Blockchain and Other Networks

BRIAN BEHLENDORF

The State of Hyperledger

By Swapnil Bhartiya

Linux Foundation | October 25, 2018
https://www.linuxfoundation.org/blog/2018/10/the-state-of-hyperledger-with-brian-behlendorf/

Hyperledger has grown in a way that mirrors the growth of the blockchain industry. "When we started, all the excitement was around bitcoin," said Brian Behlendorf, Executive Director of Hyperledger. Initially, it was more about moving money around. But the industry started to go beyond that and started to see if it "could be used as a way to reestablish how trust works on the Internet."

See also in the TTI/Vanguard archive:

- Brian Behlendorf: *Peer Production and Public Policy, Vancouver,* Canada, October 2010.
- Bill Schafer: *Enabling Sustainability with Blockchain Technology Accessed by Mobile Devices,* Boston, Massachusetts, April 2017.

BRAD CHASE

What Is Ripple and How Does it Work?

By Steve Fiorillo

The Street | July 10, 2018
https://www.thestreet.com/investing/what-is-ripple-14644949

The two biggest cryptocurrencies in the world by market cap are Bitcoin and Ethereum, but the third largest—Ripple—stands out from them. Ripple is at once a company, a digital-payment processing system and a cryptocurrency, which is also known as XRP. This is similar to bitcoin, but Ripple's blockchain system is very different, and the currency is owned by the one company—Ripple—whereas bitcoin is mined.

The Ripple Protocol Consensus Algorithm
By David Schwartz, Noah Youngs, and Arthur Britto

Ripple.com | 2018
https://ripple.com/files/ripple_consensus_whitepaper.pdf

While several consensus algorithms exist for the Byzantine Generals Problem, specifically as it pertains to distributed payment systems, many suffer from high latency induced by the requirement that all nodes within the network communicate synchronously. In this work, we present a novel consensus algorithm that circumvents this requirement by utilizing collectively-trusted subnetworks within the larger network. We show that the "trust" required of these subnetworks is in fact minimal and can be further reduced with principled choice of the member nodes. In addition, we show that minimal connectivity is required to maintain agreement throughout the whole network. The result is a low-latency consensus algorithm which still maintains robustness in the face of Byzantine failures. We present this algorithm in its embodiment in the Ripple Protocol.

See also in the TTI/Vanguard archive:

- John Henry Clippinger: *The Bitcoin Revolt: Self-Governance through Technology,* Atlanta, Georgia, February 2014.
- Yorke Rhodes: *Does Blockchain Change How We Think about Security?* Washington, D.C., September 2016.
- Simon Crosby: *Virtualization: Security's Silver Bullet: Lessons from Troy to Byzantium*, Washington, D.C., September 2016.
- Simon Crosby: *Reimagining Security for the Internet of Everything: Lessons from Troy to Byzantium*, Seattle, Washington, December 2012.

JOHN CLIPPINGER

Blockchain, Burning Man, and the Future of Governance
By Robert C. Wolcott

Forbes | Feb 16, 2017
https://www.forbes.com/sites/robertwolcott/2017/02/16/blockchain-burning-man-and-the-future-of-governance-a-conversation-with-john-clippinger/#54c24d601b0b

John Clippinger always seems to be ahead of trends. In 1965, he marched in Selma, Alabama in support of civil rights. In 2013 (more prosaically), Clippinger introduced me to blockchain. When others were just discovering this methodology underlying Bitcoin, he had already been exploring how blockchain might transform business and government.

See also in the TTI/Vanguard archive:

- John Henry Clippinger: *The Bitcoin Revolt: Self-Governance through Technology*, Atlanta, Georgia, February 2014.

DAHNA GOLDSTEIN

Why We've Created the Blockchain Impact Ledger

By Dahna Goldstein

Medium | April 15, 2019
https://medium.com/impact-ledger/why-weve-created-the-impact-ledger-86106d3affa9

The conversation about blockchain is evolving rapidly. A couple of years ago, the driving question was "what is blockchain?" Now, the questions are, "what are the best use cases for blockchain?" Or "what are some real-world examples of blockchain in action?" Social impact may not be the first application that comes to mind for many who think about blockchain, but it is perhaps one of the most promising.

See also in the TTI/Vanguard archive:

- Brewster Kahle: *Locking the Web Open: A Call for a Decentralized Web*, San Francisco, California, May 2016.

ERIC HASELTINE

Review: The Spy in Moscow Station

Kirkus Reviews | May 13th, 2019

https://www.kirkusreviews.com/book-reviews/eric-haseltine/the-spy-in-moscow-station/

Haseltine is a former director of research for the NSA—his boss there, Gen. Michael V. Hayden, contributes a foreword—and his expertise is beyond reproach. His research here is breathtaking, drawing on a bevy of sources, including his own interviews with Gandy as well as declassified U.S. governmental documents, often reproduced here at great length. In fact, his thoroughness can be a bit overwhelming at times; readers will often find themselves buried under mounds of minute detail, much of it forbiddingly technical. Even so, the story as a whole has all the power and intrigue of a cinematic thriller.

See also in the TTI/Vanguard archive:

- Eric Haseltine: *Can U.S. Elections Be Hacked?*, Washington, D.C., September 2018.

ANIKET KATE

Blockchain Access Privacy: Challenges and Directions

By Ryan Henry, Amir Herzberg, and Aniket Kate

IEEE Security & Privacy | July/August 2018
https://ieeexplore.ieee.org/document/8425613

Privacy, facilitated by a confluence of cryptography and decentralization, is one of the primary motivations for the adoption of cryptocurrencies like Bitcoin. Alas, Bitcoin's privacy promise has proven illusory and, despite growing interest in privacy-centric blockchains, most blockchain users remain susceptible to privacy attacks that exploit network-layer information and access patterns which leak as users interact with blockchains. Understanding if and how blockchain-based applications can provide strong privacy guarantees is a matter of increasing urgency. Many researchers

advocate using anonymous communications networks, e.g., Tor, to ensure access privacy. We challenge this approach, showing the need for mechanisms through which non-anonymous users can (i) publish transactions that cannot be linked to their network addresses or to their other transactions, and (ii) fetch details of specific transactions without revealing which transactions they seek. We hope this article inspires blockchain researchers to think 'beyond Tor' and tackle these important access privacy problems head-on.

See also in the TTI/Vanguard archive:

- Jacob Appelbaum: *Going Dark*, Washington, D.C., May 2012.
- Adam Ghetti: *Protecting Data, Not Networks,* Washington, D.C., September 2016.

ANNE KIM

Genie: A Secure, Transparent Sharing and Services Platform for Genetic and Health Data

By Shifa Zhang, Anne Kim, Dianbo Liu, Sandeep C. Nuckchady, Lauren Huang, Aditya Masurkar, Jingwei Zhang, Law Pratheek Karnati, Laura Martinez, Thomas Hardjono, Manolis Kellis, and Zhizhuo Zhang

arXiv | November 4, 2018
https://arxiv.org/abs/1811.01431

Artificial Intelligence incorporating genetic and medical information have been applied in disease risk prediction, unveiling disease mechanism, and advancing therapeutics. However, AI training relies on highly sensitive and private data which significantly limit their applications and robustness evaluation. Moreover, the data access management after sharing across organizations heavily relies on legal restriction, and there is no guarantee in preventing data leaking after sharing. Here, we present Genie, a secure AI platform that allows AI models to be trained on medical data securely. The platform combines the security of Intel Software Guarded eXtensions, transparency of blockchain technology, and verifiability of open algorithms and source codes. Genie shares insights of genetic and medical data without exposing anyone's raw data. All data is instantly encrypted upon upload and contributed to the models that the user chooses. The usage of the model and the value generated from the genetic and health data will be tracked via a blockchain, giving the data transparent and immutable ownership.

See also in the TTI/Vanguard archive:

- Julian Ranger: *Your Life, Your Call*, London, England, July 2014.

ANDRE LUCKOW

GM, BMW Back Blockchain Data Sharing for Self-Driving

By Ian Allison

Coindesk | April 19, 2019
https://www.coindesk.com/gm-bmw-back-blockchain-data-sharing-for-self-driving-cars

Car giants General Motors and BMW are backing blockchain tech as a way to share self-driving car data among themselves and other automakers. It's all part of a bid to unlock valuable data held in

silos, which will ultimately get autonomous vehicles on the road sooner. Exploratory work in this area is being done under the auspices of the Mobility Open Blockchain Initiative, a consortium formed last year to harmonize the development of distributed ledger technology across the "smart mobility" industry.

See also in the TTI/Vanguard archive:

- Yorke Rhodes: *Does Blockchain Change How We Think about Security?* Washington, D.C., September 2016.

ROGER MEIKE

Blockchain-Inspired Future Accounting

By Corinne Finegan and Roger Meike

Medium | March 25, 2019
https://medium.com/blueprint-by-intuit/blockchain-inspired-future-accounting-b866c9b0763d

Confidence in blockchain and Bitcoin are at an all-time low. In addition to recent bad press, blockchain has problems with scale and energy consumption. Despite this, the concepts blockchain leverages hold real promise in unlocking value and better outcomes for consumers and business. We'll explore one overlooked and potentially breakthrough aspect that blockchain has made newly relevant: Triple Entry Bookkeeping.

See also in the TTI/Vanguard archive:

- Bill Maurer: *Understanding Money,* Pittsburgh, Pennsylvania, October 2012.

CHRIS MONROE

Quantum Computing Is a Marathon Not a Sprint

By Christopher Monroe

IONQ | April 21, 2019
https://venturebeat.com/2019/04/21/quantum-computing-is-a-marathon-not-a-sprint/

I've spent more than 25 years as a physicist researching quantum computers — machines that store and process information on individual atoms or particles, like photons — and I've started a company that is building them. I am convinced quantum computing is in fact a breakthrough technology that offers the only known way to attack some of the world's hardest problems in medicine, transportation, computer security, and other areas we haven't yet foreseen. We must be clear, however, about what is and isn't happening next. The big quantum computing discoveries that will most impact society are still years away.

How to Evaluate Computers That Don't Quite Exist
By Adrian Cho

Science | Jun. 26, 2019
https://www.sciencemag.org/news/2019/06/how-evaluate-computers-don-t-quite-exist

To gauge the performance of a supercomputer, computer scientists turn to a standard tool: a set of algorithms called LINPACK that tests how fast the machine solves problems with huge numbers of variables. For quantum computers, which might one day solve certain problems that overwhelm conventional computers, no such benchmarking standard exists. Yet researchers are making some of their first attempts to take the measure of quantum computers. Last week, Margaret Martonosi, a computer scientist at Princeton University, and colleagues presented a head-to-head comparison of quantum computers from IBM, Rigetti Computing in Berkeley, California, and the University of Maryland (UMD) in College Park. The UMD machine, which uses trapped ions, ran a majority of 12 test algorithms more accurately than the other superconducting machines, the team reported at the International Symposium on Computer Architecture in Phoenix. Christopher Monroe, a UMD physicist and founder of the company IonQ, predicts such comparisons will become the standard.

See also in the TTI/Vanguard archive:

- Prem Kumar: *Quantum Communications: Current Status and Future Challenges*, San Francisco, California, December 2018.
- Rodney Van Meter: *A Blueprint for Building a Quantum Computer*, San Francisco, California, December 2014.
- Rodney Van Meter: *The Quantum Computing Industry Pushes Up Shoots*, Tokyo, Japan, March 2017

KAUSIK MUNSI and IVAN GUDYMENKO

German Government Says Blockchain Can "Support Europe's Unity at a Fundamental Level"
By Marie Huillet

Cointelegraph | March 27, 2019
https://cointelegraph.com/news/german-govt-says-blockchain-can-support-europes-unity-at-a-fundamental-level

Germany's Federal Office for Migration and Refugees (BAFM) has found that blockchain has far-reaching potential to improve asylum procedures. Following a successfully completed proof-of-concept (PoC), the findings were published on March 26 in a white paper. The PoC focused on evaluating blockchain's potential to support two crucial aspects of asylum procedures: the creation of reliable and secure digital identities and improving communication and cooperation between authorities at a municipal, state and national level.

See also in the TTI/Vanguard archive:

- Jini Kim and Mikey Dickerson: *Fixing HealthCare.gov*, Washington, D.C., September 2014

MICHAEL MYLREA

Building Trust in Blockchain for the Electric Grid
By Lynne Roeder

PNNL | Mar 29, 2019
https://www.pnnl.gov/news-media/building-trust-blockchain-electric-grid

PNNL pilots two use cases applying blockchain technology to improve the cybersecurity of critical electricity infrastructure.

See also in the TTI/Vanguard archive:

- John Woodward: *Critical Infrastructure Protection*, Austin, Texas, February 2001.

RAFAIL OSTROVSKY

A Blockchain Based on Gossip?—a Position Paper
By Robbert van Renesse

Distributed Cryptocurrencies and Consensus Ledgers (DCCL 2016) | July 2016
https://www.zurich.ibm.com/dccl/papers/renesse_dccl.pdf

A blockchain is an append-only sequence of blocks of arbitrary data. The two most popular approaches to blockchains are permissionless blockchains based on Proof of Work and permissioned blockchains based on Byzantine consensus or Byzantine Fault Tolerance. The first is based on competitions between anonymous participants solving cryptopuzzles, while the latter is a cooperative approach based on mutual trust between participants. Major problems with PoW approaches include that the energy per transaction is enormous, the transaction rate is very low, and the latency is very high. A major problem with BFT is that membership is closed. Various other approaches to blockchains have been proposed to address these problems. In this paper we propose yet another approach, based on gossip (aka epidemiological protocols).

See also in the TTI/Vanguard archive:

- Reid Williams: *Real-Time Messaging for the Decentralized Web*, Boston, April 2017

TIMOTHY PARSONS

Our New Science, Technology Assessment, and Analytics Team

WatchBlog | January 29, 2019
https://blog.gao.gov/2019/01/29/our-new-science-technology-assessment-and-analytics-team/

The Government Accountability Office (GAO) routinely provides analysis of how federal agencies manage and employ science and technology, such as regenerative medicine, 5G wireless communication, and quantum computing. In addition to our more traditional audit work, we've also conducted technology assessments for nearly two decades. These forward-looking analyses examine the potential benefits and challenges of emerging technologies, such as artificial intelligence. STAA (Science, Technology Assessment, and Analytics) will combine and enhance our technology assessment functions and our science and technology evaluation into a single, more prominent office to better meet Congress's growing need for information on these important issues.

See also in the TTI/Vanguard archive:

- Tom Kalil: *If You Don't Like the News, Go Out and Make Some of Your Own*, San Francisco, December 2017

BRIAN PLATZ

Letting Data Defend Itself: Benefits of Data-Centric Security

By Karen D. Schwartz | Jul 29, 2019

https://www.itprotoday.com/data-security-and-encryption/letting-data-defend-itself-benefits-data-centric-security

Fluree co-CEO Brian Platz discusses why data-centric security is now the best way to store and protect data.

See also in the TTI/Vanguard archive:

- Adam Ghetti: *Protecting Data, Not Networks*, Washington, D.C., September 2016.

JEANNETTE WING

CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

By Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing

Proceedings of the 33rd International Conference on Machine Learning | February 24, 2016
http://proceedings.mlr.press/v48/gilad-bachrach16.pdf

Applying machine learning to a problem which involves medical, financial, or other types of sensitive data, not only requires accurate predictions but also careful attention to maintaining data privacy and security. Legal and ethical requirements may prevent the use of cloud-based machine learning solutions for such tasks. In this work, we will present a method to convert learned neural networks to CryptoNets, neural networks that can be applied to encrypted data. This allows a data owner to send their data in an encrypted form to a cloud service that hosts the network. The encryption ensures that the data remains confidential since the cloud does not have access to the keys needed to decrypt it. Nevertheless, we will show that the cloud service is capable of applying the neural network to the encrypted data to make encrypted predictions, and also return them in encrypted form. These encrypted predictions can be sent back to the owner of the secret key who can decrypt them. Therefore, the cloud service does not gain any information about the raw data nor about the prediction it made.

See also in the TTI/Vanguard archive:

- Jeannette Wing: Computational Thinking and Thinking about Computing, Washington, D.C., May 2009.
- Raluca Ada Popa: Enhancing End-to-End Encryption with Computation on Encrypted Data, Washington, D.C., September 2016.
- Craig Gentry: *Manipulating Data While It Is Encrypted*, Washington, D.C., May 2010.

The State of Hyperledger With Brian Behlendorf

By Swapnil Bhartiya
October 25, 2018
https://www.linuxfoundation.org/blog/2018/10/the-state-of-hyperledger-with-brian-behlendorf/

Brian Behlendorf has been heading the Hyperledger project since the early days. We sat down with him at Open Source Summit to get an update on the Hyperledger project.

Hyperledger has grown in a way that mirrors the growth of the blockchain industry. "When we started, all the excitement was around bitcoin," said Brian Behlendorf, Executive Director of Hyperledger. Initially, it was more about moving money around. But the industry started to go beyond that and started to see if it "could be used as a way to reestablish how trust works on the Internet and try to decentralize a lot of things that today with led to being centralized."

"It might be OK for things like social networks or ride-sharing services to be centralized, but if you are talking about the banking or supply chain, you may not want that to be centralized," said Behlendorf.

As the industry has evolved around blockchain so did Hyperledger. "We realized pretty early that we needed to be a home for a lot of different ways to build a blockchain. It wasn't going to be like the Linux kernel project with one singular architecture," said Behlendorf.

Hyperledger projects

It was going to be more than just one architecture. Today, Hyperledger has 10 different technology projects. Five of those are called frameworks. Two of those frameworks are now production quality, including Hyperledger Fabric and Hyperledger Sawtooth.

“These two frameworks now drive about 40 production networks that we see out there and about 60 different vendors, hosts, and other companies building on top of it,” said Behlendorf. “One way we have grown is by growing the commercial ecosystem around this code.”

Hyperledger has created software stacks that organizations like banks run to participate in a blockchain project and a distributed ledger with several of other organizations with whom they want to do business.

Global growth

One region where Hyperledger is witnessing incredible interest is mainland China. “The Chinese government has actually said this is a top-level priority for them, to figure out how to make distributed ledgers work,” said Behlendorf. “About 20 percent of our members come from Chinese companies like Baidu, Tencent, and Huawei. These companies are actually contributing code, which is great to see.”

As the adoption of Hyperledger projects is growing, the organization is also working on creating training and education courses in partnership with edX to meet this growing demand. Hyperledger also has a technical working group focused on communicating in Chinese with developers who are there to help them get involved with the project.

Hyperledger aims to be a global initiative. “There are a few Silicon Valley companies involved, but it’s New York. It’s London. It’s Singapore. It’s incredibly broad,” said Behlendorf. “That’s been really reassuring because open source is a global phenomenon and really should be about kind of uplifting all regions. So it’s been a great journey.”

The Street

What Is Ripple and How Does it Work?

Ripple has the third-largest market cap of any cryptocurrency. What is it; how does it work; and what was it made for?

Steve Fiorillo
Jul 10, 2018
https://www.thestreet.com/investing/what-is-ripple-14644949

Once bitcoin established itself as a viable currency that had potential to stick around, other cryptocurrencies began popping up in its wake to try to dethrone it.

The two biggest cryptocurrencies in the world by market cap are Bitcoin and Ethereum, but the third largest—Ripple—stands out from them. What is it that sets Ripple apart, and what has people talking about it despite Bitcoin and Ethereum being worth so much more?

What Is Ripple?

Ripple is at once a company, a digital-payment processing system and a cryptocurrency, which is also known as XRP. This is similar to bitcoin, but Ripple's blockchain system is very different, and the currency is owned by the one company—Ripple—whereas bitcoin is mined.

The infrastructure of Ripple is designed to make transactions quicker and more convenient for banks, so it is a more popular cryptocurrency option for larger financial institutions. While Ripple is often used to refer to XRP cryptocurrency, it is actually the company (formerly known as Ripple Labs) that holds most of the XRP. Ripple not only offers a number of payment

systems, but owns approximately 60 billion XRP and holds 55 billion of it in an escrow account. They're able to sell up to one billion of these a month—though they rarely do.

Ripple's blockchain system, RippleNet, offers businesses and financial institutions a number of programs that help cross-border payments. This includes xCurrent (a payment processing system for banks), xRapid (allows financial institutions to minimize liquidity cost while using XRP as a bridge from one fiat currency to another), and xVia (allows businesses to send payments via RippleNet).

Ripple's website boasts dozens of clients that use their blockchain system, from smaller banks to some of the largest in the country. American Express (**AXP**—Get Report) , for example, announced a partnership with Ripple in 2017 that allowed for limited blockchain payments from U.S. businesses to U.K. businesses.

Ripple also brags of the company's versatility, ability to help large financial institutions, and its exceptionally fast transaction time. At the heart of all of this is the currency itself—XRP.

XRP: What Is Ripple Currency?

XRP, the digital asset of Ripple, is supposedly capable of settling a payment within 4 seconds and handling 1,500 transactions every second.

Though Ripple has differed from other cryptocurrencies in a number of ways, one way it remains similar is that there is a finite amount of XRP created, and that is all there will be. In the case of XRP, 100 billion exist, 60% of which are owned by Ripple.

Were a financial institution to use xRapid to help with cross-border payments from one fiat currency to another, XRP is what is used mid-transaction for liquidity. This makes Ripple and XRP a bit unusual in the world of cryptocurrency: It's not really used as a currency, to the point that Ripple CEO Brad Garlinghouse recently told a conference audience, "I don't think about the digital asset market. I think about the customer experience."

Because so much XRP is owned by Ripple and isn't really used as a currency, some have alleged that it should be considered a security. Garlinghouse, however, has said he believes it should not be, as it serves a utilitarian purpose, and owning XRP does not mean owning a part of the company Ripple.

Ripple vs. Bitcoin

Bitcoin, as the most well-known cryptocurrency with easily the largest market cap, is an easy comparison to make when discussing other cryptocurrencies. However, Ripple is quite different from bitcoin in a number of ways.

Some notable ways are how Ripple sells itself, especially with regards to transaction speeds. One of the more notable complaints about bitcoin is how long a transaction can take. With the extreme volatility of bitcoin, it creates the risk that when the transaction is finalized, you may not be getting the amount of BTC you expected when you first initiated it. But with Ripple claiming 4-second long transaction times, that's far less of a concern.

Bitcoin is entirely decentralized, as it was made with the purpose of allowing for financial transactions without the need of a third party like a bank. Ripple, on the other hand, literally sells its services to banks and financial institutions. With most of the XRP being owned by the company, the network is far more centralized.

Despite each falling under the large umbrella of "cryptocurrencies," ripple and bitcoin's purposes couldn't be further apart. Bitcoin was made in the hopes of creating a brand new financial system entirely. Ripple, creating its digital token to help with asset transfers, seeks to assist existing financial systems and upgrade their capabilities for worldwide transactions.

Is There Ripple Mining?

Another notable difference between ripple and bitcoin is mining. Bitcoins are known for being mined, a controversial process due to its combination of expensive technology required and vast amount of energy needed to do it.

Whereas all 21 million bitcoins must be mined, all 100 billion XRP already exist and require no mining. So far, ripple has a circulating supply of over 39 billion XRP. Those interested in owning any will need to purchase it via a cryptocurrency exchange—though this is much cheaper than spending money on crypto mining rigs.

XRP Price

Ripple price has fluctuated wildly in its short history. As of this writing, XRP is valued at 48 cents, with a market cap of about $18.8 billion, per CoinMarketCap.

For the overwhelming majority of XRP's existence, its value has languished under a dollar—if an XRP owner in 2016 saw it was worth nearly half a dollar now, they'd be amazed at how much it increased; by the time January 2017 came, the value of one XRP was just $0.006. It began to climb later in the year, and by the end of May it had hit nearly $0.40. It fell not long after, but stayed in the 10 cents and 20 cents range for the next 6 months or so.

December 2017 saw a major spike. By the 14th, it had surpassed 80 cents in value. One week later, on the 21st, the value was above $1 for the first time in its history. A week after that it hit $2, and on Jan. 4, 2018, it reached its high point: $3.84 in value, and over $148 billion in market cap. It even overtook Ethereum for second-highest cryptocurrency market cap. This was around the same time as bitcoin's astonishing ascension to a peak of more than $20,000 in value, leading to an explosion in the crypto market.

