Cai, and S. Jha, "Smart contracts and opportunities for formal methods", in *International Symposium on Leveraging Applications of Formal Methods*, Springer, 2018, pp. 280–299.

[196] E. Hildenbrandt, M. Saxena, N. Rodrigues, X. Zhu, P. Daian, D. Guth, B. Moore, D. Park, Y. Zhang, A. Stefanescu, *et al.*, "KEVM: A complete formal semantics of the Ethereum virtual machine", in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, IEEE, 2018, pp. 204–217.

[197] S. Blackshear, E. Cheng, D. L. Dill, V. Gao, B. Maurer, T. Nowacki, A. Pott, S. Qadeer, D. R. Rain, S. Sezer, *et al.*, *Move: A language with programmable resources*, 2019.

[198] K. Crary and M. J. Sullivan, "Peer-to-peer affine commitment using Bitcoin", in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2015, pp. 479–488.

[199] S. A. K. Thyagarajan, A. Bhat, B. Magri, D. Tschudi, and A. Kate, "Reparo: Publicly verifiable layer to repair blockchains", *arXiv preprint arXiv:2001.00486*, 2020.

[200] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain–or–rewriting history in bitcoin and friends", in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2017, pp. 111–126.

[201] Z. Liu, Y. Xiang, J. Shi, P. Gao, H. Wang, X. Xiao, B. Wen, and Y.-C. Hu, "Hyperservice: Interoperability and programmability across heterogeneous blockchains", in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 549–566.

[202] A. E. Gencer, R. van Renesse, and E. G. Sirer, "Service-oriented sharding with Aspen", *arXiv preprint arXiv:1611.06816*, 2016.

[203] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, *Enabling blockchain innovations with pegged sidechains*, 2014. [Online]. Available: https://blockstream.com/sidechains.pdf.

[204] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake", *self-published paper, August*, vol. 19, p. 1, 2012.

[205] *Intel Software Guard Extensions, Reference Number: 332680-002*, 2015. [Online]. Available: https://software.intel.com/sites/default/files/332680-002.pdf.

[206] T. Alves and D. Felton, *TrustZone: Integrated Hardware and Software Security-Enabling Trusted Computing in Embedded Systems*, 2004. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf.

[207] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors-a survey", *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, 2006.

[208] Bitcoin Wiki, *Hardware wallet*, 2020. [Online]. Available: https://en.Bitcoin.it/wiki/Hardware_wallet.

[209] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for cpu based attestation and sealing", in *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*, ACM New York, NY, USA, vol. 13, 2013, p. 7.

[210] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical", in *11th USENIX Workshop on Offensive Technologies, WOOT 2017*, USENIX, 2017.

[211] J. Götzfried, M. Eckert, S. Schinzel, and T. Müller, "Cache attacks on Intel SGX", in *Proceedings of the 10th European Workshop on Systems Security*, ACM, 2017, p. 2.

[212] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg, "Meltdown: Reading kernel memory from user space", in *27th USENIX Security Symposium (USENIX Security 18)*, 2018.

[213] J. Van Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx, "FORESHADOW: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution", in *Proceedings of the 27th USENIX Security Symposium. USENIX Association*, 2018.

[214] A. Rane, C. Lin, and M. Tiwari, "Raccoon: Closing digital side-channels through obfuscated execution", in *USENIX Security Symposium*, 2015.

[215] F. Brasser, S. Capkun, A. Dmitrienko, T. Frassetto, K. Kostiainen, and A.-R. Sadeghi, "DR.SGX: Automated and adjustable side-channel protection for SGX using data location randomization", in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19, New York, NY, USA: Association for Computing Machinery, 2019, pp. 788–800, ISBN: 9781450376280. [Online]. Available: https://doi.org/10.1145/3359789.3359809.

[216] D. Genkin, L. Pachmanov, I. Pipman, A. Shamir, and E. Tromer, "Physical key extraction attacks on pcs", *Communications of the ACM*, vol. 59, no. 6, 2016.

[217] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices", in *International Workshop on Security Protocols*, Springer, 1997, pp. 125–136.

[218] ——, "Tamper resistance-a cautionary note", in *Proceedings of the second Usenix workshop on electronic commerce*, vol. 2, 1996, pp. 1–11.

[219] O. Kömmerling and M. G. Kuhn, "Design principles for tamper-resistant smartcard processors.", *Smartcard*, vol. 99, pp. 9–20, 1999.