Unsurprisingly, such a massive increase was not sustainable for either one. By February, the value of ripple had cratered back down to under a dollar. Since late February, it has yet to reach that $1 mark.

Ripple Wallets

Ripple wallets are similar to bitcoin wallets, with secure keys that allow for transactions. With ripple, though, wallets require a minimum of 20 XRP for the initial deposit.

Like with other cryptocurrency wallets, there are different types, including software wallets and mobile wallets available for Android and iOS. It is most often recommended, however, that you store your ripple (and other cryptocurrencies) in a hardware wallet. Hardware wallets are much more secure because they store the contents offline. One notable hardware manufacturer with a ripple-supporting wallet is Ledger, whose Ledger Nano S wallet allows for ripple.

How to Buy Ripple

Buying ripple is not yet as convenient as buying bitcoin is. Occasionally a cryptocurrency exchange like Bitstamp will allow you to exchange USD for XRP, but rarely is that the case.

Other exchanges that sell ripple, including Coinbase and Binance, will instead need you to exchange a different cryptocurrency like bitcoin or ether in order to acquire XRP. Regardless of the currency you're exchanging for XRP, you'll need an account on the exchange and a ripple wallet where you will send your XRP. Because XRP is supposed to be so notoriously fast, once you have everything in order and initiate the transaction, you should have your XRP relatively quickly. Learn more here in our guide on how to buy XRP.

The Ripple Protocol Consensus Algorithm

David Schwartz
david@ripple.com

Noah Youngs
nyoungs@nyu.edu

Arthur Britto
arthur@ripple.com

Abstract

While several consensus algorithms exist for the Byzantine Generals Problem, specifically as it pertains to distributed payment systems, many suffer from high latency induced by the requirement that all nodes within the network communicate synchronously. In this work, we present a novel consensus algorithm that circumvents this requirement by utilizing collectively-trusted subnetworks within the larger network. We show that the "trust" required of these subnetworks is in fact minimal and can be further reduced with principled choice of the member nodes. In addition, we show that minimal connectivity is required to maintain agreement throughout the whole network. The result is a low-latency consensus algorithm which still maintains robustness in the face of Byzantine failures. We present this algorithm in its embodiment in the Ripple Protocol.

Contents

1. Introduction

Interest and research in distributed consensus systems has increased markedly in recent years, with a central focus being on distributed payment networks. Such networks allow for fast, low-cost transactions which are not controlled by a centralized source. While the economic benefits and drawbacks of such a system are worthy of much research in and of themselves, this work focuses on some of the technical challenges that all distributed payment systems must face. While these problems are varied, we group them into three main categories: correctness, agreement, and utility.

By correctness, we mean that it is necessary for a distributed system to be able to discern the difference between a correct and fraudulent transaction. In traditional fiduciary settings, this is done through trust between institutions and cryptographic signatures that guarantee a transaction is indeed coming from the institution that it claims to be coming from. In distributed systems, however, there is no such trust, as the identity of any and all members in the network may not even be known. Therefore, alternative methods for correctness must be

utilized.

Agreement refers to the problem of maintaining a single global truth in the face of a decentralized accounting system. While similar to the correctness problem, the difference lies in the fact that while a malicious user of the network may be unable to create a fraudulent transaction (defying correctness), it may be able to create multiple correct transactions that are somehow unaware of each other, and thus combine to create a fraudulent act. For example, a malicious user may make two simultaneous purchases, with only enough funds in their account to cover each purchase individually, but not both together. Thus each transaction by itself is correct, but if executed simultaneously in such a way that the distributed network as a whole is unaware of both, a clear problem arises, commonly referred to as the "Double-Spend Problem" [1]. Thus the agreement problem can be summarized as the requirement that only one set of globally recognized transactions exist in the network.

Utility is a slightly more abstract problem, which we define generally as the "usefulness" of a distributed payment system, but which in practice most often simplifies to the latency of the system. A distributed system that is both correct and in agreement but which requires one year to process a transaction, for example, is obviously an inviable payment system. Additional aspects of utility may include the level of computing power required to participate in the correctness and agreement processes or the technical proficiency required of an end user to avoid being defrauded in the network.

Many of these issues have been explored long before the advent of modern distributed computer systems, via a problem known as the "Byzantine Generals Problem" [2]. In this problem, a group of generals each control a portion of an army and must coordinate an attack by sending messengers to each other. Because the generals are in unfamiliar and hostile territory, messengers may fail to reach their destination (just as nodes in a distributed network may fail, or send corrupted data instead of the intended message). An additional aspect of the problem is that some of the generals may be traitors, either individually, or conspiring together, and so messages may arrive which are intended to create a false plan that is doomed to failure for the loyal generals (just as malicious members of a distributed system may attempt to convince the system to accept fraudulent transactions, or multiple versions of the same truthful transaction that would result in a double-spend). Thus a distributed payment system must be robust both in the face of standard failures, and so-called "Byzantine" failures, which may be coordinated and originate from multiple sources in the network.

In this work, we analyze one particular implementation of a distributed payment system: the Ripple Protocol. We focus on the algorithms utilized to achieve the above goals of correctness, agreement, and utility, and show that all are met (within necessary and predetermined tolerance thresholds, which are well-understood). In addition, we provide code that simulates the consensus process with parameterizable network size, number of malicious users, and message-sending latencies.

2. Definitions, Formalization and Previous Work

We begin by defining the components of the Ripple Protocol. In order to prove correctness, agreement, and utility properties, we first formalize those properties into axioms. These properties, when grouped together, form the notion of *consensus*: the state in which nodes in the network reach correct agreement. We then highlight some previous results relating to consensus algorithms, and finally state the goals of consensus for the Ripple Protocol within our formalization framework.

2.1 Ripple Protocol Components

We begin our description of the ripple network by defining the following terms:

- **Server:** A server is any entity running the Ripple Server software (as opposed to the Ripple Client software which only lets a user send and receive funds), which participates in the consensus process.
- **Ledger:** The ledger is a record of the amount of currency in each user's account and represents the "ground truth" of the network. The ledger is repeatedly updated with transactions that successfully pass through the consensus process.
- **Last-Closed Ledger:** The last-closed ledger is the most recent ledger that has been ratified by the consensus process and thus represents the current state of the network.
- **Open Ledger:** The open ledger is the current operating status of a node (each node maintains its own open ledger). Transactions initiated by end users of a given server are applied to the open

ledger of that server, but transactions are not considered final until they have passed through the consensus process, at which point the open ledger becomes the last-closed ledger.

- **Unique Node List (UNL):** Each server, *s*, maintains a unique node list, which is a set of other servers that *s* queries when determining consensus. Only the votes of the other members of the UNL of *s* are considered when determining consensus (as opposed to every node on the network). Thus the UNL represents a subset of the network which when taken collectively, is "trusted" by *s* to not collude in an attempt to defraud the network. Note that this definition of "trust" does not require that each individual member of the UNL be trusted (see section 3.2).

- **Proposer:** Any server can broadcast transactions to be included in the consensus process, and every server attempts to include every valid transaction when a new consensus round starts. During the consensus process, however, only proposals from servers on the UNL of a server *s* are considered by *s*.

2.2 Formalization

We use the term *nonfaulty* to refer to nodes in the network that behave honestly and without error. Conversely, a *faulty* node is one which experiences errors which may be honest (due to data corruption, implementation errors, etc.), or malicious (Byzantine errors). We reduce the notion of validating a transaction to a simple binary decision problem: each node must decide from the information it has been given on the value 0 or 1.

As in Attiya, Dolev, and Gill, 1984 [3], we define consensus according to the following three axioms:

1. **(C1):** Every nonfaulty node makes a decision in finite time
2. **(C2):** All nonfaulty nodes reach the same decision value
3. **(C3):** 0 and 1 are both possible values for all nonfaulty nodes. (This removes the trivial solution in which all nodes decide 0 or 1 regardless of the information they have been presented).

2.3 Existing Consensus Algorithms

There has been much research done on algorithms that achieve consensus in the face of Byzantine errors. This previous work has included extensions to cases where all participants in the network are not known ahead of time, where the messages are sent asynchronously (there is no bound on the amount of time an individual node will take to reach a decision), and where there is a delineation between the notion of strong and weak consensus.

One pertinent result of previous work on consensus algorithms is that of Fischer, Lynch, and Patterson, 1985 [4], which proves that in the asynchronous case, non-termination is always a possibility for a consensus algorithm, even with just one faulty process. This introduces the necessity for time-based heuristics, to ensure convergence (or at least repeated iterations of non-convergence). We shall describe these heuristics for the Ripple Protocol in section 3.

The strength of a consensus algorithm is usually measured in terms of the fraction of faulty processes it can tolerate. It is provable that no solution to the Byzantine Generals problem (which already assumes synchronicity, and known participants) can tolerate more than $(n-1)/3$ byzantine faults, or 33% of the network acting maliciously [2]. This solution does not, however, require verifiable authenticity of the messages delivered between nodes (digital signatures). If a guarantee on the unforgeability of messages is possible, algorithms exist with much higher fault tolerance in the synchronous case.

Several algorithms with greater complexity have been proposed for Byzantine consensus in the asynchronous case. FaB Paxos [5] will tolerate $(n-1)/5$ Byzantine failures in a network of n nodes, amounting to a tolerance of up to 20% of nodes in the network colluding maliciously. Attiya, Doyev, and Gill [3] introduce a phase algorithm for the asynchronous case, which can tolerate $(n-1)/4$ failures, or up to 25% of the network. Lastly, Alchieri et al., 2008 [6] present BFT-CUP, which achieves Byzantine consensus in the asynchronous case even with unknown participants, with the maximal bound of a tolerance of $(n-1)/3$ failures, but with additional restrictions on the connectivity of the underlying network.

2.4 Formal Consensus Goals

Our goal in this work is to show that the consensus algorithm utilized by the Ripple Protocol will achieve consensus at each ledger-close (even if consensus is the trivial consensus of all transactions being rejected), and that the trivial consensus will only be reached with a known probability, even in the face of Byzantine failures.

Since each node in the network only votes on proposals from a trusted set of nodes (the other nodes in its UNL), and since each node may have differing UNLs, we also show that only one consensus will be reached amongst all nodes, regardless of UNL membership. This goal is also referred to as preventing a "fork" in the network: a situation in which two disjoint sets of nodes each reach consensus independently, and two different last-closed ledgers are observed by nodes on each node-set.

Lastly we will show that the Ripple Protocol can achieve these goals in the face of $(n-1)/5$ failures, which is not the strongest result in the literature, but we will also show that the Ripple Protocol possesses several other desirable features that greatly enhance its utility.

3. Ripple Consensus Algorithm

The Ripple Protocol consensus algorithm (RPCA), is applied every few seconds by all nodes, in order to maintain the correctness and agreement of the network. Once consensus is reached, the current ledger is considered "closed" and becomes the last-closed ledger. Assuming that the consensus algorithm is successful, and that there is no fork in the network, the last-closed ledger maintained by all nodes in the network will be identical.

3.1 Definition

The RPCA proceeds in rounds. In each round:

- Initially, each server takes all valid transactions it has seen prior to the beginning of the consensus round that have not already been applied (these may include new transactions initiated by end-users of the server, transactions held over from a previous consensus process, etc.), and makes them public in the form of a list known as the "candidate set".
- Each server then amalgamates the candidate sets of all servers on its UNL, and votes on the veracity of all transactions.
- Transactions that receive more than a minimum percentage of "yes" votes are passed on to the next round, if there is one, while transactions that do not receive enough votes will either be discarded, or included in the candidate set for the beginning of the consensus process on the next ledger.
- The final round of consensus requires a minimum percentage of 80% of a server's UNL agreeing on a transaction. All transactions that meet this requirement are applied to the ledger, and that ledger is closed, becoming the new last-closed ledger.

3.2 Correctness

In order to achieve correctness, given a maximal amount of Byzantine failures, it must be shown that it is impossible for a fraudulent transaction to be confirmed during consensus, unless the number of faulty nodes exceeds that tolerance. The proof of the correctness of the RPCA then follows directly: since a transaction is only approved if 80% of the UNL of a server agrees with it, as long as 80% of the UNL is honest, no fraudulent transactions will be approved. Thus for a UNL of n nodes in the network, the consensus protocol will maintain correctness so long as:

$$f \leq (n-1)/5 \tag{1}$$

where f is the number Byzantine failures. In fact, even in the face of $(n-1)/5+1$ Byzantine failures, correctness is still technically maintained. The consensus process will fail, but it will still not be possible to confirm a fraudulent transaction. Indeed it would take $(4n+1)/5$ Byzantine failures for an incorrect transaction to be confirmed. We call this second bound the bound for *weak* correctness, and the former the bound for *strong* correctness.

It should also be noted that not all "fraudulent" transactions pose a threat, even if confirmed during consensus. Should a user attempt to double-spend funds in two transactions, for example, even if both transactions are confirmed during the consensus process, after the first transaction is applied, the second will fail, as the funds are no longer available. This robustness is due to the fact that transactions are applied deterministically, and that consensus ensures that all nodes in the network are applying the deterministic rules to the same set of transactions.

For a slightly different analysis, let us assume that the probability that any node will decide to collude and join a nefarious cartel is p_c. Then the probability of correctness is given by p^*, where:

$$p^* = \sum_{i=0}^{\lceil(\frac{n-1}{5})\rceil} \binom{n}{i} p_c^i (1-p_c)^{n-i} \tag{2}$$

This probability represents the likelihood that the size of the nefarious cartel will remain below the maximal

threshold of Byzantine failures, given p_c. Since this likelihood is a binomial distribution, values of p_c greater than 20% will result in expected cartels of size greater than 20% of the network, thwarting the consensus process. In practice, a UNL is not chosen randomly, but rather with the intent to minimize p_c. Since nodes are not anonymous but rather cryptographically identifiable, selecting a UNL of nodes from a mixture of continents, nations, industries, ideologies, etc. will produce values of p_c much lower than 20%. As an example, the probability of the Anti-Defamation League and the Westboro Baptist Church colluding to defraud the network, is certainly much, much smaller than 20%. Even if the UNL has a relatively large p_c, say 15%, the probability of correctness is extremely high even with only 200 nodes in the UNL: 97.8%.

A graphical representation of how the probability of incorrectness scales as a function of UNL size for differing values of p_c is depicted in Figure 1. Note that here the vertical axis represents the probability of a nefarious cartel thwarting consensus, and thus lower values indicate greater probability of consensus success. As can be seen in the figure, even with a p_c as high as 10%, the probability of consensus being thwarted very quickly becomes negligible as the UNL grows past 100 nodes.

3.3 Agreement

To satisfy the agreement requirement, it must be shown that all nonfaulty nodes reach consensus on the same set of transactions, regardless of their UNLs. Since the UNLs for each server can be different, agreement is not inherently guaranteed by the correctness proof. For example, if there are no restrictions on the membership of the UNL, and the size of the UNL is not larger than $0.2 * n_{total}$ where n_{total} is the number of nodes in the entire network, then a fork is possible. This is illustrated by a simple example (depicted in figure 2): imagine two cliques within the UNL graph, each larger than $0.2 * n_{total}$. By cliques, we mean a set of nodes where each node's UNL is the selfsame set of nodes. Because these two cliques do not share any members, it is possible for each to achieve a correct consensus independently of each other, violating agreement. If the connectivity of the two cliques surpasses $0.2 * n_{total}$, then a fork is no longer possible, as disagreement between the cliques would prevent consensus from being reached at the 80% agreement threshold that is required.

An upper bound on the connectivity required to prove agreement is given by:

$$|UNL_i \cap UNL_j| \geq \frac{1}{5}\max(|UNL_i|, |UNL_j|) \forall i, j \quad (3)$$

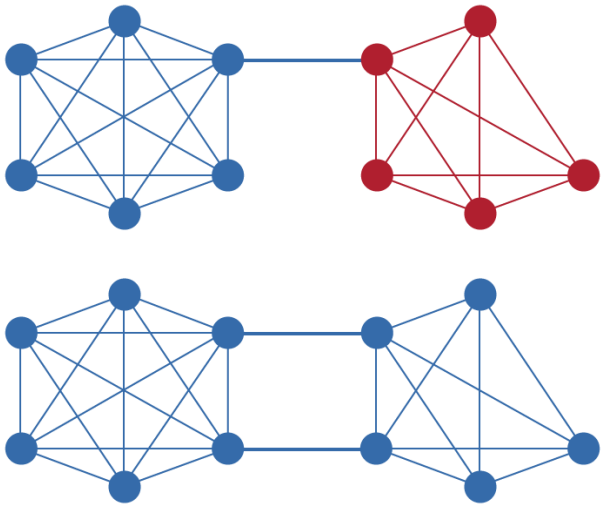


Figure 2. An example of the connectivity required to prevent a fork between two UNL cliques.

This upper bound assumes a clique-like structure of UNLs, i.e. nodes form sets whose UNLs contain other nodes in those sets. This upper bound guarantees that no two cliques can reach consensus on conflicting transactions, since it becomes impossible to reach the 80% threshold required for consensus. A tighter bound is possible when indirect edges between UNLs are taken into account as well. For example, if the structure of the network is not clique-like, a fork becomes much more difficult to achieve, due to the greater entanglement of the UNLs of all nodes.

It is interesting to note that no assumptions are made about the nature of the intersecting nodes. The intersection of two UNLs may include faulty nodes, but so long as the size of the intersection is larger than the bound required to guarantee agreement, and the total number of faulty nodes is less than the bound required to satisfy strong correctness, then both correctness and agreement will be achieved. That is to say, agreement is dependent solely on the size of the intersection of nodes, not on the size of the intersection of nonfaulty nodes.

3.4 Utility

While many components of utility are subjective, one that is indeed provable is convergence: that the consensus process will terminate in finite time.

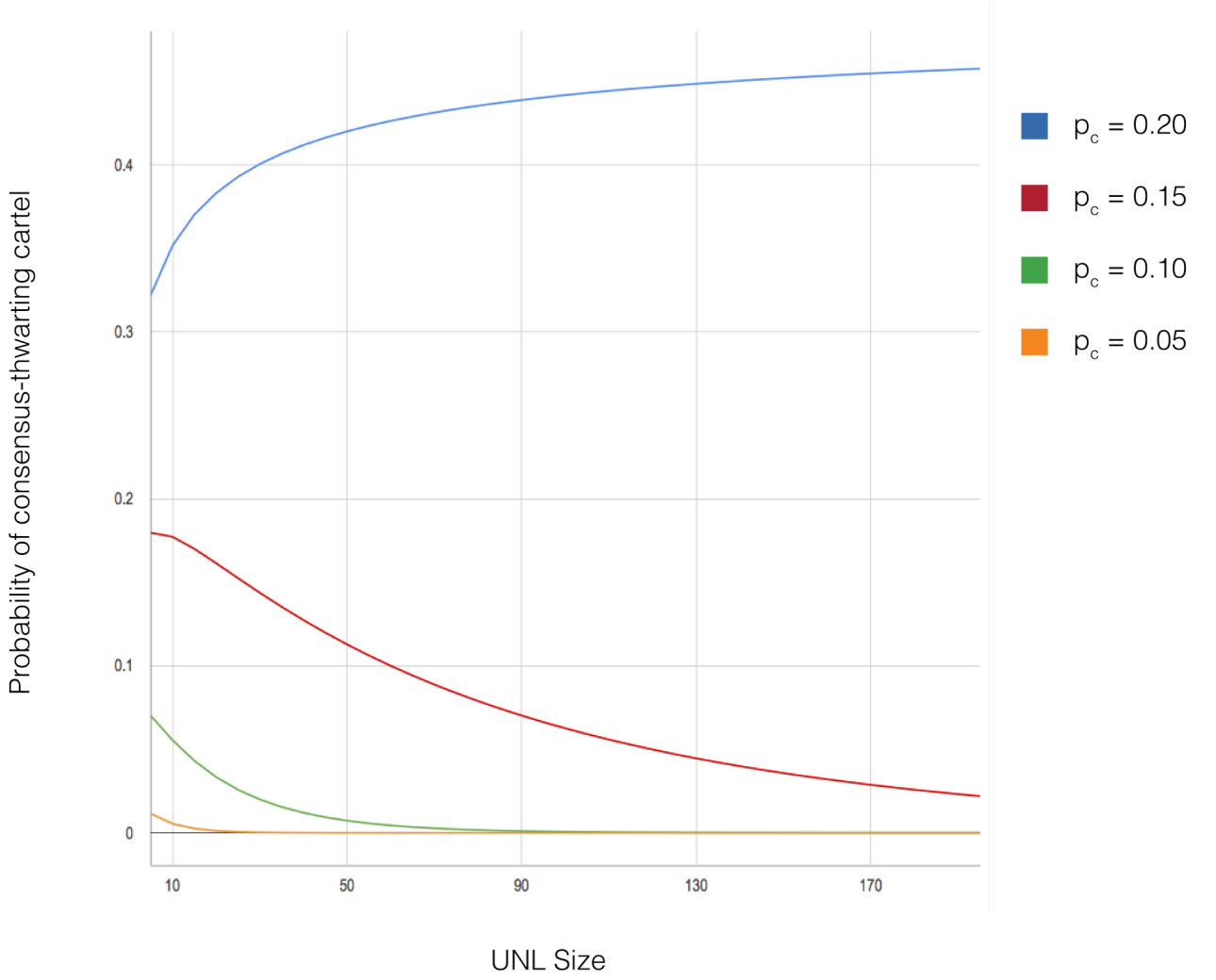


Figure 1. Probability of a nefarious cartel being able to thwart consensus as a function of the size of the UNL, for different values of p_c, the probability that any member of the UNL will decide to collude with others. Here, lower values indicate a higher probability of consensus success.

3.4.1 Convergence

We define convergence as the point in which the RPCA reaches consensus with strong correctness on the ledger, and that ledger then becomes the last-closed ledger. Note that while technically weak correctness still represents convergence of the algorithm, it is only convergence in the trivial case, as proposition **C3** is violated, and no transactions will ever be confirmed. From the results above, we know that strong correctness is always achievable in the face of up to $(n-1)/5$ Byzantine failures, and that only one consensus will be achieved in the entire network so long as the UNL-connectedness condition is met (Equation 3). All that remains is to show that when both of these conditions are met, consensus is reached in finite time.

Since the consensus algorithm itself is deterministic, and has a preset number of rounds, t, before consensus is terminated, and the current set of transactions are declared approved or not-approved (even if at this point no transactions have more than the 80% required agreement, and the consensus is only the trivial consensus), the limiting factor for the termination of the algorithm is the communication latency between nodes. In order to bound this quantity, the response-time of nodes is monitored, and nodes who's latency grows larger than a preset bound b are removed from all UNLs. While this guarantees that consensus will terminate with an upper bound of tb, it is important to note that the bounds described for correctness and agreement above must be met by the *final* UNL, after all nodes that will be

dropped have been dropped. If the conditions hold for the initial UNLs for all nodes, but then some nodes are dropped from the network due to latency, the correctness and agreement guarantees do not automatically hold but must be satisfied by the new set of UNLs.

3.4.2 Heuristics and Procedures

As mentioned above, a latency bound heuristic is enforced on all nodes in the Ripple Network to guarantee that the consensus algorithm will converge. In addition, there are a few other heuristics and procedures that provide utility to the RPCA.

- There is a mandatory 2 second window for all nodes to propose their initial candidate sets in each round of consensus. While this does introduce a lower bound of 2 seconds to each consensus round, it also guarantees that all nodes with reasonable latency will have the ability to participate in the consensus process.

- As the votes are recorded in the ledger for each round of consensus, nodes can be flagged and removed from the network for some common, easily-identifiable malicious behaviors. These include nodes that vote "No" on every transaction, and nodes that consistently propose transactions which are not validated by consensus.

- A curated default UNL is provided to all users, which is chosen to minimize p_c, described in section 3.2. While users can and should select their own UNLs, this default list of nodes guarantees that even naive users will participate in a consensus process that achieves correctness and agreement with extremely high probability.

- A network split detection algorithm is also employed to avoid a fork in the network. While the consensus algorithm certifies that the transactions on the last-closed ledger are correct, it does not prohibit the possibility of more than one last-closed ledger existing on different subsections of the network with poor connectivity. To try and identify if such a split has occurred, each node monitors the size of the active members of its UNL. If this size suddenly drops below a preset threshold, it is possible that a split has occurred. In order to prevent a false positive in the case where a large section of a UNL has temporary latency, nodes are allowed to publish a "partial validation", in which they do not process or vote on transactions, but declare that are still participating in the consensus process, as opposed to a different consensus process on a disconnected subnetwork.