[220] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, "Keystone: An open framework for architecting trusted execution environments", in *Proceedings of the Fifteenth European Conference on Computer Systems*, ser. EuroSys '20, New York, NY, USA: Association for Computing Machinery, 2020, ISBN: 9781450368827. [Online]. Available: https://doi.org/10.1145/3342195.3387532.

[221] Q. Yao, Z. Xu, and Y. Zhang, *A kind of safety method, system and the terminal of digital cash of the use based on block chain*, Jan. 2017. [Online]. Available: https://patents.google.com/patent/CN106850200B/en.

[222] D. Chaum, "Blind signatures for untraceable payments", in *Advances in cryptology*, Springer, 1983, pp. 199–203.

[223] M. Greg, *Confidential transactions*, 2015. [Online]. Available: https://web.archive.org/web/20150630144253/https://people.xiph.org/~greg/confidential_values.txt.

[224] K. Wüst, S. Matetic, M. Schneider, I. Miers, K. Kostiainen, and S. Čapkun, "Zlite: Lightweight clients for shielded zcash transactions using trusted execution", in *International Conference on Financial Cryptography and Data Security*, Springer, 2019, pp. 179–198.

[225] S. Matetic, K. Wüst, M. Schneider, K. Kostiainen, G. Karame, and S. Capkun, "BITE: Bitcoin lightweight client privacy using trusted execution", in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 783–800.

[226] *Ledger hardware wallet*, https://www.ledger.com/, 2020.

[227] *Trezor hardware wallet*, https://trezor.io/, 2020.

[228] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts", in *2016 IEEE symposium on security and privacy (SP)*, 2016.

[229] S. Gesell, *The Natural Economic Order*. London: Peter Owen Limited, 1958, Translated by Philip Pye M.A.

[230] B. Champ, "Stamp Scrip: Money People Paid to Use", Jan. 2008. [Online]. Available: https://www.clevelandfed.org/newsroom-and-events/publications/economic-commentary/economic-commentary-archives/2008-economic-commentaries/ec-20080401-stamp-scrip-money-people-paid-to-use.aspx.

[231] D. Grant, *An overview of blockchain-based universal basic income projects*, Jul. 2018. [Online]. Available: https://www.usv.com/writing/2018/07/an-overview-of-blockchain-based-universal-basic-income-projects/.

[232] A. Brenzikofer, *Encointer – an ecological, egalitarian and private cryptocurrency and self-sovereign identity system*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1912.12141.

[233] A. Howitt, *Roadmap to a government-independent basic income (UBI) digital currency*, Feb. 2019. [Online]. Available: https://basicincome.org/wp-content/uploads/2020/03/UBI-ROADMAP-v1.2.1.pdf.

[234] B. Ford, *Democratic Value and Money for Decentralized Digital Society*, Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.12375.

[235] Goldman Sachs Group, Inc., *Quantinomics*, Referenced June 2020. [Online]. Available: https://www.gsam.com/content/gsam/global/en/market-insights/gsam-insights/quantinomics.html.

[236] I. Ayres, *Super crunchers: Why thinking-by-numbers is the new way to be smart*. Bantam Books, 2007.

[237] J. Dew, "The association between consumer debt and the likelihood of divorce", *Journal of Family and Economic Issues*, vol. 32, no. 4, pp. 554–565, 2011.

[238] D. Berger and J. Vavra, "Consumption dynamics during recessions", *Econometrica*, vol. 83, no. 1, pp. 101–154, 2015.

[239] N. De, "Story from news US Treasury Department blacklists 20 Bitcoin Addresses tied to alleged North Korean hackers", *Coindesk*, 2 March 2020.

[240] J. Barrdear and M. Kumhof, "The macroeconomics of central bank issued digital currencies", Bank of England working paper No. 605, 2016.

[241] D. Andolfatto, "Assessing the impact of central bank digital currency on private banks", FRB St. Louis Working Paper No. 2018-25, 2018.

[242] M. D. Bordo and A. T. Levin, "Digital cash: Principles & practical steps", National Bureau of Economic Research Working Paper No. 25455, 2019.

[243] D. Andolfatto, "Bitcoin and central banking", *MacroMania (blog)*, 2015. [Online]. Available: http://www.andolfatto.blogspot.com/2015/11/bitcoin-and-central-banking.html.