- While it would be possible to apply the RPCA in just one round of consensus, utility can be gained through multiple rounds, each with an increasing minimum-required percentage of agreement, before the final round with an 80% requirement. These rounds allow for detection of latent nodes in the case that a few such nodes are creating a bottleneck in the transaction rate of the network. These nodes will be able to initially keep up during the lower-requirement rounds but fall behind and be identified as the threshold increases. In the case of one round of consensus, it may be the case that so few transactions pass the 80% threshold, that even slow nodes can keep up, lowering the transaction rate of the entire network.

4. Simulation Code

The provided simulation code demonstrates a round of RPCA, with parameterizable features (the number of nodes in the network, the number of malicious nodes, latency of messages, etc.). The simulator begins in perfect disagreement (half of the nodes in the network initially propose "yes", while the other half propose "no"), then proceeds with the consensus process, showing at each stage the number of yes/no votes in the network as nodes adjust their proposals based upon the proposals of their UNL members. Once the 80% threshold is reached, consensus is achieved. We encourage the reader to experiment with different values of the constants defined at the beginning of "Sim.cpp", in order to become familiar with the consensus process under different conditions.

5. Discussion

We have described the RPCA, which satisfies the conditions of correctness, agreement, and utility which we have outlined above. The result is that the Ripple Protocol is able to process secure and reliable transactions in a matter of seconds: the length of time required for one round of consensus to complete. These transactions are provably secure up to the bounds outlined in section 3, which, while not the strongest available in the literature for Asynchronous Byzantine consensus, do

allow for rapid convergence and flexibility in network membership. When taken together, these qualities allow the Ripple Network to function as a fast and low-cost global payment network with well-understood security and reliability properties.

While we have shown that the Ripple Protocol is provably secure so long as the bounds described in equations 1 and 3 are met, it is worth noting that these are maximal bounds, and in practice the network may be secure under significantly less stringent conditions. It is also important to recognize, however, that satisfying these bounds is not inherent to the RPCA itself, but rather requires management of the UNLs of all users. The default UNL provided to all users is already sufficient, but should a user make changes to the UNL, it must be done with knowledge of the above bounds. In addition, some monitoring of the global network structure is required in order to ensure that the bound in equation 3 is met, and that agreement will always be satisfied.

We believe the RPCA represents a significant step forward for distributed payment systems, as the low-latency allows for many types of financial transactions previously made difficult or even impossible with other, higher latency consensus methods.

6. Acknowledgments

Ripple Labs would like to acknowledge all of the people involved in the development of the Ripple Protocol consensus algorithm. Specifically, Arthur Britto, for his work on transaction sets, Jed McCaleb, for the original Ripple Protocol consensus concept, and David Schwartz, for his work on the "failure to agree is agreement to defer" aspect of consensus. Ripple Labs would also like to acknowledge Noah Youngs for his efforts in preparing and reviewing this paper.

References

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." Consulted 1.2012 (2008): 28.

[2] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine generals problem." ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3 (1982): 382-401.

[3] Attiya, C., D. Dolev, and J. Gill. "Asynchronous Byzantine Agreement." Proc. 3rd. Annual ACM Symposium on Principles of Distributed Computing. 1984.

[4] Fischer, Michael J., Nancy A. Lynch, and Michael S. Paterson. "Impossibility of distributed consensus with one faulty process." Journal of the ACM (JACM) 32.2 (1985): 374-382.

[5] Martin, J-P., and Lorenzo Alvisi. "Fast byzantine consensus." Dependable and Secure Computing, IEEE Transactions on 3.3 (2006): 202-215.

[6] Alchieri, Eduardo AP, et al. "Byzantine consensus with unknown participants." Principles of Distributed Systems. Springer Berlin Heidelberg, 2008. 22-40.

Forbes

Blockchain, Burning Man, and the Future of Governance: A Conversation With John Clippinger

Robert C. Wolcott
Feb 16, 2017

https://www.forbes.com/sites/robertwolcott/2017/02/16/blockchain-burning-man-and-the-future-of-governance-a-conversation-with-john-clippinger/#54c24d601b0b

John Clippinger always seems to be ahead of trends. In 1965, he marched in Selma, Alabama in support of civil rights. In 2013 (more prosaically), Clippinger introduced me to blockchain. When others were just discovering this methodology underlying Bitcoin, he had already been exploring how blockchain might transform business and government.

Clippinger's wider interest is how humans organize—from contract law to Burning Man—and how technologies like blockchain enable new approaches to business and government. I recently had the opportunity to explore his ideas while we were both visiting the Santa Fe Institute in New Mexico. Find a video of our conversation here.

The Rise of the Open Sector

As public trust in established institutions plummets, issues of governance become ever more urgent. Clippinger, founder of the Institute for Data-Driven Design (ID3) and a Research Scientist with MIT Media Lab, is a pioneer in the definition of what he and others refer to as the *open sector*, a movement challenging traditional, top-down leadership paradigms. "It's not the public sector, it's not the private sector, it's not under a government or the UN…It's owned by everyone and nobody." Founders create a set of initial conditions from which rules emerge through the interactions of participants.

As precedent, Clippinger cites a seminal article from 1881 by Oliver Wendell Holmes regarding the evolution of British Common Law. Holmes described how British Common Law, a basis for America's legal system, evolved from customs and norms, eventually being codified into constantly-evolving laws. "It wasn't top-down. It was constantly reinventing itself around the circumstances, and there was no single point of

control." According to Clippinger, to maximize overall prosperity, laws of engagement in the *digital* arena should best evolve this way as well.

He cautions that control of the digital sphere by governments or corporations should be resisted by anyone with a stake--which means all of us. "If we're going to have any kind of freedom, we have to have control over this." How can individuals verify identities and other data in ways acceptable to authorities, but not controlled by them?

Blockchain offers one open sector solution. It is essentially a distributed trust engine requiring no third party to verify transactions or other digital interactions. Such a capability can enable commerce, self-expression, even new forms of economic interactions and organizations. Some corporations are paying attention, from cyber security firms to banks and insurers. In late 2016, five European insurers announced a consortium to experiment with blockchain, initially in the reinsurance sector.

Clippinger cites Burning Man as another example of open sector development. What started as a small group in 1986 has evolved into an annual, week-long gathering of 70,000 in the Black Rock Desert of Nevada. Celebrating community, art and radical self-expression, Burning Man emerges each year out of the contributions of 'burners', as participants are known.

For years, Burning Man hadn't memorialized any rules. In the mid-1990s, through some tragic events, "They had an existential moment… it sort of went over the edge." That could have been the end. The founders eventually formalized the 10 Principles, such as radical inclusion, cooperation, gifting and leaving no trace. The community aggressively defends its principles. "It's always about how much you do from the top-down…. Sometimes you need a nudge from the top and sometimes you allow things to come up from the bottom. So it's sort of a living experiment."

The movement's success has spawned a worldwide network. The founders, Clippinger and others are investigating how to scale worldwide while remaining true to Burner principles, taking Burning Man, "from the bubble of the desert into a new kind of post-capitalist economy."

Public, Private, and Open Sectors Co-Evolve

This proliferation of ideas from the edge— a defining feature of the open sector— ultimately engages the public and private sectors. What may begin as rejection or competition often evolves into something richer and more complex. Notes Clippinger, "You want something to compete with the traditional sector. And as it starts to become more effective and gains legitimacy, then it's going to shape all sectors."

As traditional political structures such as the European Union or the United States experience increased stress, open sector movements percolate. Communal groups spontaneously arise in various forms across Europe, from Spain and Italy to Berlin, many in urban areas. “When you have a nation state that is not very effective… it goes back to the city. It’s a level of governance that people can participate in and be effective.” Some cities are already taking more assertive roles. London and Los Angeles endeavor for global engagement while their national governments seek isolation.

Generational factors also appear to play a role, “particularly among [millennials], who think of… how they gather and what’s legitimate and not legitimate.” The rapidity with which millennials engage and abandon new online platforms manifests the shifting of power to the edges, as well as the fluidity of open sector development.

As wider ranges of social, economic and political activities occur within the open sector, how should businesses respond? “What happens is they say, ‘we’ll wait till it matures, and then jump in.’ That’s what newspapers did, and look where it got them.” Traditional institutions in general won’t disappear, but roles and power relationships will change.

Rather than resist the transition, corporations that discover how to navigate co-evolution between public, private and open sectors will be more likely to thrive. Meanwhile, the experiments of people around the world-- their customers-- will continue to lead the way.

Why We've Created the Blockchain Impact Ledger

By Dahna Goldstein
April 15, 2019
https://medium.com/impact-ledger/why-weve-created-the-impact-ledger-86106d3affa9

The conversation about blockchain is evolving rapidly. A couple of years ago, the driving question was "what is blockchain?" Now, the questions are, "what are the best use cases for blockchain?" Or "what are some real-world examples of blockchain in action?"

Social impact may not be the first application that comes to mind for many who think about blockchain, but it is perhaps one of the most promising. While blockchain's origins were as the underpinning of cryptocurrencies, it has evolved to address challenges in a wide variety of industries. Blockchain is already being deployed in the financial sector, with over $1 billion invested in potential FinTech applications. The shipping industry is attempting to use blockchain to improve global trade. Organizations committed to social impact are developing blockchain solutions from managing identity to improving the efficacy of relief efforts, from increasing the speed and reducing the cost of remittances to providing pricing transparency to smallholder farmers to increase income and improve access to capital. However, the technology is still in the earliest stages of use to deliver social impact.

Googling "blockchain and impact" brings up many articles about the potential for blockchain to address important social issues. But what is really happening? Without a trustworthy source of aggregated information about blockchain for social impact projects, it is difficult for organizations interested in leveraging this powerful technology to learn of — and potentially coordinate with — organizations and projects with aligned pursuits. With increasing interest in understanding the lay of the blockchain-for-social-impact land, the absence of consolidated information about the growing number of projects has made research and analysis challenging. The state of play is changing rapidly — with new projects both coming and going — which has made gaining a thorough understanding of the developments and opportunities in the space hard to come by.

Until now. The Blockchain Trust Accelerator at New America is creating the Impact Ledger, an online registry of social impact blockchain projects, spanning the nonprofit, public, and for-profit sectors.

The Blockchain Impact Ledger is rooted in the thought leadership assembled at the Rockefeller Foundation's Bellagio Center for a Blockchain for Good Summit in May 2018. The group highlighted the need for a system to independently track and verify the status of blockchain for social good projects that included the following key features:

- Easy to access and navigate
- Projects are vetted prior to inclusion
- The process for vetting is transparent
- Information is updated regularly

Despite the relative nascence of blockchain applications for social impact, there has been a significant proliferation of projects. The potential of these projects is substantial, and information about them is highly varied in both quantity and quality. The Impact Ledger will weed through the white papers and the hype to offer an independent look into the state of the tech and social impact.

The Blockchain Impact Ledger aims to create a go-to resource for information about blockchain for social impact projects. For social impact organizations interested in how blockchain can support their work, the Blockchain Impact Ledger will provide real-world use cases of applications aligned with the 17 UN Social Development Goals (SDGs). Organizations will be able to find detailed, vetted information about blockchain projects working on the issues of interest to them, in the geographies in which they operate. Funders and investors will be able to survey the field and learn about both nonprofit and for-profit projects addressing both issue areas and geographies of interest. Academics and media will similarly be able to explore the greater field of blockchain for social impact as well as delve into specific areas of impact.

And the projects listed on the Blockchain Impact Ledger will gain exposure to social impact organizations, funders, investors, academics and media.

A beta version of the Blockchain Impact Ledger is scheduled to launch in May 2019. We are in the process of researching an initial batch of projects, and designing the initial version of the resource with an eye to user interface and how to scale the project effectively while managing the data.

Upon launch of the beta, we will be soliciting feedback from the greater blockchain and social good communities about how to make the resource as useful as possible.

As part of our mandate and commitment to transparency, we will continue to share information in this Medium publication about the research methodology and process. We plan to conduct additional research from within the Blockchain Impact Ledger's dataset to extrapolate information about different sectors within the social impact space as it relates to blockchain, any trends we see developing in the data, and other learnings we feel will be of value to the broader blockchain and social impact community. Watch this space for updates.

The Blockchain Impact Ledger is a research project managed by the Blockchain Trust Accelerator at New America and made possible by support from Social Alpha Foundation.

The Blockchain Trust Accelerator (BTA): The Blockchain Trust Accelerator was created in 2017 to harness blockchain technology to address social and governance challenges. Together with a coalition of technologists, government institutions, civil society organizations, and private sector partners, the BTA is developing and deploying pilots that deliver positive social impact alongside insights into how blockchain can enhance communities' resilience, accountability, efficiency, and transparency.

Social Alpha Foundation (SAF): Social Alpha Foundation is a not-for-profit, grant-making platform which focuses on supporting blockchain education and outreach to empower communities to utilize blockchain technology for social good. Founded in Hong Kong in 2017 by Nydia Zhang and Jehan Chu, SAF provides no-strings funding to companies and projects that educate communities on blockchain for social change. SAF also gives grants to non-commercial blockchain applications that focus on improving public health, education and the environment.

KIRKUS REVIEWS

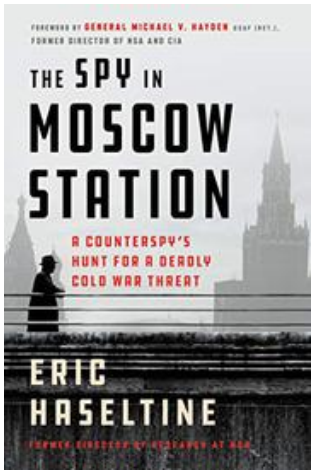
THE SPY IN MOSCOW STATION

by Eric Haseltine

A National Security Agency engineer attempts to uncover a leak in the American Embassy in Moscow in this real-life Cold War thriller.

Pub Date: April 30th, 2019
ISBN: 978-1-250-30116-1
Page count: 288pp
Publisher: Thomas Dunne Books
Review Posted Online: May 13th, 2019
https://www.kirkusreviews.com/book-reviews/eric-haseltine/the-spy-in-moscow-station/

In 1978, Gus Hathaway, the CIA chief of station at the U.S. Embassy in the Soviet capital, made an unconventional decision that was unlikely to win him either friends or approval: He asked another intelligence agency, the National Security Agency, for help. The stakes for Hathaway, though, were immeasurably high—the KGB was discovering and executing American assets, and he suspected a leak somewhere within the Moscow embassy. It was a reasonable hypothesis, as the "KGB bugging of the embassy was an accepted fact of life." Also, he knew that the KGB transmitted microwaves into the most information-sensitive areas of the building, although the CIA couldn't figure out why. To make matters worse, American operatives discovered that a chimney shaft, from which one could sometimes hear "mysterious scraping noises," wasn't connected to any actual fireplaces; it was likely a KGB listening post of some kind. Hathaway recruited the help of Charles Gandy, an engineer at the NSA who'd risen to the highest levels of civilian authority and was a ranking member of R9, a group considered the "most prestigious and glamorous at NSA." Haseltine (*Brain Safari*, 2018, etc.), with all the painstaking scrupulousness of an investigative journalist, details Gandy's remarkable efforts to produce a "smoking gun" that could prove the Soviets were spying on the embassy—evidence that could justify a complex countermission that he himself had designed.



Haseltine is a former director of research for the NSA—his boss there, Gen. Michael V. Hayden, contributes a foreword—and his expertise is beyond reproach. His research here is breathtaking, drawing on a bevy of sources, including his own interviews with Gandy as well as declassified U.S. governmental documents, often reproduced here at great length. In fact, his thoroughness can be a bit overwhelming at times; readers will often find themselves buried under mounds of minute detail, much of it forbiddingly technical. Even so, the story as a whole has all the power and intrigue of a cinematic thriller. In one memorable scene, for instance, Gandy was visited in his working quarters at the embassy by a "KGB honey trap," a beautiful woman who attempted to gain access to his room; no one could figure out how she—and her male escort—managed to make it past embassy guards. The story isn't only about the contest between Americans and Russians, but also about the turf-war rivalry of the CIA and the NSA. One declassified CIA memorandum, in shockingly explicit terms, notes the "NSA's new feeling of importance" and its "ceaseless effort to assert itself more vigorously in the intelligence process." Gandy, in particular, emerges as a captivatingly complicated figure—endlessly motivated to defeat his adversaries but also impressed by their ingenuity. The book ends with provocative reflections on what Americans can learn from the Russians about espionage today and on interagency cooperation.

An immersive, dramatic, and historically edifying work.

Blockchain Access Privacy: Challenges and Directions

Ryan Henry, Amir Herzberg, and Aniket Kate

Abstract—Privacy, facilitated by a confluence of cryptography and decentralization, is one of the primary motivations for the adoption of cryptocurrencies like Bitcoin. Alas, Bitcoin's privacy promise has proven illusory and, despite growing interest in privacy-centric blockchains, most blockchain users remain susceptible to privacy attacks that exploit network-layer information and access patterns which leak as users interact with blockchains.

Understanding if and how blockchain-based applications can provide strong privacy guarantees is a matter of increasing urgency. Many researchers advocate using anonymous communications networks, e.g., Tor, to ensure access privacy. We challenge this approach, showing the need for mechanisms through which non-anonymous users can (i) publish transactions that cannot be linked to their network addresses or to their other transactions, and (ii) fetch details of specific transactions without revealing which transactions they seek. We hope this article inspires blockchain researchers to think 'beyond Tor' and tackle these important access privacy problems head-on.

1 Introduction and Motivation

A *blockchain* is a distributed, append-only log of time-stamped records that is cryptographically protected from tampering and revision. In the eight years since blockchains were first proposed, their use as publicly accessible and verifiable ledgers for online financial transactions has become widespread. This rapid adoption has largely been spurred by the success of *Bitcoin*[1], a digital currency that—owing to its decentralized and pseudonymous nature, support for complex financial instruments (enabled by a powerful, built-in scripting language), and capacity to facilitate fast and inexpensive transactions across the globe—has proven to be a highly disruptive force in the finance and e-commerce sectors.

As Bitcoin and alternatives like *Ethereum*[2] and *Ripple*[3] continue to mature and grow in market value, it is becoming increasingly likely that blockchains as a means to facilitate financial transactions are here to stay. Yet blockchains represent far more than a mere monetary innovation; researchers and industry members alike are only just beginning to understand the true potential of blockchain-based distributed ledgers, with their strong integrity and availability guarantees and their ability to leverage community consensus to eschew centralized trusted curation. Indeed, beyond the sorts of payment transactions for which blockchains are already widely deployed, potential applications for blockchains abound in areas as diverse as electronic voting, certificate authorities, the Internet of Things, and smart systems. Moreover, the past few years were marked by announcements from numerous companies—ranging from startups like *R3*[4] to established technology firms like IBM, and financial institutions like Visa—about forthcoming products based on innovative blockchain designs that are specially tailored to meet organizational and business logic needs. The target applications for these products range from payment settlement through supply-chain management and beyond.

Just how private are today's blockchains? The ephemeral nature of users' pseudonymous identities in Bitcoin played a key role in its early success. However, eight years of intense scrutiny by privacy researchers has brought to bear an arsenal of powerful heuristics using which attackers can effectively link disparate Bitcoin transactions to a common user and, in many cases, to that user's real-world identity. Ultimately, instead of providing the bastion of privacy for financial transactions that its early adopters envisioned, Bitcoin and its altcoin brethren are in many ways *less* private than traditional banking, where government regulations mandate basic privacy protections. In an attempt to address this situation, the cryptography and privacy research communities have proposed and implemented several protocols aiming to improve blockchain privacy. These protocols all try to decouple users' pseudonymous identities from the specific transactions they make, thereby frustrating attempts to link transacting parties *based on data that appears in the blockchain*. However, none of the proposed protocols attempts to hide the identities of users from network-level adversaries *as the users publish or retrieve data from the blockchain*. Instead, the proposed protocols 'outsource' this crucial step, relying on an external anonymous communications network such as *Tor*[5]. However, running complex protocols over general-purpose, low-latency anonymity networks such as Tor is fraught with risks, and can expose users to subtle-yet-devastating deanonymization attacks, thereby undermining the privacy guarantees of the entire blockchain system. We can do better!

2 Cryptography to the Rescue?

Most blockchains are, at their core, massively distributed and publicly accessible databases; therefore, beyond ensuring that the *data* they store do not, in and of themselves, betray user privacy, any research program that seeks to fully address blockchain privacy must additionally consider (at the very least) privacy for two

- *Ryan Henry is with Indiana University Bloomington.*
- *Amir Herzberg is with Bar Ilan University.*
- *Aniket Kate is with Purdue University.*

[1] https://www.bitcoin.org/
[2] https://www.ethereum.org/
[3] https://ripple.com/
[4] https://www.r3.com/
[5] https://www.torproject.org/

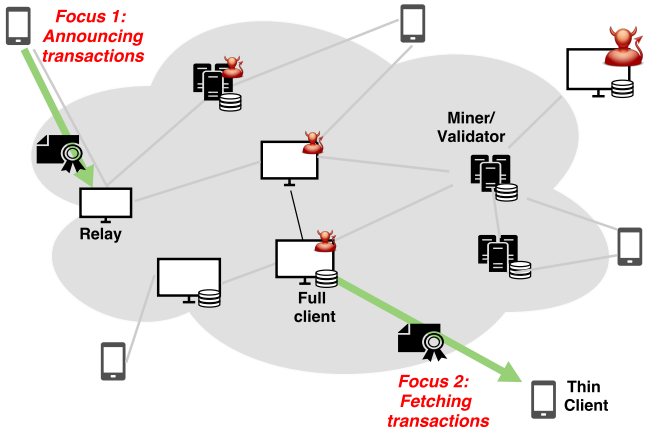


Fig. 1. Topology of a typical blockchain system. The two bold arrows (highlighted in **green**) illustrate sensitive information flows that must be protected in order to prevent attackers from leveraging network-level information to compromise the privacy of blockchain users.

fundamental types of transactions: *reading data from* and *writing data to* a blockchain.

In the context of cryptocurrencies like Bitcoin, the database represented by the blockchain is a publicly accessible and verifiable *ledger* of financial transactions. Specifically, whenever a transaction occurs, the originating party publicly *announces* the transaction to a handful of selected entities, who then spread the details of that transaction throughout the network via a gossip protocol. The transaction is ultimately aggregated with several other (unrelated) transactions into a discrete *block*, which then gets irreversibly appended to a *chain* comprising all earlier blocks. The chain of blocks can—indeed, to obtain strong integrity and availability, *must*—be replicated and shared in its entirety among many nodes in a network, thereby providing each node with a global, eventually consistent view of every transaction that has ever taken place. New transactions are reflected in all replicas of the blockchain within some predefined expected time, which can range from a few seconds (e.g., in Ripple) to a few minutes (e.g., in Bitcoin).

Each transaction is associated with a pair of *pseudonyms* (often called *wallets*), respectively identifying the sender and receiver of some digital assets. Users can generate new pseudonymous wallets with which to receive digital assets arbitrarily and at will; indeed, it is considered a best practice for Bitcoin users to generate a fresh, ephemeral wallet whenever they wish to conduct a new transaction. The primary motivation for generating such ephemeral wallets is to protect user privacy by making it difficult for an attacker to link together the various transactions involving a given user by simply examining the sender and receiver pseudonyms appearing in transactions recorded in the ledger. However, as Bitcoin and related altcoins grow ever-more prevalent, there is a growing concern that the "privacy" offered by this approach is illusory at best. Indeed, as mentioned previously, the past eight years of research into blockchain privacy has given rise to a veritable treasure trove of effective heuristics using which attackers can link Bitcoin transactions back to a common user, despite the widespread use of ephemeral wallets [1]–[3].

Figure 1 depicts a traditional blockchain architecture. (We use the qualifier "traditional" here to differentiate the blockchain architectures we consider from those involving *payment channels* and other layer-2 applications, which introduce a host of new privacy concerns that go beyond the scope of this article.) For the purposes of this article, we focus on the two arrows that are **bolded** and highlighted in **green**; specifically, we focus on the need for innovative mechanisms that allow users to

(i) *announce and publish transactions anonymously*, a task for which we envision a tailor-made anonymity mechanism that is integrated directly into the blockchain architecture; and to

(ii) *fetch transactions privately*, a task for which we envision using special private information retrieval (PIR) protocols designed and optimized to support efficient and expressive queries for transactions stored in a blockchain.

We note that a handful of second-generation altcoins—including *Zcash*[6] and *Monero*[7]—natively employ cryptographic techniques to prevent the *contents* of transaction on the blockchain from leaking private information about transacting parties. Likewise, the research literature contains several proposals (a selection of which we summarize in the next subsection) that aim to provide similar transaction privacy atop the deployed Bitcoin, Ripple, and Ethereum blockchains. While such approaches are indeed effective at protecting blockchain users against a subset of the deanonymization heuristics that plague mainstream deployed blockchains, we emphasize that the existing approaches, so far, focus on preventing the *data* stored in a blockchain from leaking private information—they do nothing significant to mitigate against inferences that leverage *network-level information* (e.g., IP addresses) or *access patterns* (e.g., specific blocks or portions thereof) revealed when users interact with the blockchain data. As such, the existing proposals all fall far short of solving the blockchain privacy problem in its entirety.

2.1 Existing protocols for transaction privacy

As the insufficiency of ephemeral pseudonyms became apparent to the Bitcoin community, a proposal called *CoinJoin* emerged as a potential solution. In CoinJoin, users route their transactions through a centralized *mixing service* (sometimes called a *tumbler*), which serves to obscure the relationships between the senders and receivers of those transactions before they are posted to the ledger. However, such centralized mixing services introduce a single point of trust and failure; indeed, the mixing service always knows the link between the sender and receiver of each transaction and, perhaps more troublingly, there is nothing to stop the mixing service from stealing assets that users try to route through it. A series of progressively more sophisticated protocols have been proposed to address CoinJoin's limitations.

The first improvement was *Mixcoin*, which attempts to mitigate the risk of theft by holding the mixing service "accountable" if it steals a user's assets (though theft is still technically possible and the mixing service still learns who is transacting with whom). Building on a series of incremental improvements to this basic idea (including *BlindCoin* and *Blindly Signed Contracts*), a proposal called *TumbleBit* [4] finally addressed the accountability and anonymity weaknesses of Mixcoin in a manner fully compatible with Bitcoin; however, the TumbleBit approach requires upwards of 20 minutes (i.e., two Bitcoin block) per transaction on average and introduces additional transaction fees. The third author's own *CoinShuffle* and *CoinShuffle++* [5] take a different approach, having users perform a special multi-party computation among themselves so that no third-party mixing service is necessary.

[6] https://z.cash/ [7] https://getmonero.org/

The emerging privacy-centric cryptocurrencies, such as Zcash and Monero, employ cryptographic primitives such as zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK), traceable ring signatures, confidential transactions and stealth addresses to offer significantly better privacy properties than those possible for Bitcoin transactions.

2.2 Inadequacy of existing proposals

The above transaction privacy protocols all aim to sever the link between senders and receivers as recorded in the transactions that get published to the blockchain. However, the approaches are all susceptible to attacks that reestablish links between transacting parties using network-level information and/or access patterns observed both as users announce their transactions and as they probe the blockchain to learn which of their transactions have posted to the ledger [2]. For example, an attacker who observes that a given user visits a website immediately before that website receives a donation via Zcash or Monero might surmise that the user made the donation; moreover, the attacker can all-but-confirm this suspicion if it later observes the same user checking whether the transaction in question has posted to the ledger.

To define away this elephant in the room, the developers of such privacy protocols typically assume that users communicate over an anonymous-communication protocol such as Tor; in fact, some privacy-centric altcoins—like Zcash, Anoncoin, and Torcoin—include native support for Tor and expect that all users interact with their blockchains *exclusively* through Tor. As an example, the Zcash website[8] clearly states (and we quote) that "a unique IP address can allow network observers to correlate your Zcash transactions with each other and with your other traffic" to which it adds that "advanced users may opt to connect through Tor to obfuscate their node's IP address, however, further exploration is needed on a vulnerability combining Bitcoin's Denial of Service mitigation (inherited into Zcash) and anonymous communication networks like Tor before we can recommend users who are not familiar with the attack to route their Zcash nodes through Tor."

This dependency on Tor for anonymity introduces some rarely-acknowledged-yet-undeniably-troubling weaknesses. One source of weakness stems from the fact that Tor is specifically designed to support *low-latency* communication, such as interactive web browsing and real-time instant messaging; indeed, it seems inherent (and real-world attacks seem to confirm) that such low-latency low-bandwidth anonymous communication systems can provide at most a relatively weak form of anonymity compared to high-latency approaches like Chaumian mix networks or high-bandwidth approaches like dining-cryptographers (DC) networks. Indeed, a recent paper by Das et al. [6] analyzed the so-called "anonymity trilemma" and concluded that, in the presence of a global passive (network-level) adversary, anonymous communications networks can hope to provide just two of three desirable properties: strong anonymity, low bandwidth overhead, and low latency overhead. Fortunately, because financial transactions are naturally able to tolerate moderate latency—indeed, so-called "permissionless blockchains", like the one used in Bitcoin, already impose latencies on the order of *several minutes* even without the use of an anonymous communications network—users need not settle for the relatively weak anonymity guarantees that low-latency systems like Tor can provide.

[8] https://z.cash/support/security/privacy-security-recommendations.html

Further, Biryukov and Pustogarov [7] demonstrated how Bitcoin's "blacklisting" measures may ultimately leave users conducting Bitcoin transactions over Tor *more* vulnerable to active deanonymization attacks than those announcing their transaction non-anonymously. They describe man-in-the-middle attacks that exploit the Bitcoin network's built-in reputation-based DoS protection mechanism to force specific Bitcoin peers to ban Tor exit relays of the attacker's choice, thus forcing *all* Bitcoin traffic to exit the Tor network through a small set of attacker-controlled relays. Once in this privileged position, the attacker can launch several troubling privacy attacks, including deanonymization via traffic correlation (which is made easier because the attacker automatically controls one end of the communication), correlating multiple wallet addresses to a common user, and launching "double-spending" attacks by lying to thin clients about previous transactions involving a given wallet address.

Yet another problem arises from the fact that Tor is often blocked by IT departments within organizations or even subject to state-level censorship by authoritarian governments. This has direct negative consequences for the privacy of users connecting from such organizations or countries, even though the censorship is almost certainly intended to quell some other, unrelated usage of Tor. As a workaround for such censorship, Tor ships with support for some censorship-evasion techniques including *Tor bridges* and *pluggable transports*; however, the effectiveness of these mechanisms is far from perfect and censorship events continue to affect Tor users. In general, it seems unwise to advocate the wholesale use of censorship circumvention tools for activities that are typically not subject to censorship.

Moreover, a third-party anonymous communication network such as Tor may not be willing or able to support blockchain traffic on a large scale. A dual concern is some blockchain systems may be hesitant to use Tor since Tor has also been used for nefarious purposes, ranging from ransomware and botnet command and control through to child pornography. As an anecdotal example of this, the third author has learned through communications with developers at Ripple that, despite being very keen on improving privacy for their clients, Ripple's developers are unwilling to leverage a Tor-based solution to do so.

Finally, due to their decentralized design, blockchain systems seem like prime candidates for fulfilling their own anonymity and privacy needs, avoiding the dependency on external services and providing performance and privacy/anonymity guarantees tailored to their own needs.

In short, we believe that effective blockchain privacy necessitates rethinking the one-size-fits-all approach of using external anonymous communications infrastructures to solve all problems requiring anonymity. Although anonymity does indeed love company, mixing two dissimilar types of traffic together does not necessarily improve anonymity for either type and, if not done very carefully and correctly, may in fact provide weaker anonymity than protecting each type of traffic with its own tailor-made solution.

3 Publishing Transactions Anonymously

By their very design, blockchain systems require extensive overlay networks through which participants announce transactions and agree on what transactions should ultimately appear on the blockchain. Thus, it seems natural to leverage the existing overlay structure to realize anonymous transaction publishing,

rather than relying on an external service like Tor. We propose that blockchain privacy protocols should de-link users' network-level information from their transactions using mechanisms that piggybacks on the overlay network that is already in place for announcing transactions. The specifics of how such a mechanism might work vary, depending on the structure of the overlay network imposed by the *consensus protocol*—that is, depending on how participants decide which transactions qualify for inclusion in the blockchain.

Permissionless versus permissioned blockchains. Proposed and deployed blockchains fall into two distinct categories based on the mechanism they use to build a consensus around what data to immortalize in the blockchain: *permissionless blockchains* and *permissioned blockchains*.

The blockchains underlying Bitcoin and Ethereum constitute two prominent examples of *permissionless blockchains*. As their name implies, permissionless blockchains place no restrictions on who participates in the consensus process. Instead, unrestricted entities called *miners* collectively decide which blocks should be appended to the chain by providing an associated proof of work. In the case of Bitcoin, this proof of work takes the form of a "partial hash inversion", wherein the miners seek inputs that lead a cryptographic hash function to produce a digest whose numerical value does not exceed some global-parameter target. Such a permissionless consensus guarantees that only valid blocks get appended to the blockchain (approximately) under the assumption that more than half of all mining resources in the network are controlled by honest—or, at least, non-colluding—entities.

The blockchains underlying Ripple and the Linux Foundation's *Hyperledger*[9] are two prominent examples of *permissioned blockchains*. In contrast to permissionless blockchains, permissioned blockchains do place restrictions on who participates in the consensus process. A group of highly available entities (with strong identities) collectively decide which blocks should be appended to the chain by leveraging a Byzantine fault-tolerant atomic broadcast protocol. This approach allows permissioned blockchains to reach consensus very rapidly, requiring as little as a few seconds for each transaction to be reflected in the ledger.

The contrasting security assumptions and efficiency guarantees of permissionless and permissioned blockchains make them well suited to different use cases and, indeed, the two varieties are prospering together: traditionally structured organizations/consortiums are increasingly adopting permissioned blockchains, while peer-to-peer solutions continue to leverage permissionless blockchains.

3.1 Publishing to permissionless blockchains

Permissionless blockchain systems (like Bitcoin and Ethereum) employ peer-to-peer (P2P) networks of relays to propagate transactions and blockchain updates throughout the network using a best-effort gossip protocol. Such P2P networks typically experience considerable churn, with relays joining, leaving, and rejoining the network at will; however, the average number of relays in the network at any given time can remain relatively high. For example, at the time of writing, the number of online relays in the Bitcoin network at any given time is about one-and-a-half times the number of Tor relays. (As of October 4, 2017, *Tor Metrics*[10] estimates about 6700 Tor relays versus the *Bitnode*[11] estimate of about 9600 full Bitcoin nodes.) One might, therefore, consider employing the elaborate Bitcoin communication infrastructure toward improving the anonymity of users' announcements. Given the P2P nature of the network, we believe it may be possible to leverage the existing academic research on P2P anonymous communications networks. For instance, such a solution could be based upon Pisces [8], employing the social trust links to construct anonymous communication paths that are robust to compromise in the presence of route-capture attacks and Sybil nodes. However, given the dynamic and open nature of permissionless blockchains such as Bitcoin, establishing trust in relays will be a prominent challenge.

The *Kovri project*[12], an offshoot of the Monero and the Bitcoin developers' recent interest in the Dandelion networking policies [9], clearly indicate the blockchain community's awareness of the problem; nevertheless, significantly efforts are necessary going forward. In general, it will be an interesting challenge to analyze and establish security, privacy, and viability of realizing a P2P anonymous communications system over permissionless blockchain systems.

3.2 Publishing to permissioned blockchains

Permissioned blockchain systems (like Ripple, *Corda*[13], and Hyperledger) employ a clique of highly available validator nodes for agreeing on transactions and blocks. These nodes employ traditional asynchronous Byzantine-tolerant consensus protocols to append a block of transactions to the blockchain. Here, validators select valid transactions to be agreed upon from those transactions forwarded by the users of the system. As typically transactions from several users are added to any given block, a simple approach to provide anonymity here will be to perform all the communication between users and validators over an anonymous communications network. However, we advocate improving efficiency and reducing the overhead by combining the consensus process for agreeing on transactions with the process of mixing users' announcements.

This problem can be modeled as an asynchronous multi-party computation (AMPC) problem, and can be solved using the generic AMPC techniques; however, we propose development of tailored solutions to further improve the efficiency. A possible tailored approach for agreeing on a randomly permuted set of transactions can involve combining Newton's identity method for power sums (as employed by Ruffing et al. [5]) with asynchronous verifiable secret sharing and asynchronous Byzantine consensus. Nevertheless, a key challenge will be to make these solutions scale well (possibly sublinearly) with the number of mixed transactions.

4 Fetching Transactions Privately

Blockchains differ from traditional databases in their use of cryptography as a means to eschew both centralization and trusted curators, all the while ensuring strong resistance to "tampering" (i.e., history rewriting). Yet this remarkable combination of attributes is guaranteed only for users that hold a complete local replica of the blockchain. With a blockchain currently over 100 GB and growing,

[9] https://www.hyperledger.org/

[10] https://metrics.torproject.org/

[11] https://bitnodes.21.co/

[12] https://getkovri.org

[13] https://www.corda.net/

this local-storage requirement is quickly becoming infeasible for casual Bitcoin users; as a result, many such users now employ so-called *thin clients*, which bypass the need to hold a local copy of the blockchain by forwarding blockchain queries to semi-trusted intermediaries.

Specifically, thin clients run in what is called *Simplified Payment Verification (SPV) mode*—so named after the section of the original Bitcoin whitepaper [10] that details it—wherein the initial syncing process connects to an arbitrary *full node* and downloads only the block headers (each of which includes a Merkle root committing to the actual block). The thin client then verifies that the given headers indeed form a blockchain (with sufficient difficulty value), after which they can request the details of transactions matching certain patterns (e.g., payments to or from particular addresses) from any full node. The full nodes reply to such requests with a copy of any relevant transactions together with Merkle branches linking those transactions to their associated block headers. This process exploits the Merkle tree structure to allow proofs of inclusion in a block without needing to provide the thin client with the full contents of the block.

The SPV approach has the distinct advantage that the cost of initial syncing scales linearly with the length of the blockchain (about 80 bytes per header, or 4.2 MB per year) and is independent of the size of the actual blocks. However, a naive implementation of SPV exposes thin clients to potentially devastating attacks on privacy. As a thin client will typically request details about precisely those transactions that correspond to keys it owns, it may end up revealing to the full node a complete list of its public addresses. In particular, *Bitcoin users that rely upon such thin clients are subject to deanonymization*. This is a serious risk; there have been numerous reports of high-rolling Bitcoin users being identified and targeted by miscreants to steal their digital fortunes.[14]

A tempting response is to route thin-client queries through an anonymity network like Tor; however, this leaves clients susceptible to low-cost deanonymization and double-spending attacks [7]. Indeed, the root problem for thin clients is not a lack of anonymity for the *querier* but, rather, a lack of privacy for the *queries*—anonymity, quite simply, solves the wrong problem.

Instead, we observe that the problem of realizing private blockchain queries is imminently solvable using a well-known cryptographic primitive called *private information retrieval (PIR)*. PIR is a cryptographic primitive that solves the seemingly impossible problem of letting clients query a remote database, while not exposing the clients' query terms or the responses they generate to the database operator. PIR has received considerable attention from the cryptography, privacy, and theoretical computer science research communities. Alas, despite a series of significant advances over the past two decades, existing PIR techniques are notoriously inefficient and, consequently, to date not one of the numerous PIR-based applications proposed in the research literature has been deployed at-scale to protect the privacy of users "in the wild".

As a result, transitioning the idea of using PIR to fetch blockchain transactions privately into practice still necessitates some basic research and rather substantial engineering and implementation efforts. Fortunately, some recent advances in PIR research yield the promise of PIR protocols that are sufficiently practical to deploy on databases of size commensurate with Bitcoin's blockchain.

[14] https://bitcointalk.org/index.php?topic=16457.0

4.1 Private blockchain queries from PIR

The key goals here are to create protocols that enable thin clients to (i) determine if particular transactions are reflected in the blockchain (and, if so, how many blocks have been appended since, a rough proxy for the computational effort that would be required to "undo" that transaction), and (ii) find out the balances associated with a set of public keys, reflecting all transactions that have occurred so far involving those keys.

This will involve defining appropriate data structures that lend themselves to being queried via PIR, as well as efficient mechanisms for keeping those data structures up to date as the blockchain grows. Although one could conceptually employ any PIR protocol for this purpose, thinking towards mass adoption among the millions of present and potential Bitcoin users, we suggest very strict requirements on acceptable communication and computation overhead. In effect, the target will be communication costs that are reasonable for a smart phone communicating over a mobile data connection, and computation costs low enough for a modestly equipped server to process tens or hundreds of queries every second. Such strict requirements preclude most existing PIR protocols; however, the recent introductions of (i) *distributed point functions* [11], (ii) Intel's *software guard extensions* (*SGX*) architecture[15], and (iii) the first author's *indexes of queries* [12] provide three very elegant—and, we believe, highly practical—ways to realize the kinds of PIR-based private blockchain queries we envision. Each approach brings its own performance characteristics and its own security assumptions, ranging from non-collusion, through computational assumptions, to trusted hardware. The research objective here will be to devise appropriate data structures to facilitate PIR-based queries over blockchain data, and then to implement and evaluate the suitability of the various approaches.

Moreover, by leveraging the anonymous communications framework we advocated in Section 3, it may be possible to realize lower-cost relaxations of information-theoretic PIR that satisfies a differentially private notion for private queries [13].

5 Concluding Remarks

General-purposes anonymous communications systems like Tor are not a panacea for communication privacy issues. Indeed, not all applications are anonymized equally well by low-latency anonymity networks, and not all privacy problems are adequately addressed by making users anonymous. In this article, we highlighted two prominent communication privacy issues that afflict current blockchain solutions: the problems of announcing blockchain transaction anonymously and fetching blockchain transactions privately. We proposed research directions that shift from the current norm of just saying 'do it over Tor' and instead seek to tackle these important problems head-on. In particular, for the problem of announcing blockchain transaction anonymously, we suggested to leverage blockchain consensus infrastructures instead of the external, general-purposes networks like Tor, while for the problem of fetching transaction privately, we offered directions towards making private information retrieval (PIR) schemes suitable and efficient for blockchain transactions.

While we only considered ways to address privacy challenges arising from network-level and access pattern leakage on traditional blockchains, new blockchain extensions—such as the

[15] https://software.intel.com/sgx

lightning network[16], which has been recently proposed as a way to greatly improve the scalability of permissionless blockchains—introduce new subtle privacy challenges that will also require novel solutions. Although some solutions are already emerging towards improving privacy in these path-based transactions [14], [15], it is an interesting open challenge to devise scalable mechanisms for performing (multi-hop) payment-channel transactions *privately* against a network-level adversary.

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant Numbers 1718595 and 1719196, and by United States-Israel Binational Science Foundation (BSF) under Grant Number 2016718.

References

[1] 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," *Communications of the ACM*, vol. 59, no. 4, pp. 86–93, 2016. Available: https://doi.org/10.1145/2896384

[2] P. Koshy, D. Koshy, and P. D. McDaniel, "An analysis of anonymity in Bitcoin using P2P network traffic," in *Proceedings of FC 2014*, ser. LNCS, vol. 8437, Christ Church, Barbados, March 2014, pp. 469–485. Available: https://doi.org/10.1007/978-3-662-45472-5_30

[3] E. Androulaki, G. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in Bitcoin," in *Proceedings of FC 2013*, ser. LNCS, vol. 7859, Okinawa, Japan, April 2013, pp. 34–51. Available: https://doi.org/10.1007/978-3-642-39884-1_4

[4] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumbleBit: An untrusted Bitcoin-compatible anonymous payment hub," in *Proceedings of NDSS 2017*, San Diego, CA, USA, February–March 2017. Available: https://doi.org/10.14722/ndss.2017.23086

[5] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "P2P mixing and unlinkable Bitcoin transactions," in *Proceedings of NDSS 2017*, San Diego, CA, USA, February–March 2017. Available: https://doi.org/10.14722/ndss.2017.23415

[6] D. Das, S. Meiser, E. Mohammadi, and A. Kate, "Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency—choose two," Proceedings of IEEE S&P 2018 (to appear), San Francisco, CA, USA, May 2018. Available: https://eprint.iacr.org/2017/954

[7] A. Biryukov and I. Pustogarov, "Bitcoin over Tor isn't a good idea," in *Proceedings of IEEE S&P 2015*, San Jose, CA, USA, May 2015, pp. 122–134. Available: https://doi.org/10.1109/sp.2015.15

[8] P. Mittal, M. K. Wright, and N. Borisov, "Pisces: Anonymous communication using social networks," in *Proceedings of NDSS 2013*, San Diego, CA, USA, February 2013. Available: https://arxiv.org/abs/1208.6326

[9] S. Bojja Venkatakrishnan, G. Fanti, and P. Viswanath, "Dandelion: Redesigning the bitcoin network for anonymity," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, pp. 22:1–22:34, June 2017. Available: http://doi.acm.org/10.1145/3084459

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org, Tech. Rep., November 2008. Available: https://bitcoin.org/bitcoin.pdf

[11] N. Gilboa and Y. Ishai, "Distributed point functions and their applications," in *Proceedings of EUROCRYPT 2014*, ser. LNCS, vol. 8441, Copenhagen, Denmark, May 2014, pp. 640–658. Available: https://doi.org/10.1007/978-3-642-55220-5_35

[12] S. M. Hafiz and R. Henry, "Querying for queries: Indexes of queries for efficient and expressive IT-PIR," in *Proceedings of CCS 2017*, Dallas, TX, USA, October–November 2017, pp. 1361–1373. Available: https://doi.org/10.1145/3133956.3134008

[13] R. R. Toledo, G. Danezis, and I. Goldberg, "Lower-cost ϵ-private information retrieval," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2016(4), Darmstadt, Germany, October 2016, pp. 184–201. Available: https://doi.org/10.1515/popets-2016-0035

[14] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proceedings of CCS 2017*, October–November 2017, pp. 455–471. Available: https://doi.org/10.1145/3133956.3134096

[16] https://lightning.network/

[15] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *Proceedings of CCS 2017*, October–November 2017, pp. 473–489. Available: http://doi.acm.org/10.1145/3133956.3134093

Genie: A Secure, Transparent Sharing and Services Platform for Genetic and Health Data

Shifa Zhang†, Anne Kim*, Dianbo Liu*, Sandeep C. Nuckchady†, Lauren Huang†, Aditya Masurkar†, Jingwei Zhang†, Law
Pratheek Karnati‡, Laura Martinez§, Thomas Hardjono*, Manolis Kellis*, Zhizhuo Zhang*
* Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts 02142–1308
Email: {anneykim, dianbo,hardjono , manoli,zhizhuo}@mit.edu
† Secure AI Labs, Cambridge, Massachusetts 02142–1308
Email: {shifa, sandeep, lauren, aditya, jingwei, larry}@secureailabs.io
‡ IBM, 4205 S Miami Blvd, Durham, NC, 27703-9141
Email: karnatip@us.ibm.com
§ Intel Software Guard Extensions, Intel 2200 Mission College Blvd, Santa Clara, CA 95054
Email: laura.martinez@intel.com

***Abstract*—Artificial Intelligence (AI) incorporating genetic and medical information have been applied in disease risk prediction, unveiling disease mechanism, and advancing therapeutics. However, AI training relies on highly sensitive and private data which significantly limit their applications and robustness evaluation. Moreover, the data access management after sharing across organization heavily relies on legal restriction, and there is no guarantee in preventing data leaking after sharing.**

Here, we present Genie, a secure AI platform which allows AI models to be trained on medical data securely. The platform combines the security of Intel Software Guarded eXtensions (SGX), transparency of blockchain technology, and verifiability of open algorithms and source codes. Genie shares insights of genetic and medical data without exposing anyone's raw data. All data is instantly encrypted upon upload and contributed to the models that the user chooses. The usage of the model and the value generated from the genetic and health data will be tracked via a blockchain, giving the data transparent and immutable ownership.