[244] K. Assenmacher and S. Krogstrup, *Monetary policy with negative interest rates: Decoupling cash from electronic money*. International Monetary Fund Working Paper No. 18/191, 2018.

[245] M. L. Bech and R. Garratt, "Central bank cryptocurrencies", *BIS Quarterly Review September*, pp. 55–70, 2017.

[246] C. on Payments and M. Infrastructures, *Digital currencies*, 2015.

[247] ——, *Central bank digital currencies*, 2018.

[248] B. Broadbent, "Central banks and digital currencies", Speech at Centre for Macroeconomics, London School of Economics, 2016. [Online]. Available: https://www.bis.org/review/r160303e.pdf.

[249] A. Carstens, "Money in the digital age: What role for central banks?", Lecture at the House of Finance, Goethe University, Frankfurt, 2018.

[250] W. Engert and B. S.-C. Fung, "Central bank digital currency: Motivations and implications", Bank of Canada Staff Discussion Paper No. 2017-16, 2017.

[251] B. S. Fung and H. Halaburda, "Central bank digital currencies: A framework for assessing why and how", Bank of Canada Staff Discussion Paper No. 2016-22, 2016.

[252] T. M. Griffoli, M. M. S. M. Peria, M. I. Agur, M. A. Ari, M. J. Kiff, M. A. Popescu, and M. C. Rochon, *Casting Light on Central Bank Digital Currencies*. International Monetary Fund Staff Discussion Note, 2018.

[253] A. Grym, P. Heikkinen, K. Kauko, K. Takala, *et al.*, "Central bank digital currency", *Bank of Finland Economics Review No. 5/2017*, 2017.

[254] S. Ingves, "Do we need an e-krona?", Speech at Swedish House of Finance, Stockholm, 2017. [Online]. Available: https://www.riksbank.se/en-gb/financial-stability/the-financial-system/payments/does-sweden-need-an-e-krona/.

[255] E. Prasad, "Central banking in a digital age: Stock-taking and preliminary thoughts", Brookings Institution Report, 2018.

[256] M. Raskin and D. Yermack, "Digital currencies, decentralized ledgers and the future of central banking", National Bureau of Economic Research Working Paper No. 22238, 2018.

[257] H. Rey, "Dilemma not trilemma: The global financial cycle and monetary policy independence", Proceedings of the Jackson Hole Symposium, Federal Reserve Bank of Kansas City, 2015.

[258] M. Tolle, "Central bank digital currency: The end of monetary policy as we know it?", *Bank Underground (blog), Bank of England*, 2016. [Online]. Available: https://bankunderground.co.uk/2016/07/25/central-bank-digital-currency-the-end-of-monetary-policy-as-we-know-it/.

[259] Q. Yao, "The application of digital currency in interbank cash transfer scenario", *Finance Comput.*, vol. 5, pp. 16–19, 2017.

[260] K. Qin, L. Zhou, B. Livshits, and A. Gervais, "Attacking the DeFi ecosystem with flash loans for fun and profit", *arXiv preprint arXiv:2003.03810*, 2020. [Online]. Available: https://arxiv.org/abs/2003.03810.

[261] *DeFi Pulse*, Referenced July 2020. [Online]. Available: defipulse.com.

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

[263] B. Marino and A. Juels, "Setting standards for altering and undoing smart contracts", in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, Springer, 2016, pp. 151–166.

[264] A. B. Laffer, W. H. Winegarden, and J. Childs, "The economic burden caused by tax code complexity", *The Laffer Center for Supply-Side Economics*, 2011.

[265] L. M. LoPucki and E. Warren, *Secured Transactions: A Systems Approach*. Wolters Kluwer, 2019.

[266] R. J. Mann, "Reliable perfection of security interests in crypto-currency", *SMU Sci. & Tech. L. Rev.*, vol. 21, p. 159, 2018.

[267] K. V. Tu, "Crypto-collateral", *SMU Sci. & Tech. L. Rev.*, vol. 21, p. 205, 2018.

[268] J. L. Schroeder, "Bitcoin and the uniform commercial code", *U. Miami Bus. L. Rev.*, vol. 24, p. 1, 2015.

[269] X.-T. Nguyen, "Lessons from case study of secured transactions with bitcoin", *SMU Sci. & Tech. L. Rev.*, vol. 21, p. 181, 2018.