I. INTRODUCTION

A. Background

Genomics-based personalized medicine began more than ten years ago [6]. Genetic big data has shown promise in conducting breast cancer studies, building the cancer genome atlas (TCGA), and improving screening and diagnosis [37]. Many recent studies have prospective results with advanced machine learning and artificial intelligence (AI) technologies on genotypic and phenotypic big data [23], [9], [38], [15], [25], [17]. Using large amounts of federated genetic and medical data to train AI models and using these models to predict diseases, drug responses, and personality traits will allow for great advancements benefiting human health.

At the same time, the amount of data is growing very fast. Th DNA sequencing has become cheaper, better, and faster in recent years [21], [33]. The Electronic Health Record (EHR) systems are more wildly adopted and generating huge amount of data. However, regulations of both the Health Insurance Portability Accountability Act (HIPAA) in U.S. and General Data Protection Regulation (GDPR) [12] in EU require a strict protection on the private information in the data. Most of the medical data are no accessible to many promising health care AI algorithms because of privacy protection regulations. There is a great need for a secure AI platform for AI models to process on sensitive medical data.

B. Challenges

The central problem we tackle is how to protect private information and preserve data ownership while sharing information derived from the data in an open, transparent online environment. Data are easily copied when shared. Once the data are copied, ownerships of the data are eroded. Furthermore, there is no way to track data accesses and modifications for those copied data.

Work by Hardjono, Shrier, and Pentland [35] on the *open algorithms* (OPAL) paradigm points to the need for the sharing of data and insights in a privacy-preserving manner. Additionally, personal data is now recognized as a new asset class [40], which introduces the need for individuals to have the ability to consent to their data being used in computations [12]. There is a clear need for a system that can respect a person's rights to their genetic and health data in order for that data to be accessible to others.

A centralized database could have security facilities to provide a secure environment for data users to access the data without compromising privacy, but these databases are isolated systems largely incompatible with each other, vulnerable to attacks from insiders, and challenging to track once data has been copied to external locations. Therefore, a centralized solution is insufficient.

In this paper, we introduce the *Genie* (an acronym for Genetic data Exploration by blockchaiN Interconnected Encryption) platform which is an open, distributed, transparent, and secure marketplace to provide high quality genetic and phenotypic big data, AI models, and a secure computation platform, accelerating AI advancing in health care.

II. Existing studies

A. Homomorphic Encryption

Kim and Lauter introduced private genome analysis through homomorphic encryption [20]. Homomorphic encryption allows for computation on encrypted data without needing decryption. The result of the computation is encrypted and can only be decrypted with the same key used to encrypt the input data. It is possible to do statistics or AI model training on homomorphically encrypted user data without decryption. Only computational results are decrypted. This approach protects private information contained in the raw genetic and phenotypic data.

Homomorphic computation was first raised in 1978 [29], but there was little progress until Gentry published his thesis about a fully homomorphic encryption scheme [14]. In 2012, Fan and Vercauteren published an improved homomorphic encryption scheme (FV) based on Gentry's scheme [13], bringing it a step closer to real applications. Some open source homomorphic encryption libraries have been developed based on the FV scheme, such as Microsoft SEAL [8].

Even though the performance of homomorphic encryption and computation has improved significantly in recent years, it is still too expensive to do useful computation for the purposes of genotypic and phenotypic big data analysis, based on performance data from Bajard, et al. [2].

There are some other issues with the homomorphic encryption for private data sharing. First, If allowing arbitrary computations (normally required by AI training) on the encrypted data, it can infer raw data from the computation results. Second, it is difficult to use data from multiple data owners because homomorphic computation requires that all input data are encrypted with the same encryption key.

B. Hybrid Homomorphic Encryption and Intel SGX [19]

The *Secure gwAs in Federated Environment Through a hYbrid solution with Intel SGX and Homomorphic Encryption* (SAFETY) framework with hybrid homomorphic encryption and Intel Software Guard Extensions (SGX) was proposed for genome-wide association study (GWAS) research in 2017 by Sadat, et al. [32]. The framework uses homomorphic encryption to encrypt a data owner's data and do basic statistics with homomorphic computation on the encrypted data. Afterwards, the statistics results are sent into SGX private regions of memory, called enclaves, to be decrypted for further computation. Researchers using the enclave can query the results in the enclave. This hybrid framework can get higher performance than pure homomorphic computation because time-consuming multiplying operations can be done in the enclave with the unencrypted data. Because computations on the raw data occur only inside the enclave, this approach still protects the privacy of data owners.

However, if the data user can program the software in the enclave, it is still possible to reverse the computation on the homomorphically encrypted data and extract the raw data. If the user cannot program the software in the enclave, it limits the extent of analysis that can be done on the data.

In the SAFETY framework, data usage is not tracked. Therefore, data owners cannot be compensated for data usage.

C. Blockchain for Health Care

Kuo, Kim, and Machado introduced blockchain distributed ledger technologies for biomedical and healthcare applications [22]. The blockchain has the advantages of being distributed, robust, tamper-resistant and transparent compared with traditional relational databases. Using the blockchain in biomedical and healthcare fields bring the benefits of improved medical record management, enhanced insurance claim processes, accelerated clinical/biomedical research, and advanced biomedical/healthcare ledgers. Disadvantages of the blockchain in these fields include too much transparency when handling confidential information, restrictions on speed and scalability, and possible >50% malicious attacks [22].

Linn and Koo introduced blockchain for health data [24]. The authors pointed out that blockchain technology addresses interoperability challenges in health data management. Blockchain is based on open standards and is widely accepted.

III. Primary Principles

This section introduces the primary principles of the Genie platform: privacy-protected and data ownership-preserved data sharing with open algorithm/open source code, Intel Software Guarded eXtensions (SGX), and the blockchain technologies.

A. Open Algorithm and Source Code

The two techniques that *open algorithms* (OPAL) [35] use to protect privacy in shared data are as follows:

1) Algorithms are sent to secure and trusted data storages to be evaluated directly on the data instead of sending the private data somewhere else to be processed. The algorithms are publicly inspected and share only results that will never compromise the raw data.
2) The algorithms and each evaluation of the algorithms are logged in an immutable database, such as a blockchain.

In the Genie platform, open source code is an important aspect of the OPAL paradigm. The source code of the distributed application (Dapp), SGX enclave, and AI model evaluation algorithms are all openly shared online. The cryptographic hashes of these open source codes are registered on the blockchain and can be used by anyone to verify that they have a copy of the original source code.

Open source software has become very popular, 20 years since the book *Open sources: Voices from the open source revolution* was published [11]. The movement has been driven by not only lower development costs but also better security. Open source software has more eyes looking at it, making it less likely to has security flaws. Users can be confident that open source software will not maliciously access or distribute personal information.

Genie provides a trusted ecosystem of data handling software. All of the platform's software, including the SGX enclave software, user-side software, and distributed application (Dapp), that processes raw private data and generates outputs

to any human users must be open source. Professional security auditors and the platform users can inspect everything the system does and ensure that it does nothing harmful to their data. The auditors can also re-compile the code to generate the initial state of the software, which can be used to verify the software's installation package.

Open source software also provides several other advantages. The accessibility of open source code allows a large number of people to contribute to ongoing improvements of the software, including the identification of vulnerabilities. One disadvantage, however, is that vulnerabilities in the software are also open to the public, which could provide an attacker the knowledge to mount attacks on deployments of the software that have not yet been patched.

To minimize the risk of vulnerabilities, we carry out security inspection, testing, and third party auditing for each new release. The platform's users can also inspect the software by themselves because they have the entirety of the source code.

B. The SGX Trusted Execution Environment

Intel CPU Software Guard eXtensions (SGX) can create secure enclaves. An enclave is a hardware-isolated section of CPU memory which cannot be accessed from outside of the enclave, even with system privileges [10]. An SGX enclave can be used to run secure software and store sensitive data such as passwords, private keys, and personal data.

While an SGX enclave is isolated from the outside world, it is not safe if the software running inside it is malicious. The process of attestation ensures that code running inside an enclave is tamper-free [26], [18], [5]. Basically, the attestation process asks the platform on which the enclave is running to provide proof of the software's initial state in the enclave. Then, the proof is signed by the secret private key of the CPU on the platform. The Intel Attestation Service (IAS) can verify the signature and approve that the software running in the enclave has specific initial state which is the same as the executable image of the enclave.

But there are still two issues even when the enclave has been attested. First, how to know if the software running in the enclave is safe? This question can be answered by the audit reports on the open source codes. And the attestation can ensure the running enclave is identical to the software being audited.

Second, how can the attester know that the public key belongs to secure hardware and not a malicious device? The answer is that the public key must be provided or certified by a trusted organization. For SGX, a secret private key and a public key pair is generated during the manufacturing of the CPU. The secret key is stored and kept secret inside the CPU. The public key is stored by Intel. Intel doesn't publish the CPU public keys. Instead, Intel provides an attestation service to verify the signature of SGX CPU.

Therefore, the trust of the enclave relies on both open source code and Intel. Some people criticize SGX because it requires trust in Intel [34]. However, trusted systems must always rely on a root of trust. Users have to trust the manufacturing of the secure CPU and trust that there are no mistakes when handling the keys. Users have to trust the Public-key Infrastructure (PKI) [1] providing the attestation report certificates. Even if Intel directly published CPU public keys, users would still need to trust the certificates for the public keys signed by Intel.

C. Blockchain

Blockchain is a distributed public ledger based on cryptographic technologies first introduced by Nakamoto in 2008 [30].

Blockchain is a peer-to-peer network without any centralized administration. A new blockchain user or node can be created at any time. Each user account includes a key pair: one private key and one public key. The public key is used as the account ID and the private key is kept secret to prove account ownership and to generate signatures.

Blockchain generates a new block to store the new transactions in a fixed period of time. A proof-of-work mechanism is used to select an account (miner) to create the new block. The miner adds the new transactions, a hash of the previous block, and its signature into the new block and adds it to the blockchain. Data written into the blockchain is incorruptible, because any modification on a block needs the block's miner's private key to generate a new signature. Any changes would also need all the following blocks' miners' private keys in order to update the hashes of the modified blocks in their adjacent blocks. Therefore, even if a particular miner's private key is compromised sometime after the block was mined, it is still not possible to modify the block mined by the miner.

The transactions on public blockchains are transparent to anyone in the world. We can take advantage of this transparency to track usage of a data donor's data, even if it is incorporated into an AI model trained with many data donors' data. Both the acts of donating data to a model and querying a model are recorded as transactions.

A smart contract on a blockchain is a piece of executable code which can be used to define business logic and automate transactions. We developed two smart contracts for the platform to provide data registration, payment escrow, and tokens.

The purposes of using blockchain for the platform are:

1) Storing immutable data. This includes cryptographic hashes of raw data and ownership information. The hashes of the data can be used later to audit the data, namely to prove that the off-chain data has not been modified in an unauthorized manner. The validated genetic and health data, enclave images, enclave source codes, AI model data, and attestation reports are too large to be put on the blockchain and must be stored elsewhere. As such, only the hashes of these data are put on the blockchain and are used later to prove the data are tamper-free. Ownership information includes the account IDs of the owners of the data, models, and the enclave instances. Ownership information is used to send revenue generated from the data services to the

correct accounts according to the business logic defined in the smart contract.

2) Decentralizing authorizations. This allows users to create anonymous accounts by themselves, helpful for protecting the privacy of users.
3) Transparency of transactions. It is important for AI model data owned by many data donors who contribute their data to the model training and the model trainer. Provenance is important for data donors who want to know exactly how their data is being used and for model trainers who want to verify the validity of the data.
4) Business automation and monetary/financial incentives. Blockchain smart contracts support payment escrow and auto redistribution of revenues. This not only reduces transaction costs and delays but also enforces the incentive structure for data donors and all other contributors.

Even though blockchain has many advantages such as decentralized authorization, transparency, and business automation, it also has limitations we need to overcome.

Firstly, the storage space on the blockchain is very limited and costly. It is not feasible to put complete genetic and health data, AI models, or even registration information on the blockchain.

The data on the blockchain is publicly open. No private information should be put onto the blockchain.

On Genie platform, we use off-chain storages for either large or private data. The private data are stored in the data owner's storage behind a firewall or in SGX secure environments. Public data are stored on multiple public storages such as GeneTank data storage services, IPFS (InterPlanetary File System), Github, and/or Dropbox at the same time. The integrity of these off-chain data is ensured by the data's cryptographic hashes registered on the blockchain.

Secondly, blockchain mining performance is low. Each new block can only be generated in a fixed period (a few minutes) and each block can only store a limited number of transactions, due to block size and/or computational power restrictions. When there are thousands of transactions happening in a short period of time, the mining delay can be long. Many transactions have to increase their transaction fees to let them be mined earlier. The only way to improve the number of transactions per second (TPS), reduce latency, and reduce transaction fees is to do as much as possible off-chain [27]. We carefully designed the system to minimize necessary blockchain transactions to mitigate the performance and high transaction fee issues of the blockchain.

Thirdly, it is not possible to verify the authenticity of user-contributed data with blockchain alone, and encrypted genotype/phenotype information cannot be verified without decrypting it. The Genie platform uses real world, trustable public key infrastructure (PKI) [3] and trusted SGX enclaves to ensure the data registered on the blockchain are trustworthy. For example, the originality of the genetic and health data registered by the data owner can be verified with a digital signature provided by a data validation SGX enclave. The data validation enclave's integrity can in turn be verified with certificates included in its attestation report, signed by Intel IAS under PKI.

D. Data ownership preservation and privacy protection

Inspired by the ChainAnchor architecture for anonymously registering ownership of the IoT devices on the blockchain [16], we register the attested enclave to the blockchain to ensure unchangeable ownership.

Figure 1 depicts the ownership-preserving and privacy-protecting framework. We open source the source codes of the

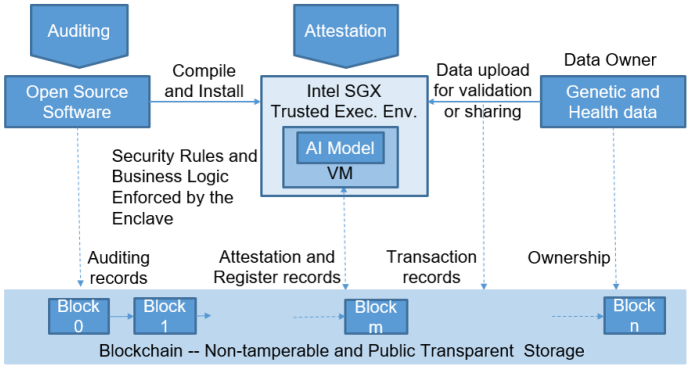


Fig. 1. The Security Framework

software that handles sensitive data, so security experts and/or data owners can audit the software to ensure its safety. Intel SGX gives us a secure execution environment (or enclave) in which we can run the audited code. The enclave's CPU hardware prevents any access from outside of the enclave even with system privileges, and the software that runs in the enclave can be attested by the CPU-signed measurement of the software loaded in the memory.

The enclave software itself is another important factor for the safety of the enclave. We open source the source codes of the enclave; they must pass security audits before users can trust them.

With source code auditing and enclave attestation, the enclave software becomes a trustable entity which can be used to enforce data safety rules and some business logic when working with the blockchain. These business logic and security rules include:

1) Never send out any data during model training.
2) Delete user data after use.
3) Only accept data validated by a validation enclave.
4) Record every use and payment transaction of AI model prediction services on the blockchain.

Based on the first rule, no human can see the raw data in the enclave. The shared data can only be used by the enclave software to train an AI or statistical model. The query results are generated from the model, not from individual user data. In this way, the insights of genetic and health data are shared without exposing anyone's raw data.

IV. The Architecture of the Genie Platform

The Genie platform currently is implemented with Ethereum as the blockchain backbone to integrate all the data and services into an anonymous, secure, privacy protected, and open system.

The system is trustless. All the trainers, runners, and users are anonymous participants, none of whom are assumed to be trustful for security. No real-world identification information is required for any of the participants. The openness of the platform allows people to donate data, utilize data, provide services, and use services with very low management costs.

On this trustless platform, the safety of the AI Model and user genetic and prototypic data is well-protected with a combination of open source, blockchain, and secure execution environment technologies as described in the Principles section. The security mechanism of the platform protects all user data (including the AI model algorithms, model parameters, and user genetic and phenotypic data) while these data are being processed, transferred, and stored.

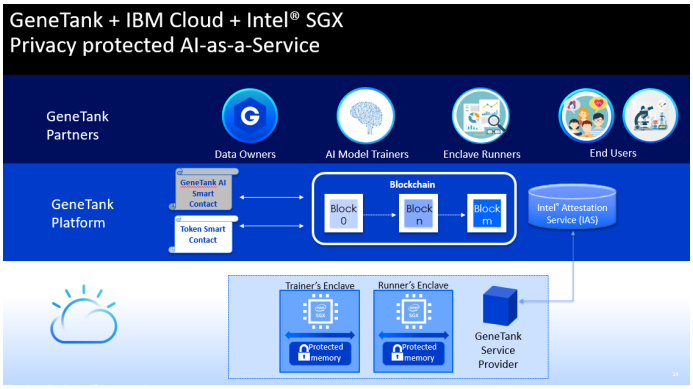


Fig. 2. The Architecture of the Genie Platform

As shown in Figure 2, the platform uses IBM Cloud to host the Genie SGX enclaves powered by Intel processors. Genie users can use the enclaves to train AI models and run the models in a trustable way.

The picture shows the entities of the system as well:

- Data owner: Individuals or organizations who own the genetic and health data.
- Model trainer: Pharmaceutical companies, biomedical companies, or medical researchers who want to use the data from the data owners to train their artificial intelligence or statistical models.
- Enclave runner: People or organizations who have computation resources which can be used to run the enclaves for data validation or model prediction services.
- End user: Individuals, hospitals, or pharmaceutical companies who want to use the model prediction services to predict the risk of diseases, drug responses, and traits of individuals with certain genotypes.
- Blockchain smart contract: The smart contract developed by GeneTank to register the data, models, and enclaves and automate the business logic.
- Intel IAS: The attestation service which is provided by Intel for the enclaves running on Intel CPU platform.
- GeneTank Service Provider: A proxy for the enclaves which need to be attested. It forwards the attestation request to Intel IAS, and sends back the attestation reports from Intel IAS.
- Enclave: There are three types of enclaves: data validation enclave, model training enclave, and model query service enclave.
  1) The data validation enclave validates data from data owners and provide a signature for the data if the data are valid.
  2) The model training enclave collects data from data owners and performs model training securely. There is no single output which can be sent out of the enclave.
  3) The query service enclave uses genetic and/or phenotypic data from the end user as input, runs the model to generates prediction results, and sends the results to the end user.

V. Data Flow

To describe how the platform works, we introduce four main data flows of the system in following sections:

A) New Enclave registration, auditing and attestation flow
B) Data preprocessing, validation and registration
C) Model registration and data recruiting
D) AI model query flow.

A. Enclave registration, auditing, and attestation flow

Enclaves can be developed by anyone who wants to contribute to the platform. The basic requirement for the enclaves is that they must be registered on the blockchain and accepted by the participants of the platform. GeneTank currently develops the enclaves for the initial phase of the platform.

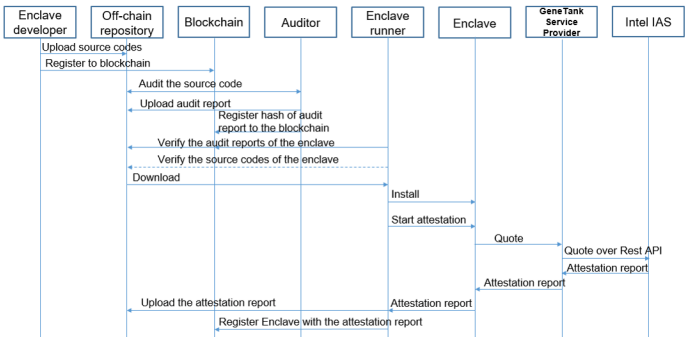


Fig. 3. The Enclave Attestation and Registration Flow

As shown in Figure 3, the enclave developer uploads the source codes, an executable binary image, and description information of the enclave to one or more publicly accessible repositories (e.g. Github, or GeneTank repository), then registers a hash of the source code and a measurement of the executable image of the enclave to the blockchain.

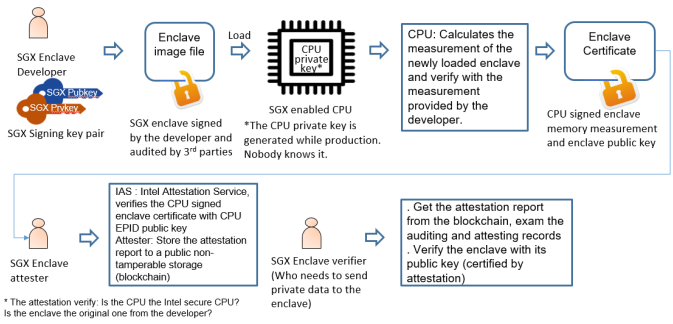


Fig. 4. The Enclave Attestation Flow

The registrations of any information on the blockchain are done with a Distributed Application (Dapp) developed by GeneTank.

The auditors (can be professionals or data owners) audit the source codes of the enclave, and compile the source codes into an executable image, and calculate the measurement of the generated image. The measurement must be the same with the one registered by the enclave developer. The auditor uploads an audit report to the public / off-chain repositories and registers a hash of the audit report on the blockchain.

The enclave runner verifies the enclave by reviewing the audit reports and/or source codes on the repository and checks against the registration records on the blockchain. If the enclave meets all the security requirements, the enclave runner downloads and installs the enclave.

After installation, the enclave runner starts attestation. The enclave generates a quote for itself. The quote includes the measurement of the enclave memory image before initialization, the public key of a self-generated key pair for its identification, some other information, and the signature by the CPUs secret key. The enclave sends the quote to the GeneTank Service Provider server. The server forwards the quote to the Intel IAS through a secure rest API.

The IAS verifies the quote with the EPID of the CPU and generates an attestation report. The report is sent back to the GeneTank Service Provider, then forwarded to the enclave and the enclave runner.

The enclave runner uploads the enclave information including an attestation report, the p2p address of the enclave, and some other information to the repository and registers the hash of the enclave information to the blockchain.

Some details of the enclave attestation flow of the platform are shown in Figure 4.

B. Data owner registration

When data owners obtain data from DNA sequencing companies or upload phenotypic information by answering questionnaires or providing medical records, they encrypt and upload the data to a data validation enclave for preprocessing and validation.

The validation enclave decrypts data in the secure environment and processes it with an artificial intelligent algorithm to identify fabricated data. Fake data will fail to pass the data validation.

The enclave sends back the processed data and an enclave signed report about the data through an encrypted communication channel. Then the data owner stores the data and the validation report locally and safely and registers the hash of the report to the blockchain.

The registered hash of the report acts as an ownership record of the data and will be used by the model trainer to verify the report. Only the data registered on the smart contract can be used for model training.

The data registration can be withdrawn at any time by the data owner.

This procedure is described in Figure 5.

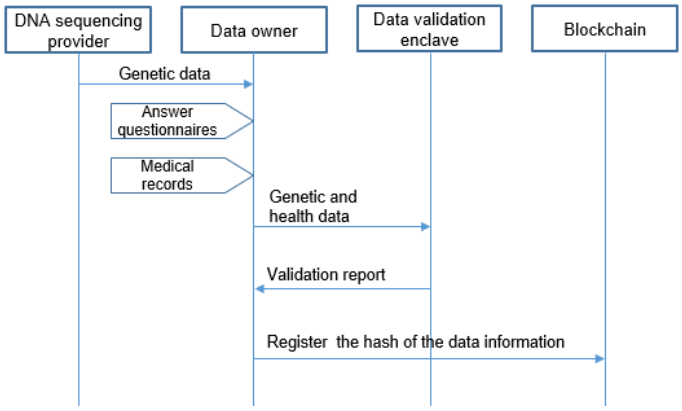


Fig. 5. The Data Validation and Data Owner Registration Flow

Each time the data owners update their data, the data must be sent to a data validation and preprocessing enclave. The data owners will get a data validation report and pre-processed data.

C. Model registration and data recruiting

A model trainer recruits data from the Genie platform to train a new model. The overall model registration and data recruiting is depicted in Figure 6.

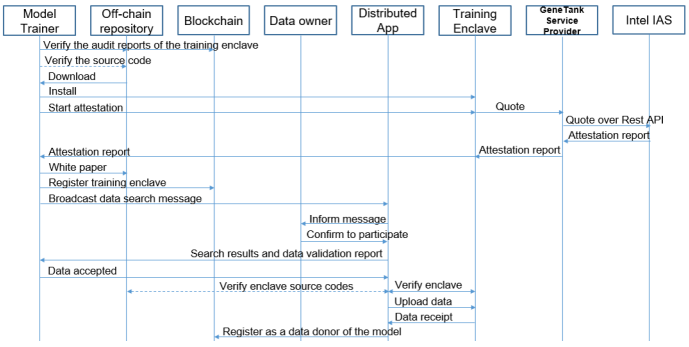


Fig. 6. The Model Trainer Registration and Data Searching

The trainer has to go through the following steps:

- Install an enclave package registered on the blockchain

- Get the newly installed enclave instance attested
- Upload a white paper of the model to the off-chain repository to describe what service the model will provide, what data are required for the model training, what are the instant payments for the data donors, how the revenues of the model will be shared, etc.
- Register the model as recruiting model to the blockchain.

After a training model is registered, the data recruiting follows these steps:

- The trainer sends data recruiting messages (p2p message through the blockchain) to all the data owners
- Each Dapp (the same application as for registration) processes the message and analyzes whether the local user data meet the data requirements. If yes, the Dapp informs the data owner to check the conditions and rewards of the model. If the data owner confirms to share the data, the Dapp sends a search result and the validation report of user data to the model trainer. The data owners can also configure their interests' types of models (whitelist) or the types of the models they dont like (blacklist).
- The model trainer checks the search result and the data report. If it is acceptable, the trainer asks the data owner to send the data to the training enclave
- The data owner reviews the auditing reports of the enclave; (optional) redoes the auditing by themselves to further ensure safety of the enclave; verifies the enclave with the enclave public key in attestation report registered on the blockchain and some other information in the off-chain repository; (optional) asks the enclave to redo the attestation if the attestation report is not up to date; if the enclave is safe, sends the data to the enclave provided by the trainer and gets a receipt from the enclave through an encrypted secure channel
- The data owner registers himself as a data donor of the model with the receipt (including quality level, a signature of the enclave)
- The smart contract verifies the receipt from the enclave before it accepts the donor registration.

The data owners may withdraw their data anytime during model training but if the data have been used, the effects of the data in the model may not be removed.

D. Model training and model runner enclave registration

Model training is done within the enclave. Nobody can see the intermediate and final results of the training. The trainer may use their own data to query the model to evaluate the training result. These evaluation activities must be recorded on the blockchain (enforced by the SGX enclave) as other ordinary queries which must be paid on the blockchain.

Data from the owners are deleted soon after the model is trained to avoid long-term vulnerabilities regarding data safety. This is enforced by the audited codes of the enclave. The donor's data is not possible to be used for training other models unless the data owner decides to participate in these models.

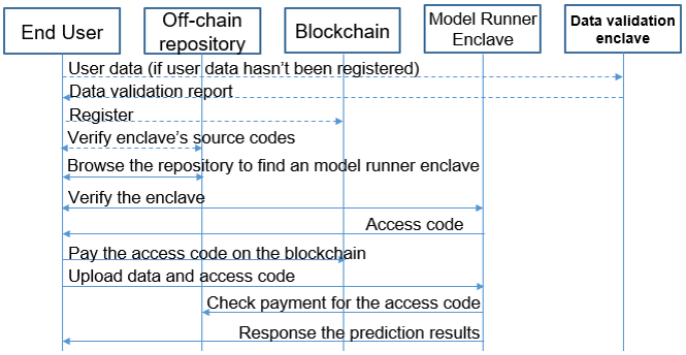


Fig. 7. The Model Query Flow

The trainer changes the status of the model from training model to trained model on the blockchain so that it becomes ready for model running. The trained model is stored inside the enclave. The model trainer registration keeps all the ownership information on the blockchain.

The model running includes the steps of installing model runner enclave, attesting the enclave, and registering the enclave.

The data flow of the runner enclave installation, attestation, and registration is largely the same as the validation and training enclaves. The differences are that the model codes must be open source and have passed auditing to prevent leaking model data when generating query result outputs.

E. Model querying

When anyone (end user) wants to use an AI model service to get predictions based on their genetic and health data, they can follow the data flow as shown in Figure 7 through the Dapp. If an end users data has not been registered yet, they need to validate and register the data to link ownership of that data. The genetic data and an optional part of health data are encrypted and uploaded to the enclave.

The enclave accesses the blockchain to verify a successful payment of the access code. Once the payment is valid, the enclave runs the AI model and returns prediction results. Otherwise, it refuses the request.

After the service is provided successfully, the payment from the end user is transferred to all of the accounts of the model's stakeholders (which include data donors, model trainers, and model runners) or temporarily saved in the smart contract of the blockchain until the stakeholders claim it.

VI. Details of Design

We explained how the Genie platform works in previous sections. Some details of the implementation of the platform are described below about the P2P communication, blockchain, smart contract, data management, and enclave designs.

A. P2P communication

The platform is built on a distributed peer-to-peer (P2P) blockchain network which has the advantages of being anonymous and privacy-protected. The native blockchain can only

send broadcasting messages. We extended the P2P protocol to support point to point messages as well. The Dapp can create P2P accounts to receive/send instant messages from/to other platform users. The P2P account is different from the blockchain account which can be found publicly on the blockchain. This can further protect user privacy and avoid unwanted harassment.

The enclaves are also on the P2P communication network. The data owners/donors and the model users can communicate with the enclaves over the P2P network.

Over the P2P network, platform users can:

Search for peers: Search the endpoint (IP addresses and port numbers) of any given P2P account ID;

Send unicast messages: Send messages to a specific P2P account;

Send broadcast messages: Send messages to all the Dapps. The broadcast message sender must be registered on the blockchain as a model trainer. The Dapp can filter the messages according to the preference setting of its user;

Chat in a chat room: A chat room is a channel for a specific topic. The Dapp can join these channels based on the user's interests.

B. Blockchain and Smart Contract

The underlying design eliminates the need for a trustable third party through the use of Ethereum smart contracts [7] and SGX. Smart contracts are symbiotically linked to the blockchain. Writing to a blockchain requires someone to send a transaction which needs to be confirmed by other nodes. The mechanism to achieve this uses either proof of work or proof of stake or a hybrid which is the case in the latest version of Ethereum. Ethereum provides a Turing complete Ethereum virtual machine where smart contracts run. Smart contracts are codes that are executed depending on conditions set within the code. Ether is used to incentivize people to participate and can also be used as a cryptocurrency to facilitate payments in different applications. Each computation in Ethereum has a gas fee.

The Genie platform uses smart contracts to store access control policies [39], [28] like who are registered and whether anybody has tampered with the enclaves, donor's data, and model training parameters. The GeneTank smart contract logs information about data donors, registration information by the creators, trainers and runners, model details trained by the trainer, and any relevant information about access codes generated by the runner for the users. Access codes are used to identify specific user data sent for prediction of a disease. Payments made towards a prediction result of a user's data are distributed to the different parties involved in this process: trainers, runners, and donors. Using tokens as a form of payments is easier than relying on a central authority to distribute fiat currencies.

C. Data management

Private personal data is usually stored locally on a users device where can protect privacy and ownership very well. For users on mobile devices, they can use the secure data storage service with an SGX enclave on the cloud.

There is also publicly available data such as the source codes and binary codes of enclaves, the white papers for data recruiting, the attestation reports of the enclaves, and the information of the resources available on the platform. These public data have at least one copy which is stored on the GeneTank server. The platform supports other public storages such as Github, IPFS, etc. to improve accessibility.

D. The enclave software and AI model container

The enclave software design is one of the cornerstones of the platform. Security is the first priority but the performance and scalability for biomedical big data processing is also critical for the success of the platform.

We create a virtual machine (VM) as a container for the AI model software which run the AI algorithms. The container isolates the AI model from other parts of the enclave. The AI model running in the container can only access data provided through an input channel. The output from the container is tightly controlled by the enclave software. This design allows the container to run secret AI model training algorithms without fears of compromising the security of the data. It can protect trainer's proprietary AI algorithms and the privacy of data owners at the same time.

The virtual machine currently supports AI algorithms written in the R programming language. R is wildly adopted for both biomedical and deep learning applications. It matches the requirements of both bioinformatics data processing and machine learning very well.

The VM runs the R bytecode compiled with an R compiler [36]. The bytecode programs can be transferred into the enclave and run on the VM so that the enclave can do many different models without changing the code of the enclave. It makes the enclave software more stable and much less auditing work is required.

Any R programs which use the strictly controlled output channel must be open source. R programs with malicious code sending out secret information from the VM can't pass the security audit.

To prevent closed-source AI training model algorithms from sending out private data with side-channel attacks [4], [31], the enclave software randomizes the data input, out-of-enclave memory access, disk operation, inter-enclave communication by inserting dummy operations and rescheduling them to prevent reading out data by monitoring the AI models external behaviors.

Each enclave has a very limited memory size, much less than the requirements of biomedical big data processing. We developed a virtual memory system using the memory outside of the enclave to expand the memory size in the enclave. All the data written to the external memory or disk are encrypted. The encryption key for external memory is generated each time the enclave starts up. The files permanently stored on the disk are encrypted with an SGX Sealing key [18].

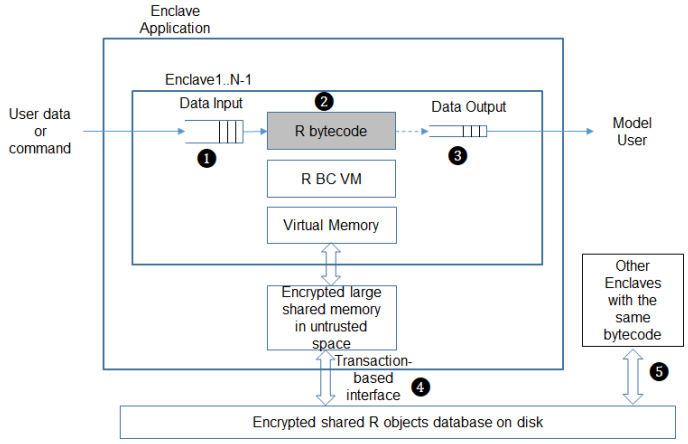


Fig. 8. The Enclave Design with Virtual Machine

We also support scale-out parallel processing which can divide the tasks into smaller ones and allocate to multiple enclaves to process. The enclaves can work in a coordinated manner to improve the performance of training. The enclaves can be deployed on one or multiple computers.

Private data in the enclave can be stored on permanent storage such as hard drives securely with the SGX data sealing feature. The sealed data are encrypted and can only be decrypted by the enclave which saved them. The design of the enclave with virtual machines is shown in Figure 8.

Some explanation about the numbered items in the figure:

1) The inputs include the data from data owners or the query users or commands from the trainers. The depth of the input buffer is randomly changed to prevent side-channel attack from malicious AI Model training R bytecode.
2) The R bytecode is driven by the data inputs include the commands. The bytecode is a blackbox (close source) for training and a whitebox (open source) for querying.
3) Only the querying bytecode which has been registered on the blockchain can generate outputs. The output is encrypted and can only be decrypted by the model query end user.
4) The external disk is a database of R objects. Any read-modify-write operations are managed within transactions to protect data in the event that the system crash.
5) The database can be shared with other enclaves with the same R bytecodes to support parallel training

VII. Conclusion

Genie is a AI-as-a-service marketplace platform, empowering by public genetic and health data. Genie provide highest standard of secure and immutability with Intel SGX, blockchain, and open source technologies to protect the privacy and preserve ownership of data.

The platform is decentralized and open to all individuals or organization data owners. Pharmaceutical companies, biotech companies, and biomedical researchers can use the platform to recruit data easily through distributed data searches for AI model training and create powerful models for disease predictions, drug responses, and personal traits. Individuals and hospitals can access the services provided by these AI models.

The platform enforces the rights of data ownership including possession, control, distribution, and disposal of the owner's genetic and health data. The platform also maintains the data donors' and model trainer's ownership of the AI models. The revenues generated by the AI models' prediction services are transferred automatically to all the owners of the models and the people who run the models. It encourages all the participants to continue contributing to the platform and forms a positive incentive loop.

VIII. Acknowledgements

Bixin Zhang who offered the ideas of P2P communications and edited the text.

This paper was supported by IBM cloud with Intel SGX cloud servers. MIT Sandbox program provides start-up fund for GeneTank. Creative Destruction Lab provides mentorship support.

References

[1] Carlisle Adams and Steve Lloyd. *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional, 2003.
[2] Jean-Claude Bajard, Julien Eynard, M Anwar Hasan, and Vincent Zucca. A full rns variant of fv like somewhat homomorphic encryption schemes. In *International Conference on Selected Areas in Cryptography*, pages 423–442. Springer, 2016.
[3] Sharon Boeyen and Tim Moses. Trust management in the public-key infrastructure. 2003.
[4] Ferdinand Brasser, Srdjan Capkun, Alexandra Dmitrienko, Tommaso Frassetto, Kari Kostiainen, Urs Müller, and Ahmad-Reza Sadeghi. Dr. sgx: Hardening sgx enclaves against cache attacks with data location randomization. *arXiv preprint arXiv:1709.09917*, 2017.
[5] Ernie Brickell and Jiangtao Li. Enhanced privacy id: A direct anonymous attestation scheme with enhanced revocation capabilities. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 21–30. ACM, 2007.
[6] Wylie Burke and Bruce M Psaty. Personalized medicine in the era of genomics. *Jama*, 298(14):1682–1684, 2007.
[7] Vitalik Buterin. A next-generation smart contract and decentralized application platform.
[8] Hao Chen, Kim Laine, and Rachel Player. Simple encrypted arithmetic library-seal v2. 1. In *International Conference on Financial Cryptography and Data Security*, pages 3–18. Springer, 2017.
[9] Melina Claussnitzer, Chi-Chung Hui, and Manolis Kellis. Fto obesity variant and adipocyte browning in humans. *The New England journal of medicine*, 374(2):192, 2016.
[10] Victor Costan and Srinivas Devadas. Intel sgx explained. *IACR Cryptology ePrint Archive*, 2016:86, 2016.
[11] Sam Ockman DiBona Chris. Open sources: Voices from the open source revolution. 1999.
[12] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88, 2016.
[13] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
[14] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford University, 2009.

[15] Elizabeta Gjoneska, Andreas R Pfenning, Hansruedi Mathys, Gerald Quon, Anshul Kundaje, Li-Huei Tsai, and Manolis Kellis. Conserved epigenomic signals in mice and humans reveal immune basis of alzheimers disease. *Nature*, 518(7539):365, 2015.

[16] Thomas Hardjono and Ned Smith. Cloud-based commissioning of constrained devices using permissioned blockchains. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 29–36. ACM, 2016.

[17] Karen Y He, Dongliang Ge, and Max M He. Big data analytics for genomic medicine. *International journal of molecular sciences*, 18(2):412, 2017.

[18] Intel. Intel sgx developer guide.

[19] Intel. Intel software guard extensions (intel sgx).

[20] Miran Kim and Kristin Lauter. Private genome analysis through homomorphic encryption. In *BMC medical informatics and decision making*, volume 15, page S3. BioMed Central, 2015.

[21] M.S. Kris Wetterstrand. Dna sequencing costs: Data.

[22] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6):1211–1220, 2017.

[23] Yue Li and Manolis Kellis. Joint bayesian inference of risk variants and tissue-specific epigenomic enrichments across multiple complex human diseases. *Nucleic acids research*, 44(18):e144–e144, 2016.

[24] Laure A Linn and Martha B Koo. Blockchain for health data and its potential use in health it and health care related research. Gaithersburg, Maryland, United States: ONC/NIST, 2016.

[25] Dianbo Liu, Fengjiao Peng, Andrew Shea, and Rosalind Picard. Deepfacelift: Interpretable personalized models for automatic estimation of self-reported pain. *Journal of Machine Learning Research*, 66:1–16, 2017.

[26] Rafael Pass, Elaine Shi, and Florian Tramer. Formal abstractions for attested execution secure processors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 260–289. Springer, 2017.

[27] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016.

[28] Aravind Ramachandran and Dr. Murat Kantarcioglu. Using blockchain and smart contracts for secure data provenance management. *arXiv preprint arXiv:1709.10000*, 2017.

[29] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. 1978.

[30] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system.

[31] Nakamoto S. Intel software guard extensions (intel sgx).

[32] Md Nazmus Sadat, Md Momin Al Aziz, Noman Mohammed, Feng Chen, Shuang Wang, and Xiaoqian Jiang. Safety: Secure gwas in federated environment through a hybrid solution with intel sgx and homomorphic encryption. *arXiv preprint arXiv:1703.02577*, 2017.

[33] Zachary D Stephens, Skylar Y Lee, Faraz Faghri, Roy H Campbell, Chengxiang Zhai, Miles J Efron, Ravishankar Iyer, Michael C Schatz, Saurabh Sinha, and Gene E Robinson. Big data: astronomical or genomical? *PLoS biology*, 13(7):e1002195, 2015.

[34] Yogesh Swami. Intel sgx remote attestation is not sufficient. 2017.

[35] Alex Pentland Thomas. Hardjono, D. Shrier. *Trust::Data: A New Framework for Identity and Data Sharing*. CreateSpace Independent Publishing Platform, 2016.

[36] Luke Tierney. A byte code compiler for r. *system*, 6:0–010, 2016.

[37] Jill U. Adams. Genetics: Big hopes for big data. 527:S108–S109, 11 2015.

[38] Lucas D Ward and Manolis Kellis. Haploreg v4: systematic mining of putative causal variants, cell types, regulators and target genes for human complex traits and disease. *Nucleic acids research*, 44(D1):D877–D881, 2015.

[39] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger.

[40] World Economic Forum. Personal Data: The Emergence of a New Asset Class, 2011. http://www.weforum.org/reports/ personal-data-emergence-new-asset-class.

coindesk

GM, BMW Back Blockchain Data Sharing for Self-Driving

Ian Allison
Apr 8, 2019
https://www.coindesk.com/gm-bmw-back-blockchain-data-sharing-for-self-driving-cars

Car giants General Motors and BMW are backing blockchain tech as a way to share self-driving car data among themselves and other automakers.

It's all part of a bid to unlock valuable data held in silos which will ultimately get autonomous vehicles on the road sooner. Exploratory work in this area is being done under the auspices of the Mobility Open Blockchain Initiative (MOBI), a consortium formed last year to harmonize the development of distributed ledger technology (DLT) across the "smart mobility" industry.

CoinDesk has learned that the next MOBI working group on so-called autonomous vehicle data markets (AVDM) will be chaired by General Motors (GM). The automaker has clearly been thinking about using blockchain to share data for some time, having filed a patent detailing such a system for fleets of self-driving cars at the end of last year.

The new AVDM working group chairman Michal Filipowski, manager global innovation of General Motors, told CoinDesk in an email:

"I am excited to chair the AVDM working group and kickoff the development of our collaborative efforts with the other OEM [original equipment manufacturers] and supplier MOBI members."

And notably, BMW, a founding member of MOBI, has also expressed its interest in the data-sharing use case for the first time. (Previously BMW tested blockchain to track mileage of leased vehicles.)

The German manufacturer, like many others in the auto space, has realized that keeping self-driving data in silos is a "major barrier" to widespread adoption of autonomous vehicles.

“With the advent of blockchain, decentral[ized] data management can be implemented in a privacy-preserving and efficient way,” Andre Luckow, blockchain lead at BMW Group, told CoinDesk. “Further, emerging technologies, such as decentral machine learning, secure multi-party confidential computing, and decentral data markets, will provide the fabric for data processing in the autonomous age.”

Stepping back, the push to foster autonomous vehicles faces a key hurdle: the sheer volume of data self-driving cars must consume in order to “learn” how to drive in different places and scenarios. Driving around a test track is one thing but negotiating a busy city center on a rainy day is quite another.

Crown jewels

According to a Rand Corp report, getting to the stage where AVs are safe in all conditions could take hundreds of billions of self-driven miles, a process by which data are gathered using cameras and Lidar (a detection system which works on the principle of radar, but uses light from a laser).

Pooling this data together to train artificial intelligence on might seem like a no-brainer, but autonomous vehicle companies – be they carmakers or Uber or Google-affiliated, Waymo – tend to think of their self-driving data as their crown jewels.

This is where blockchain comes in, explains Sebastien Henot, head of business innovation at the Renault-Nissan-Mitsubishi Alliance Innovation Lab in the Silicon Valley (who chairs MOBI’s vehicle identity working group). He told CoinDesk:

“The old-fashioned way is that everybody thinks their data alone is so precious. The new way is to consider data sets like cooking ingredients: you need to be able to mix multiple ones to create something really valuable. Data marketplaces call technically for blockchain because you can create an environment where rules are clear in terms of who shares what data with who.”

Another MOBI member, Ocean Protocol (which went live Monday), is focused on the creating blockchain-based data markets and running a shared AI on them. Ocean co-founder Trent McConaghy is aiming to create a kind of enterprise data commons where everyone can benefit, yet at the same time, this data can be prevented from escaping beyond the firewalls of any one company.

McConaghy explained that Ocean takes “federated machine learning” (machine learning built without direct access to training data, where data remains in its original location, such as on a smartphone, for instance) and gives it an additional dose of decentralization.

Google and others have been “pushing pretty hard on centralized federated learning,” said McConaghy, where they control the whole process.

“The makes the holders of the data feel pretty uneasy. So if you can actually remove that creepiness and the process of learning if done from silo to silo to silo in a decentralised fashion, that is much better. Decentralised federated leaning is what Ocean unlocks,” he told CoinDesk.

And this more decentralized approach is what BMW and GM, as part of AVDM group at MOBI, seem to be enthusiastic about.

Michael Ortmeier of BMW Group IT communications said Ocean’s approach to data sharing is one the company is “following with great interest.”

“We used the opportunity of the MOBI colloquium to speak with Ocean and other members and we will definitely continue and intensify these discussions,” he said.

Waymo data

It’s no secret that Waymo, the self-driving technology development company owned by Google parent Alphabet, is further ahead than anyone else in terms of how much data it has collected.

However, if you run the numbers, says Chris Ballinger, the founder and CEO of MOBI, it could still take many years for Waymo get there on its own.

Ballinger, the former head of mobility at Toyota, estimates that Waymo is accumulating a million miles of self-driven miles a month, adding:

“So you can say in miles it’s going to take millennia. Something has to be done and obviously it will speed up as more cars get on the road. Once everybody gets involved and once they start sharing it will be an order of magnitude increase.”

However, Vint Cerf, vice president and chief internet evangelist for Google, countered the claim the Waymo might way off when it comes to reaching its AV goals.

It depends what you mean by “driving data,” said Cerf. “We have billions of miles in simulation by generating direct inputs into the software that emulate what the sensors see,” he told CoinDesk by email.

Regards the possibility of car companies using blockchain networks to share data, Cerf added:

“I do not see additional value in the overhead of blockchain vs digital signatures.”

Blockchain-Inspired Future Accounting

March 25
By Corinne Finegan and Roger Meike
https://medium.com/blueprint-by-intuit/blockchain-inspired-future-accounting-b866c9b0763d

Exploring Triple Entry Bookkeeping

Confidence in blockchain and Bitcoin are at an all-time low. In addition to recent bad press, blockchain has problems with scale and energy consumption. Despite this, the concepts blockchain leverages hold real promise in unlocking value and better outcomes for consumers and business. We'll explore one overlooked and potentially breakthrough aspect that blockchain has made newly relevant: Triple Entry Bookkeeping.

Headlines over the last year have decried the fall of blockchain, and in recent months have reached a fever pitch. A sampling includes "Cryptocurrencies Have Failed, And Blockchain Still Has Yet to Be Proven Useful" (Forbes), "Blockchain companies go silent when their promises fall short." (MarketWatch). Or the particularly rosy, "Don't believe the hype: There are no good uses for blockchain" (American Banker).

The news about Bitcoin, the most high-profile cryptocurrency based upon blockchain technology, is equally bad or worse. Hacks, news such as the $137 million in cryptocurrency apparently lost forever due to the untimely death of a cryptocurrency CEO who had the only key (the wallets were later found to be empty), and rampant speculation have plagued Bitcoin of late.

Confidence in blockchain and Bitcoin are at an all-time low after reaching soaring heights. Like Icarus, blockchain flew too close to the sun and its proverbial wings have melted.