[270] Internal Revenue Service (IRS), *Notice 2014-21*, 2014.

[271] Tether, *Tether: Digital money for a digital age*. [Online]. Available: tether.to, (accessed: 07.03.2020).

[272] Jinze and Etiene, *First look: China's central bank digital currency*, Aug. 2019. [Online]. Available: https://research.binance.com/analysis/china-cbdc.

[273] L. Baitao, *Central bank digital currency will become the biggest magic weapon for RMB internationalization*, Aug. 2019. [Online]. Available: https://finance.sina.com.cn/blockchain/coin/2019-08-12/doc-ihytcitm8648461.shtml.

[274] J. Martin, *Alipay patents reveal more details about China's forthcoming CBDC*, Mar. 2020. [Online]. Available: https://cointelegraph.com/news/alipay-patents-reveal-more-details-about-chinas-forthcoming-cbdc.

[275] H. Murphy and Y. Yang, *Patents reveal extent of China's digital currency plans*, Feb. 2020. [Online]. Available: https://www.ft.com/content/f10e94cc-4d74-11ea-95a0-43d18ec715f5.

[276] M. del Castillo, *Alibaba, Tencent, five others to receive first Chinese government cryptocurrency*, Aug. 2019. [Online]. Available: https://www.forbes.com/sites/michaeldelcastillo/2019/08/27/alibaba-tencent-five-others-to-recieve-first-chinese-government-cryptocurrency/#4202c28b1a51.

[277] X. Jing, *Method and device for opening digital currency wallet and electronic equipment*, Feb. 2020. [Online]. Available: https://patents.google.com/patent/CN110852729A/en.

[278] Y. Qian, *Blockchain and central bank digital currency*, Mar. 2020. [Online]. Available: https://www.ccvalue.cn/article/216773.html.

[279] P. Boring and M. Kaufman, *Blockchain: The breakthrough technology of the decade and how China is leading the way – an industry white paper*, Feb. 2020. [Online]. Available: https://digitalchamber.org/wp-content/uploads/dlm_uploads/2020/02/Blockchain-The-Breakthrough-Technology-of-the-Decade-and-How-china-is-Leading-the-Way.pdf.

[280] Y. Qian, "Experimental research on central bank digital currency prototype system", *Journal of Software*, vol. 29, no. 09, pp. 2716–2712, 2018.

[281] F. Yifei and L. Jiechen, *Several considerations about central bank digital currency*, Jan. 2018. [Online]. Available: https://www.yicai.com/news/5395409.html.

[282] J. Ossinger, *Pboc wants 'controllable anonymity' in China's digital currency*, Nov. 2019. [Online]. Available: https://www.bloomberg.com/news/articles/2019-11-13/pboc-wants-controllable-anonymity-in-china-s-digital-currency.

[283] Y. Wei, *Anonymous transaction method and system based on digital currency*, Mar. 2020. [Online]. Available: https://patents.google.com/patent/CN110889681A/en.

[284] Y. Qian, D. Gang, Q. Youcai, H. Lieming, C. Haibo, Z. Xinyu, W. Jiwei, and Z. Dawei, *A kind of method of commerce and device based on digital cash*, Oct. 2017. [Online]. Available: https://patents.google.com/patent/CN107358424A/en.

[285] ——, *Use the method for commerce and device of digital cash*, Nov. 2017. [Online]. Available: https://patents.google.com/patent/CN107392603A/en.

[286] Y. Wei, *Digital currency account control method and device*, Feb. 2020. [Online]. Available: https://patents.google.com/patent/CN110838061A/en.

[287] Q. Yao, *Digital cash management method and system based on the triggering of loan interest rate condition*, Aug. 2018. [Online]. Available: https://patents.google.com/patent/CN108416671A/en?oq=+CN108416671A.

[288] X. Jing, *Method and device for opening digital currency wallet and electronic equipment*, Feb. 2019. [Online]. Available: https://patents.google.com/patent/CN110852729A/en.

[289] Z. Meng, S. Xu, and H. Zhou, *Method and device for executing digital currency transaction and electronic equipment*, Oct. 2019. [Online]. Available: https://patents.google.com/patent/CN110827146A/en.

[290] Z. Meng, H. Yang, and H. Zhou, *Transaction processing method and device based on digital currency and electronic equipment*, Oct. 2019. [Online]. Available: https://patents.google.com/patent/CN110852730A/en.