Given all the bad press, it would be easy to focus on the shortcomings of blockchain, such as its challenges with scaling or the huge energy consumption of Bitcoin mining, known as proof of work (POW). Bitcoin POW is estimated to consume the same amount of electricity that Switzerland does in one year. Focusing on this would miss an important opportunity, however.

Beyond the hype and bad news cycle, concepts underlying blockchain hold real promise in unlocking value and better outcomes for consumers and businesses. Warren Buffet may have put it best recently when he said that Bitcoin is a "delusion" but Blockchain is "ingenious." It's worth exploring how the reduced costs of verification and greater levels of trust engendered through blockchain-adjacent tech could benefit businesses and consumers.

Blockchain's original purpose was that of a shared and decentralized public ledger. This public ledger is verified by multiple parties. This establishes some trust in the veracity and provenance of the information. Because multiple parties verify the information on the blockchain, through a process called mining, the data is immutable. This means data can't be changed.

Example: Triple Entry Bookkeeping

Bookkeeping, the process of tracking financial transactions and assets for a business, can be traced back thousands of years to the ancient civilization of Mesopotamia. The "books" consisted of a "ledger" or list that was used to record transactions or changes of state in an organization's financial situation. Over time this ledger represented the accumulated financial history of the organization. Back then, anyone who touched the books had to be completely trustworthy as it would be easy to fudge the numbers. Mistakes were easily missed as there was no cross check. These single ledgers limited the growth of organizations.

Fast forward a few millennia to the 1300s and the invention of double entry bookkeeping. This was a major innovation. Luca Pacioli, an Italian mathematician and colleague of Leonardo da Vinci, described the system of double-entry bookkeeping used by Venetian merchants in his "Summa de Arithmetica, Geometria, Proportioni et Proportionalita" in 1494. With double entry bookkeeping, businesses could now grow beyond trusted insiders via the introduction of a symmetry that was more difficult to fake and easier to keep accurate. It employed checks and balances across the different parts of the organization. This meant that collusion was required across an organization to commit fraud, not just changing one entry in a ledger, thus making it much more difficult.

This advance has become central to modern business and has been the standard for 700 years. Double entry accounting has allowed the growth of businesses as we see them today. However, while much better than its predecessor, double entry accounting is also vulnerable to fraud. Enron is a well-known example — in the 1990s, the company used creative accounting to obfuscate massive trading related debts and losses through manipulating revenue recognition. Part of the issue stems from the fact that transactions are still represented individually by each party involved. Each organization represents their version of a transaction independently of the other. There is no synchronization across organizations. In this case Enron was able to represent transactions in a way best suiting itself. As a result of the Enron scandal, regulations were tightened with the Sarbanes-Oxley Act of 2002. This Act requires more intense oversight and audit of the books of large public corporations. By shining a light onto the books of public companies it is hoped that fraud will be easier to spot.

Whether intentionally fraudulent or accidental, the two participants in a transaction can get out of sync because there are no checks and balances other than outside audit. Looking at the transactions that make up an economy, it can be difficult to arrive at a single source of truth.

Can Triple Entry Bookkeeping help?

One concept central to blockchain is that it acts as a central ledger across parties to be the single source of truth of a transaction. The transaction itself resides on a ledger outside and independent of any particular organization's books. This is a clear use of triple entry bookkeeping which was developed by Yuji Ijiri in 1986 and resurfaced by Ian Grigg's 2005 paper, just before Satoshi Nakamoto's bitcoin-defining paper was published in 2008.

A third ledger shared between multiple organizations is a potential solution to some of the limitations of double entry accounting. Just as double entry accounting allows individual businesses to grow, triple entry bookkeeping may allow whole economies to scale with trust and transparency. An entry in this third public ledger now carries more weight because it is vouched for by both parties involved in the transaction.

The India Goods and Services Tax (GST) is an early move toward triple entry bookkeeping. In this system, businesses are required to submit invoices each quarter. This is required so that they can pay taxes on the transactions. However, there is an interesting exception to this. If you sold widget X to your customer, but you can show that widget X is made up of parts (or services) that you acquired from someone else who has already paid their taxes when you acquired them, you don't have to pay that portion of the taxes. This encourages businesses to deal with legitimate, tax paying businesses, and allows the government to see a chain of transactions across the various players that ends in a product. In this case the government records become the third ledger and essentially the single source of truth about transactions in that economy. GST aims to simplify sales tax, increase transparency and limit fraud by ensuring only the incremental value created at each stage of a supply chain is taxed. The GST relies entirely on digital reporting. The jury is still out on how successful this will be given implementation issues and the fact that many small businesses in India do not have digital records, but it's a use case to watch.

Blockchain takes this a step further by only recognizing transactions that exist in the public ledger. It is, by definition, the single source of truth for all transactions. There is no way to carry out an official bitcoin transaction without putting it on the blockchain. As an auditor of a bitcoin account I need to look no further than blockchain. Any and all transactions are on the blockchain and if they are not there, they don't exist.

Let's consider how this changes common financial activities such as approving loans. Imagine you are a consumer or small business looking for a loan. In today's double entry accounting world, a loan officer requires proof of your financial status through income statements (like paychecks), tax filings and other records. Bad actors invent transactions and fake these documents to seem much lower financial risk than they actually are. When the loan officer looks

at your books, in a double entry system, they only know that your books are self-consistent. They would have to do a lot of digging to make sure each transaction is legitimate.

Now let's consider the loan officer above in the world where all transaction are on bitcoin blockchain or some other single source of truth that is outside of the individual players in the transaction. There is high confidence that any transaction listed is recorded by both parties involved and therefore highly likely to have actually occurred. In a triple entry system, this loan officer has much higher confidence that everything on the third ledger is likely to be legitimate as it would require complex collusion across organizations to be faked. This reduces friction for consumers and businesses and reduces verification costs for lenders.

The loan officer or auditor who has access to the transactions on such a third ledger would have access to a complete financial history of that account. While this may appear to add security issues, it is also an opportunity to increase privacy. Unlike blockchain, not all third ledgers need to be publicly readable. They can be permissioned. Imagine that the loan officer above has a set of criteria that they use to determine eligibility for a loan. They can encode this decision criteria into an algorithm, that returns a boolean value of true (you are eligible) or false (you are not eligible). Then, rather than giving access privileges to the loan officer, the loan applicant grants permission to run the algorithm against their account. The loan officer doesn't get to see your income or your taxes paid, but the algorithm can, in a trustworthy way, access this information and return true or false to the loan officer.

In the increasingly distributed and impersonal world of digital commerce, the trust and confidence once built from in person dealings no longer exists. By and large, business is no longer done by a handshake or dealing with the bank teller that you've seen for years. Triple Entry Bookkeeping has the ability to bridge this gap and foster trust and transparency.

Quantum Computing Is a Marathon Not a Sprint

CHRISTOPHER MONROE, IONQ APRIL 21, 2019
https://venturebeat.com/2019/04/21/quantum-computing-is-a-marathon-not-a-sprint/

Image Credit: Juhari Muhade/Getty Images

If you watch the technology headlines you might think something called quantum computing is the Next Big Thing. In January, USA Today declared IBM's new quantum computer one of the four most "wow worthy" announcements at CES, the annual gadget fest in Las Vegas. Gartner also listed quantum computing as one of the top technology trends for 2019, joining fan favorites like blockchain and virtual reality.

I've spent more than 25 years as a physicist researching quantum computers — machines that store and process information on individual atoms or particles, like photons — and I've started a company that is building them. I am convinced quantum computing is in fact a breakthrough technology that offers the only known way to attack some of the world's hardest problems in medicine, transportation, computer security, and other areas we haven't yet foreseen.

We must be clear, however, about what is and isn't happening next. The big quantum computing discoveries that will most impact society are still years away. In the meantime, we will see breathless announcements of records broken as the technology rapidly develops. These incremental advances are important for government, which has a role in encouraging this research, as well as for industries that need to start developing ways to use quantum computers as they become more powerful. But too much hype risks disillusionment that may slow the progress.

The first thing to know about quantum computers is that they are not a faster, better version of the computers we have now. You'll never trade in your laptop or smartphone for a quantum version. Quantum computers almost certainly won't run social networks, animate Pixar movies, or keep track of airline reservations. They solve different problems in different ways.

Quantum computers were proposed in 1982 by Richard Feynman, the Nobel prize winning physicist, who worried that conventional computers could never tackle problems in quantum mechanics, the well-established theory that predicts the behavior of small isolated particles such as atoms or electrons. Today, we do use conventional computers to simulate quantum models of material and chemical processes, but these simulations grind to a halt when faced with all the possible arrangements of electrons in even a small molecule or chunk of material.

Feynman's idea was simple: build a computer that stores information on individual particles — later named qubits — that already follow the very rules of quantum mechanics that seem to perplex conventional computers.

What's the difference? Ordinary computers think in certainties, digitizing every aspect of the world to well-defined numbers. Quantum computers probe all possibilities, constantly updating the probabilities of multiple scenarios. Add more qubits, and they can consider exponentially more scenarios. A quantum computer is programmed to consider all these possibilities and narrow them down to just a few, and then when the output is measured, it can tell us information about all those scenarios. It is critical that a quantum computer not be measured or looked at while it considers the uncountable number of possibilities. For that reason, qubits are like senators before a controversial vote: They shouldn't reveal their position until they are forced to.

Our world is filled with uncertainty, and quantum computers can be very helpful in selecting the best of several options. Thus a bank wouldn't use a quantum computer to track checking accounts. When you look at your balance, you want a single answer you can count on. But the bank might use a quantum computer to estimate how much money you will have in your account a year from now, based on the probability you will get a raise or get fired, whether your teenager will crash the car, if the stock market will crash, and how these factors interact.

To be clear, nobody has yet written a program that makes financial projections on a quantum computer. One reason is that, until now, there haven't been any quantum computers to try them out on. But after a lot of work, that's changed. Over the last few years, corporate, academic, and government groups have built machines that can isolate and manipulate particles or other types of qubits well enough to handle basic programs.

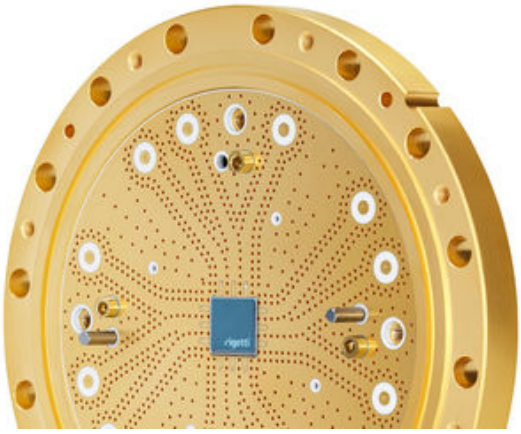
It takes exacting precision and extreme conditions to isolate and control qubits. Some quantum computers freeze solid-state circuits to close to absolute zero. Others uses electric fields to levitate atoms in a vacuum that is more pure than deep space, while using lasers to manipulate them with an accuracy of 1/10,000 the width of a human hair. These atomic qubits in particular can scale to much larger systems because they are all the same isolated atomic element, perfectly replicable, and they are so well isolated that they never reveal their qubit states until forced to.

In 3-5 years, these machines will perform certain calculations that would not be possible using ordinary computers. But it may be 5-10 years before any of these machines have the capacity and accuracy to solve useful problems. Along the way, I worry that some who read about quantum computing being the next big thing will feel let down and lose interest. We can't let that happen. Government needs to continue to support basic research, as Congress did passing the National Quantum Initiative Act last year. And the industrial community needs to start working with the current generation of quantum computers so they can develop the know-how and the software that will give them an edge as the technology improves.

Even then, you won't have a quantum computer on your desk or in your pocket. But you may start to see better drugs, more flexible materials, and organizations running more efficiently. All that will definitely be wow worthy.

Christopher Monroe is the Bice Zorn Professor of Physics and Distinguished Professor at the University of Maryland and co-founder and CEO of IonQ, a quantum computing startup.

Science AAAS



Quantum computers rely on superconducting chips like this one from Rigetti Computing in Berkeley, California.

How to Evaluate Computers That Don't Quite Exist

https://www.sciencemag.org/news/2019/06/how-evaluate-computers-don-t-quite-exist

By Adrian Cho
Jun. 26, 2019

To gauge the performance of a supercomputer, computer scientists turn to a standard tool: a set of algorithms called LINPACK that tests how fast the machine solves problems with huge numbers of variables. For quantum computers, which might one day solve certain problems that overwhelm conventional computers, no such benchmarking standard exists.

One reason is that the computers, which aim to harness the laws of quantum mechanics to accelerate certain computations, are still rudimentary, with radically different designs contending. In some, the quantum bits, or qubits, needed for computation are embodied in the spin of strings of trapped ions, whereas others rely on patches of superconducting metal resonating with microwaves. Comparing the embryonic architectures "is sort of like visiting a nursery school to decide which of the toddlers will become basketball stars," says Scott Aaronson, a computer scientist at the University of Texas in Austin.

Yet researchers are making some of their first attempts to take the measure of quantum computers. Last week, Margaret Martonosi, a computer scientist at Princeton University, and colleagues presented a head-to-head comparison of quantum computers from IBM, Rigetti Computing in Berkeley, California, and the University of Maryland (UMD) in College Park. The UMD machine, which uses trapped ions, ran a majority of 12 test algorithms more accurately than the other superconducting machines, the team reported at the International Symposium on Computer Architecture in Phoenix. Christopher Monroe, a UMD physicist and founder of the company IonQ, predicts such comparisons will become the standard.

“These toy algorithms give you a simple answer—did it work or not?” But even Martonosi warns against making too much of the tests. In fact, the analysis underscores how hard it is to compare quantum computers—which leaves room for designers to choose metrics that put their machines in a favorable light.

A conventional computer manipulates bits of information, encoded in transistors that can be switched on or off to represent zero or one. A qubit, however, can be set to zero and one simultaneously, say, by encoding it in an ion that can spin down for zero, up for one, or both ways at once. Qubits enable the machine to process many inputs simultaneously instead of one at a time. But the machine’s real power comes not through that massive parallelism, but in problems where possible solutions can be encoded in quantum waves that slosh among the qubits. The waves then interfere so that wrong solutions wash out and the right one emerges.

A quantum computer would be able to, for example, crack internet encryption schemes based on the factoring of huge numbers—a tough problem for a classical computer. But solving such problems would require 100,000 qubits and the means to correct errors in the delicate quantum waves. Such machines are decades away, researchers say. But quantum computers with even a few dozen noisy qubits will soon best conventional computers at certain tasks, developers say, and they’re searching for the metrics to prove it.

A quantum leap

With a quantum computer that relies on a superconducting chip, Rigetti Computing is seeking an application that will give it a practical advantage over conventional computers. Other companies are pushing other metrics to gauge progress.

COMPANY/ UNIVERSITY	COMPUTER TYPE	NUMBER OF QUBITS	PREFERRED METRIC
Google	Super-conducting	72	Quantum supremacy
IBM	Super-conducting	20	Quantum volume
Rigetti Computing	Super-conducting	16	Quantum advantage
University of Maryland	Trapped ions	5	Benchmark comparison

Solving a problem that a conventional computer cannot—so-called quantum supremacy—is one well-publicized metric. “It’s a ‘Hello world!’ project that shows your quantum computer works,” says John Martinis, a physicist in Santa Barbara, California, who leads Google’s efforts to achieve supremacy on a machine with 72 superconducting qubits.

The problem Google researchers have chosen is exceedingly abstract. Essentially, they program the quantum computer to repeatedly perform a random set of operations on the qubits. Thanks to quantum interference, the machine should spit out certain strings of zeros and ones with greater probability than others, instead of producing all strings with equal probabilities, as it would if there were no interference. What's more, predicting this exact distribution of outcomes overwhelms classical computers once the number of qubits climbs too high. So if Google researchers can measure that telltale distribution for their 72-qubit machine, then, in a sense, it will have achieved quantum supremacy by calculating something a conventional computer cannot. However, the arcane exercise won't usher in practical quantum computers, says Greg Kuperberg, a mathematician at the University of California, Davis. "It's supremacy to do something completely useless."

In contrast, researchers at Rigetti aim to show that a quantum computer can perform some useful task more accurately, faster, or more cheaply than conventional computers—a metric they call quantum advantage. "What we want are things that put us on the shortest path to commercial value," says Chad Rigetti, a physicist and founder of the startup. For example, he says, a quantum computer might be ideal for modeling the complex interplay of financial assets in a hedge fund.

In September 2018, Rigetti pledged $1 million to the first user who achieves quantum advantage on its publicly available machines. The current version has 16 superconducting qubits. Because the measure includes factors like cost, quantum advantage is not so tightly defined, says Aram Harrow, a physicist at the Massachusetts Institute of Technology in Cambridge. "If it's a little vague, that's not bad for Rigetti," Harrow says.

IBM researchers have defined a metric, called quantum volume, that measures a quantum computer's performance without comparing it to a conventional machine. It involves testing a quantum computer using random calculations like those Google is using. And it depends on both the number of qubits and the number of computational cycles a machine can handle before its quantum states fuzz out.

Using a machine with 20 superconducting qubits, IBM scientists have reached a quantum volume of 16 and aim to double it every year, says Jay Gambetta, a physicist at IBM's Thomas J. Watson Research Center in Yorktown Heights, New York. Breakthrough applications will follow naturally, he says. "I don't think that supremacy is something you shoot for. It's something we'll recognize once we've passed on to bigger and bigger things."

Then there are head-to-head comparisons like Martonosi's. In her test, the 5-qubit ion-based machine solved most test problems correctly 90% of the time, compared with 50% or less for superconducting-qubit machines. That difference reflects the current states of the technologies and not their potential, Martonosi says. For example, in a superconducting machine each qubit interacts only with its neighbors, but every ion in the UMD machine interacts with all the others, giving it an edge. Bigger ion-based machines won't share that advantage, however.

Martonosi says such comparisons show that all the quantum computers performed significantly better when programmed to account for differences in qubit noise and connectivity. "Across quite a wide range of [hardware] implementations, this appears to work," she says. "That's quite exciting."

Harrow questions how useful any of the current metrics will prove in the long run. The main challenge in quantum computing remains finding a technology that will scale up to thousands of qubits, he says. "These metrics are only loosely related to the scaling question."

COINTELEGRAPH
The Future of Money

German Government Says Blockchain Can "Support Europe's Unity at a Fundamental Level"

By Marie Huillet
MAR 27, 2019
https://cointelegraph.com/news/german-govt-says-blockchain-can-support-europes-unity-at-a-fundamental-level

Germany's Federal Office for Migration and Refugees (BAFM) has found that blockchain has far-reaching potential to improve asylum procedures. Following a successfully completed proof-of-concept (PoC), the findings were published on March 26 in a white paper.

The paper was edited by BAFM and authored by the Project Group Business & Information Systems Engineering of the Fraunhofer Institute for Applied Information Technology FIT.

The PoC — undertaken by BAFM, Fraunhofer FIT and an unnamed technology partner in the first half of 2018 — focused on evaluating blockchain's potential to support two crucial aspects of asylum procedures: the creation of reliable and secure digital identities and improving communication and cooperation between authorities at a municipal, state and national level.

For the PoC, the three partners used a private and permissioned version of an Ethereum-derived blockchain, using a proof-of-authority consensus algorithm.

The white paper outlines that blockchain can enable the creation of tamper resistant digital identities for refugees that arrive without ID documents, based on biometric data collected at the moment of their initial registration in the receiving country. This immutable blockchain-based identity would then support further aspects of the asylum procedure and ensure the consistent and secure identification of each asylum applicant across multiple organizations.

The white paper's authors propose that a robust, blockchain-based identity solution could have far-reaching positive pan-European implications, noting that:

"Blockchain could be the 'digital enabler' of European federalism in the asylum context. [...] A European platform for the decentralised management of asylum procedures [...] would enable

the transparent storage of a person's place of initial registration. [...] Digital identities are per se nationally agnostic and could thus support Europe's unity at a fundamental level."

The white paper notes that data protection laws pose a key challenge for blockchain innovation within a European context — a reference to the General Data Protection Regulation (GDPR), a landmark EU-wide legal framework for personal data privacy, which took effect in May 2018.

Nonetheless, a GDPR-compliant architecture for a blockchain-powered asylum system could be possible, the white paper suggests.

A Cointelegraph analysis published in fall 2018 studied the prospective benefits blockchain can bring to strained immigration systems worldwide.

Pacific Northwest
NATIONAL LABORATORY

Building Trust in Blockchain for the Electric Grid

PNNL pilots two use cases applying blockchain technology to improve the cybersecurity of critical electricity infrastructure

Lynne Roeder, PNNL
Mar 29, 2019
https://www.pnnl.gov/news-media/building-trust-blockchain-electric-grid

In the digital age, the speed and size of data transfer are increasing rapidly, raising concerns about privacy and security. This information management juggernaut presents a major challenge for energy utilities that need to exchange and store data securely, while at the same time increase the speed, reliability and efficiency of power delivery.

In one of the largest blockchain grid-cyber projects of its kind, PNNL is working with Guardtime, Washington State University, Avista, various industry vendors of industrial control systems and energy delivery systems, the U.S. Departments of Energy and Defense, and over a dozen industry advisors to test and demonstrate blockchain's ability to increase the cybersecurity resilience of electricity infrastructure.

In March, the team demonstrated two of the project's first use cases. The first use case focused on securing critical data stored and exchanged between the energy management system or distribution management system and energy delivery systems. The second use case demonstrated how blockchain can help improve asset management and supply chain security for critical energy delivery systems.

These use cases are important as energy utilities work to secure an increasing number of end points from evolving cyber threats, while at the same time managing the rapid expansion of distributed energy resources and other smart devices. These initial pilots suggest that blockchain shows potential to help secure these transactions while also enhancing grid resiliency by providing a novel security solution for managing and securing critical energy delivery systems and data.

Since its debut about a decade ago, this highly touted blockchain technology is still in its formative stages and often misunderstood. By validating and verifying the opportunities versus the hype, this blockchain project helped add clarity for securing the nation's power grid and other critical infrastructures.

"Think of the early days of the Internet," said Michael Mylrea, the project's primary investigator and a senior advisor for cyber security and resilience at PNNL. "While blockchain technology is still at a nascent stage and lacks a browser-level ease of use, these initial pilots demonstrated that blockchain could potentially unlock a new wave of innovation to help secure complex energy exchanges."

Blockchain primer

Blockchain software provides a digital ledger that records transactions of value using a cryptographic signature. The transactions are maintained in a continuous list of records, called blocks, with built-in protections against tampering.

Each block contains a timestamp and a cryptographic link to a previous block. Because the data that forms a block cannot be altered retroactively, the chain is very difficult to modify.

While blockchain technology is often linked with the cryptocurrency Bitcoin, the digital ledger that underpins it varies greatly in its characteristics, definition and application. In fact, an article last spring in The New York Times noted that a volatile cryptocurrency market has demonstrated that blockchain technology lacks oversight and standards, which continues to distract from the real potential of the technology. Namely, the application of digital ledger technology can increase trust and auditability of data in a way that will remove the need for third parties in sectors ranging from energy to finance.

In August, The Economist ran a piece directly linking blockchain to the energy industry and the possibility of decentralized energy transactions by directly linking consumers and producers of energy. Because blockchain-based transactions can be executed without middlemen, this "prosumer" approach could improve today's inefficient multi-tiered system, in which intermediaries transact on various levels.

Energy utilities are also taking notice of blockchain's transformative potential. However, they remain cautious due to lingering questions related to the interoperability, security and scalability of the technology.

Better, stronger, more efficient

Funded by the U.S. Department of Energy, the ***Keyless Infrastructure Security System (KISS)***project led by PNNL focused on:

- Developing a blockchain-enabled cybersecurity controller that uses PNNL's VOLTTRON™ platform to execute complex energy exchanges that cannot be modified or manipulated by cyber attacks
- Developing the first blockchain prototype to continuously monitor and autonomously verify the integrity of critical energy delivery systems
- Validating and verifying opportunities and challenges in applying blockchain to mitigate cyber threats to energy delivery system operation, configuration, and supply chain.

The recent blockchain pilots were important milestones that helped increase the energy sector's nascent understanding of blockchain. These findings helped fill several major blockchain research gaps in determining technical requirements for scalability, interoperability and security.

"Instead of assuming that blockchain is the panacea to all grid-cyber challenges, we performed a deep dive into over a dozen different blockchain technologies, closely examining the feasibility of its application to the energy sector," said Mylrea.

These findings helped pave the way for future research in applying the project's blockchain solution to live grid telemetry.

"How can we break the blockchain? Can we build it back in a way that is more resilient and easier to use? How can blockchain make our grid more efficient and resilient in response to evolving cyber threats?" Mylrea said these are the types of questions that future research will help answer.

PNNL's pilots provide valuable insight on both blockchain opportunities and challenges and provide greater clarity for a technology that lacks standards, regulations and definitions. Namely, not all blockchain applications are created equal.

Several blockchain solutions are energy intensive and often vulnerable to different cyberattacks. Public and private blockchains, and applications, configurations and implementations vary greatly in cost, latency, interoperability and applicability for securing critical energy infrastructure. PNNL's work focuses on addressing these very issues, particularly how to increase the trustworthiness and integrity of blockchain energy applications.

In 2019, the team expects to start commercializing a blockchain-enabled cybersecurity controller and move beyond field testing to utility-level deployments. They'll also explore opportunities to partner with industry and utilities to increase the speed, security and interoperability of complex energy transactions.

Above the radar

An ambitious project to be sure, but the idea of breaking the blockchain and building it back even stronger has gained a lot of attention. Greentech Media and other outlets reported on the project launch, and progress and pilot findings were featured at this year's RSA Conference, the largest global cyber security conference.

The project has also helped inform lawmakers and regulators. PNNL's Paul Skare testified at a U.S. Senate committee hearing in August on Energy Efficiency of Blockchain and Similar Technologies, and again in March to lawmakers in Washington state. Skare is the Chief of Cyber Security and Technical Group Manager for Energy and Environment at PNNL. His testimony included remarks about the KISS blockchain project.

"Our research is helping the energy sector move beyond the blockchain buzz, to validating and verifying the potential of blockchain for a modern power grid that is more resilient and efficient," said Mylrea.

Race to the finish

It's still too early to call blockchain the magic bullet for critical infrastructure cyber resilience. Evolving blockchain definitions create governance challenges for regulators and policy makers.

To add clarity and help give impetus to broad industry adoption, Mylrea and colleague Sri Nikhil Gourisetti lead the cybersecurity task force for the IEEE Blockchain Standards Committee and working group. The objective of that group is to develop standards and frameworks for the technology's application to the energy sector.

But the world is not waiting. Both startups and established companies around the globe are galloping into the wild-west of blockchain application development. In fact, Sri Nikhil Gourisetti notes that many Fortune 500 companies are exploring various blockchain applications for their businesses.

PNNL's work on blockchain leverages its core capabilities and domain expertise in grid cybersecurity to support DOE's Office of Cybersecurity, Energy Security, and Emergency Response (CESER) Multiyear Plan for Energy Sector Cybersecurity. Specifically, the work aligns with the DOE's goal of accelerating game-changing research, development and demonstration of resilient energy delivery systems and their ability to survive a cyberattack while sustaining critical functions.

A Blockchain Based on Gossip? – a Position Paper

Robbert van Renesse

Cornell University

A blockchain is an append-only sequence of blocks of arbitrary data. The two most popular approaches to blockchains are *permissionless blockchains* based on *Proof of Work* (PoW) and *permissioned blockchains* based on *Byzantine consensus* or *Byzantine Fault Tolerance* (BFT). The first is based on competitions between anonymous participants solving cryptopuzzles, while the latter is a cooperative approach based on mutual trust between participants. Major problems with PoW approaches include that the energy per transaction is enormous, the transaction rate is very low, and the latency is very high. A major problem with BFT is that membership is closed. Various other approaches to blockchains have been proposed to address these problems.

In this paper we propose yet another approach, based on *gossip* (*aka* epidemiological protocols) [1]. Gossip is an approach to agreement in so-called *eventually consistent systems*, and is particularly popular in NoSQL Key-Value Stores such as Dynamo, Cassandra, and so on. In a basic gossip protocol, there is a fixed group of participants. Periodically, each participant randomly selects a peer and exchanges state. This state is reconciled in a way so that all non-faulty participants eventually converge to the same state. It can be shown that this approach is efficient, converging in $O(\log N)$ gossip rounds where N is the number of participants, even in the face of participants failures and message loss [1]. Moreover, gossip protocols are amenable to open and dynamic membership whereby the membership itself is gossiped along with other state [4].

To make the application of gossip to blockchains more concrete, we propose, for simplicity, a setting similar to the Bitcoin blockchain. Blocks are 1 MByte. We assume participants have access to a good source of time, such as a GPS clock. We divide time into 10 minute epochs. Each hour would have six such epochs starting at the top of the hour. Each epoch is further subdivided into 10-second gossip rounds, allowing for 60 rounds of gossip. The intention is for the participants in an epoch to use gossip to agree upon a block. Note that the transaction rate would be the same as for the Bitcoin blockchain—here we only try to address energy consumption.

Each participant, in the background, collects transactions from clients, and at the top of an epoch fills a block with new transactions. The block also contains a *history hash*: a hash of the content of the previous block on the chain, as determined by the participant. The participant then starts gossiping its block and learning about the blocks of other participants. Before we go into further specifics about how the participants may end up agreeing on a particular block, we point out that gossip is prone to Byzantine attack. We will introduce and address various attacks as we go.

The first question is how participants would end up agreeing on a block. The straightforward

approach is to order blocks according to some agreed-upon metric. Each participant would simply keep track of the maximum block that it has heard from and gossip this block, possibly along with membership information. A participant would only consider blocks that contains a history hash matching its concept of the block agreed upon in the last epoch. If all participants were honest, then this approach would lead to a very high likelihood that all participants agree on the maximum among the proposed blocks at the end of the epoch.

Unfortunately, a Byzantine participant could introduce one or more new blocks that are higher than the previous maximum in the last few rounds of the epoch and cause the participants to end up disagreeing with one another. This is similar to the concept of *forks* in PoW chains, where miners advertise new blocks at approximately the same time, either accidentally or with selfish intention. In Bitcoin chains, this is resolved by the *longest-chain-wins* policy: when one branch in the fork has become longer than the other(s), the longest branch becomes part of the main chain. Because it is highly unlikely that branches both keep adding a block at exactly the same times, forks are *metastable*.

Can we create a gossip protocol that is also metastable in a way that once agreement has been reached it would be very hard to destabilize it? The answer is yes. Instead of ordering the blocks a priori based on some metric, the participants use a randomized approach that converges with high probability. In this approach, each participant keeps track of a table with a row for each peer (including self). Each row in the table contains

- a unique identifier for the peer;
- a gossip round number for the peer;
- the peer's *favorite* block in that round;
- a public key signature to prevent forging and tampering.

When two peers gossip, they reconcile their tables by adopting, for each participant, the row with the highest round number. Once reconciled, a participant counts for each block in the table how many participants favor that block. The participant then chooses the most favored block as its favorite block and increments its round number in its own row. If there is a tie among several blocks, the participant selects one of them uniformly at random.

In order to prevent an attack whereby a participant tries to gossip with every other participant in every gossip round, we use a *pull gossip*: a correct participant explicitly requests the state of k randomly chosen peers in each round, rather than sending its state to the peers. This way each correct participant updates its state based on that of at most k participants in each round. In the rest of this paper, we use $k = 1$.

Figure 1 shows results of simulations of this protocol for various numbers of participants up to 4096, slightly fewer than the number of miners in Bitcoin today (which is now close to 6000). Each experiment was run 250 times. The x-axis shows the number of participants on a log scale and the y-axis shows the number of rounds before all participants agree upon a block. With 4096 members it takes rarely more than 20 rounds to converge, and certainly within 60 rounds convergence is all but certain. Note that, like Bitcoin, we assume that the participants somehow form a connected network—without it, the blockchain could diverge until connectivity is established. BFT does not suffer from this problem.

With open membership, a Sybil attack is possible on the membership of the protocol described thus far. We leverage that, although it is easy to spoof an IP return address in an IP packet, it is

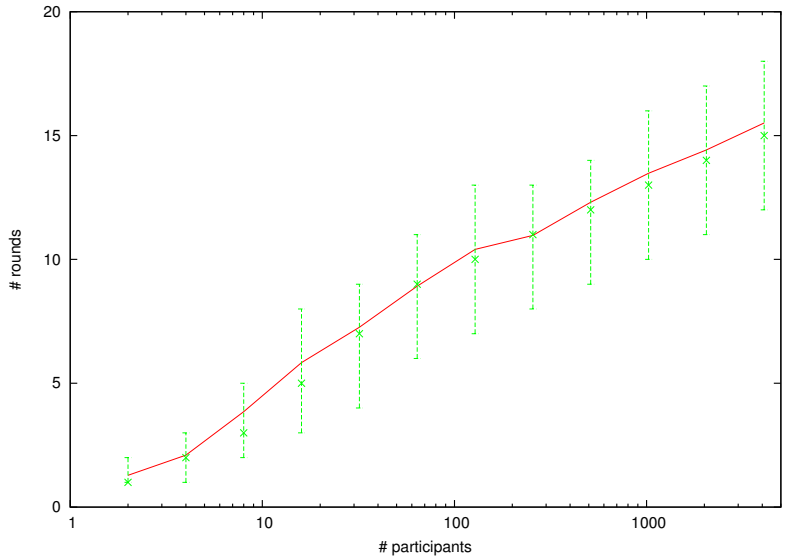


Figure 1: Gossip convergence time (in rounds) as a function of the number of participants. The line shows the average, and the error bars show the median as well as the 5^{th} and 95^{th} percentiles.

not easy to subvert routing and receive messages on a spoofed IP address. An adversary is thus limited by the number of public IP addresses that it actually has control over. More specifically, we propose that peers use TLS connections with endpoints identified by IP address.

In addition, we propose to use an intrusion-tolerant overlay network such as S-Fireflies [3, 2]. S-Fireflies is a self-stabilizing and Byzantine tolerant overlay network that provides its members with a view (approximation) of the current membership as well as each member with a small subset of the view. Those subsets induce a pseudo-random graph of members in which the correct members are connected and the diameter is logarithmic in the number of correct members, both with high probability. We have repeated the experiments above on a simulation of S-Fireflies and found slightly increased convergence times, but no more than one round of gossip.

S-Fireflies requires that each member has a public key certificate issued by a certification authority, further increasing the difficulty for an adversary to launch a Sybil attack. In addition, S-Fireflies would allow permissioned deployment by only accepting a certain class of public key certificates.

Another type of collusion attack against our approach is for a group of rational or Byzantine participants to agree before beginning of an epoch on a block to propose in order to significantly increase the chances of that block emerging as the agreed-upon block. Such collusion is of course possible, if not common, in PoW and BFT approaches as well. However, we will see that there is little incentive to do so as the proceeds of agreeing on a block are evenly divided among the participants.

We will now address is why somebody would deploy a node to participate in the proposed blockchain. In other words, what incentivizes participants? Some incentive comes from trans-

action fees. We would like, however, to pay participants for their work just like miners get paid for their work, if probabilistically.

In general, it is hard to prove for participants that they have actively participated in the gossip protocol. We propose to use platforms such as Intel SGX to provide such proofs. We envision that certified gossip code is released. When deployed in an SGX enclave, the code will acquire a timestamp from a certified time source, gossip for 24 hours, and provide a proof of correct execution. This proof can then be used to obtain renumeration in the next 24 hours. We envision that a certain amount of funds is distributed daily to all participants who can prove (using SGX or otherwise) that they have run the code during the previous 24 hour period. Note that unlike proposed blockchain protocols based on SGX such as PoET / Sawtooth Lake (intelledger.github.io), we do not rely on SGX for the correctness of the protocol—we only rely on SGX to renumerate participants.

At this point, a gossip-based blockchain is just a preliminary proposal, another point in the design space of blockchains, combining low energy with relatively open membership. Although it can most likely be sped up, in its proposed form it has the same (low) throughput as the Bitcoin blockchain but uses significantly less energy. Latency is at most 10 minutes, whereas for Bitcoin the latency until a block can be trusted to persist is closer to an hour. However, particularly in a permissionless environment, much analysis and refinement will need to happen to ensure that gossip-based blockchains can be trusted for critical applications.

References

[1] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC'87)*, 1987.

[2] E. Hoch, D. Dolev, and R. van Renesse. Self-stabilizing and Byzantine-tolerant overlay network. In *Proceedings of the 11th International Conference On Principles Of Distributed Systems (OPODIS'07)*, volume LNCS 4878. Springer, December 2007.

[3] H. Johansen, R. van Renesse, Y. Vigfusson, and D. Johansen. Fireflies: Secure and scalable membership and gossip service. *ACM Transactions on Computer Systems*, 33(5), June 2015.

[4] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-based failure detection service. In *Middleware'98, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*, September 1998.

WATCHBLOG
U.S. GOVERNMENT ACCOUNTABILITY OFFICE
FOLLOWING THE FEDERAL DOLLAR

Our New Science, Technology Assessment, and Analytics Team

Posted on January 29, 2019 by WatchBlog
https://blog.gao.gov/2019/01/29/our-new-science-technology-assessment-and-analytics-team/

Today we launched a new Science, Technology Assessment, and Analytics (STAA) team, expanding our work on cutting-edge science and technology issues.
STAA will focus on:

1. Technology assessments and technical services for the Congress,
2. Auditing federal science and technology programs,
3. Compiling and utilizing best practices in the engineering sciences, including cost, schedule, and technology readiness assessment, and
4. Establishing an audit innovation lab to explore, pilot, and deploy new advanced analytic capabilities, information assurance auditing, and emerging technologies that are expected to greatly impact auditing practices.

Watch our video, featuring U.S. Comptroller General Gene Dodaro and STAA's Managing Directors Tim Persons (GAO's Chief Scientist) and John Neumann, to learn about how this team will enhance our ability to help Congress oversee federal science and technology programs.

Enhancing and expanding our work
GAO routinely provides analysis of how federal agencies manage and employ science and technology, such as regenerative medicine, 5G wireless communication, and quantum computing.
In addition to our more traditional audit work, we've also conducted technology assessments for nearly two decades. These forward-looking analyses examine the

potential benefits and challenges of emerging technologies, such as artificial intelligence.
STAA will combine and enhance our technology assessment functions and our science and technology evaluation into a single, more prominent office to better meet Congress' growing need for information on these important issues. We plan to fill the team's roster with both experienced staff and new hires, so look out for future job postings.
Visit our site to learn more about STAA.

- Questions on the content of this post? Contact Tim Persons at personst@gao.gov or John Neumann at neumannj@gao.gov.
- Comments on GAO's WatchBlog? Contact blog@gao.gov.

ITPro Today

Letting Data Defend Itself: Benefits of Data-Centric Security

Fluree co-CEO Brian Platz discusses why data-centric security is now the best way to store and protect data.

Karen D. Schwartz | Jul 29, 2019
https://www.itprotoday.com/data-security-and-encryption/letting-data-defend-itself-benefits-data-centric-security

As we transition to a data-centric world, data is becoming more vulnerable. Startup Fluree believes the answer lies in more closely integrating storage and security, to the point where they are symbiotic.

Fluree co-CEO Brian Platz discusses the concept of data-centric security and letting data defend itself—and why it's so important in 2019.

ITPro Today: Your company is based on the premise that the traditional approach to data storage and protection doesn't work well anymore. Why?

Platz: Today, data doesn't just talk to one application. It talks to many—sometimes hundreds or thousands of applications. And data today is available to multiple customers, applications and partners. So the old approach to storage, which worked well in the app-centric world, doesn't work as well in today's data-centric world. We're struggling to figure out how to solve data security in a world where we have introduced a lot of vulnerabilities.

ITPro Today: What do you mean by letting data defend itself?

Platz: If you can embed data security with what you're using to store the data—typically a database—you're essentially letting the data defend itself. It requires creating data security rules for every application that is distributing information to the data, and keeping everything in sync.

If you have multiple applications talking to the same database, for example, those data security rules can be reproduced identically across every application. As a result, queries will dynamically filter the data based on the user connecting to the data. In other words, it's about using a tool to store managed data that has everything you need to allow users to connect to it without having to worry about leaking data or having invalid updates.

ITPro Today: As opposed to having separate tools or systems for storage and security?

Platz: Exactly. All of the data security is coded into the application tier, which has the root access to the database. If put the security and the data-centric rules alongside the data being stored, you have centralized the security around the information, and it will automatically be changed and updated as needed in one place.

ITPro Today: How would the data-centric rules work?

Platz: In our platform, you can write code stored as data that can create rules to enforce the security around it. Essentially, we're providing a programming language to embed right in the database that controls access, and the code you're writing itself is also treated as data, so it's managed with the same security as the rest of the data. And we actually store and manage the data as a blockchain, which brings a lot of integrity to the information. You can't possibly manipulate the data without detection. You can't even change a period in historical data without it being flagged as having been tampered with.

ITPro Today: Can organizations find ways to integrate data storage and security without buying a specific solution like Fluree?

Platz: Sure. The main way to do it is through APIs. That's the way most of us share data today; we build an API. The issues are that APIs are rigid, which lends organizations to creating a lot of them. It's easy to end up with hundreds of APIs very quickly. But building and maintaining APIs can be expensive, not only to build and test it but to maintain it, because your data rules and what you're storing and managing changes. If you want to change the security rules around your data, or you are storing more or less data or storing data differently, it changes everything. You have to know every single place where code has been written that enforces security around the information and update it at every one of those places in the identical way.

ITPro Today: Are there any other options?

Platz: Facebook's GraphQL is a way to address the problem of the explosion of APIs, and there are plenty of open-source tools around the GraphQL interface. GitHub, for example, now has a GraphQL interface, which can be used to replace multiple API endpoints.

ITPro Today: Are there other benefits to essentially marrying data storage and security other than the increased security itself?

Platz: Many of us have spent our whole lives being fearful of [data access], and we put layers of firewalls in front of our databases to protect the data. But this actually opens up the possibility that your database can be more valuable, because it has the proper protection. You can have richer interfaces where people can describe the data they want out of the system and it will come back in the exact shape and parameters that they described. They can just describe how they want the data, hit it once basically through this interface and get it.

ITPro Today: With all of these options, there is clearly a way around the data-centric security problem. What advice would you offer IT professionals on how to do it right?

Platz: Think about everything with a data-first mentality when building applications instead of an application-first approach. Think about applications as a portal into the data. This goes a long way toward increasing security, because you're not maintaining security in as many places. It also reduces cost because you don't have to build a lot of API endpoints. And it allows you to be more collaborative around your data with your partners and consumers. They can even update directly if you give them the permission to do so.

CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

Nathan Dowlin[1,2], Ran Gilad-Bachrach[1], Kim Laine[1], Kristin Lauter[1], Michael Naehrig[1], and John Wernsing[1]

[1]Microsoft Research, Redmond, USA
[2]Princeton University, New-Jersey, USA

February 24, 2016

Abstract

Applying machine learning to a problem which involves medical, financial, or other types of sensitive data, not only requires accurate predictions but also careful attention to maintaining data privacy and security. Legal and ethical requirements may prevent the use of cloud-based machine learning solutions for such tasks. In this work, we will present a method to convert learned neural networks to *CryptoNets*, neural networks that can be applied to encrypted data. This allows a data owner to send their data in an encrypted form to a cloud service that hosts the network. The encryption ensures that the data remains confidential since the cloud does not have access to the keys needed to decrypt it. Nevertheless, we will show that the cloud service is capable of applying the neural network to the encrypted data to make encrypted predictions, and also return them in encrypted form. These encrypted predictions can be sent back to the owner of the secret key who can decrypt them. Therefore, the cloud service does not gain any information about the raw data nor about the prediction it made.

We demonstrate *CryptoNets* on the MNIST optical character recognition tasks. CryptoNets achieve 99% accuracy and can make more than 51000 predictions per hour on a single PC. Therefore, they allow high throughput, accurate, and private predictions.

1 Introduction

Consider a hospital that would like to use a cloud service to predict the probability of readmission of a patient within the next 30 days, in order to improve the quality of care and to reduce costs. Due to ethical and legal requirements regarding the confidentiality of patient information, the hospital might be prohibited from using such a service. In this work we present a way by which the hospital can use this valuable service without sacrificing patient privacy. In the proposed protocol, the hospital encrypts the private information and sends it in encrypted form to the prediction provider, referred to as *the cloud* in our discussion below. The cloud is able to compute the prediction over the encrypted data records and sends back the results that the hospital can decrypt and read. The encryption scheme uses a public key for encryption and a secret key (private key) for decryption. It is important to note that the cloud does not have access to the secret key, so it cannot decrypt the data nor can it decrypt the prediction. The only information it obtains during the process is that it did perform a prediction on behalf of the hospital. Hence, the cloud can charge the hospital for its services, but does not learn anything about the patient's medical files or the predicted outcomes. This procedure allows for private and secure predictions without requiring the establishment of trust between the data owner and the service provider. This may have applications in fields such as health, finance, business, and possibly others.

It is important to note that this work focuses on the inference stage. We make the assumption that the cloud already has a model. In our case it would be a neural network that was previously trained, for example, using a set of unencrypted data. Training models for these kinds of applications might be a challenge as well due to the same concerns regarding privacy and security. The problem of training such a model is sometimes referred to as *privacy preserving data-mining* (Agrawal & Srikant, 2000). One possible solution to training while preserving privacy lies

in the concept of *differential privacy* (Dwork, 2011). Working with a statistical database, differential privacy allows one to control the amount of information leaked from an individual record in a dataset. Therefore, when training, one can use this concept to ensure privacy for any entity whose information is contained in the dataset as well as to create models that do not leak this information about the data they were trained on. However, the notion of differential privacy is not useful in the inference phase since at this stage, we are interested in examining a single record. Other options include working on encrypted data for training as well, in which case either simple classification techniques should be used (Graepel et al., 2013), or other assumptions on the data representation should be made (Aslett et al., 2015a).

The main ingredients of CryptoNets are *homomorphic encryption* and *neural networks*. Homomorphic encryption was originally proposed by Rivest et al. (1978) as a way to encrypt data such that certain operations can be performed on it without decrypting it first. In his seminal paper Gentry (2009) was the first to present a *fully* homomorphic encryption scheme. The term "fully homomorphic" means that the scheme allows arbitrarily many operations to be performed on the encrypted data. Gentry's original scheme was highly inefficient, but since then the work of several researchers has produced significantly more practical schemes. In this work, in particular, we use the homomorphic encryption scheme of Bos et al. (2013), which is very closely related to the schemes in López-Alt et al. (2012); Stehlé & Steinfeld (2011). This scheme is a *leveled* homomorphic encryption scheme, which allows adding and multiplying encrypted messages but requires that one knows in advance the complexity of the arithmetic circuit that is to be applied to the data. In other words, this cryptosystem allows to compute polynomial functions of a fixed maximal degree on the encrypted data. High degree polynomial computation requires the use of large parameters in the scheme, which results in larger encrypted messages and slower computation times. Hence, a primary task in making practical use of this system is to present the desired computation as a low-degree polynomial. We refer the reader to Dowlin et al. (2015); Bos et al. (2013); López-Alt et al. (2012) for details on the encryption scheme, and only give a brief introduction to it in Section 3. We used the *Simple Encrypted Arithmetic Library (SEAL)* for homomorphic encryption[1].

To allow accurate predictions we propose using neural networks, which in recent years have shown great promise for a wide variety of learning tasks. Much of the revival in the interest in neural networks is due to the unprecedented accuracy achieved in tasks such as image classification (Krizhevsky et al., 2012) and speech recognition (Dahl et al., 2012). In Section 2 we present a brief background on neural networks, as well as the necessary adjustments for them to work with homomorphic encryption, thus creating CryptoNets.

One line of criticism against homomorphic encryption is its inefficiency, which is commonly thought to make it impractical for nearly all applications. However, combining together techniques from cryptography, machine learning and software engineering, we show that CryptoNets may be efficient and accurate enough for real world applications. We show that when CryptoNets are applied to the MNIST dataset, an accuracy of 99% can be achieved with a throughput of 51739 predictions per hour on a single PC, and a latency of 570 seconds. Note that a single prediction takes 570 seconds to complete, however, the same process can make 8192 predictions simultaneously with no added cost. Therefore, over an hour, our implementation can make 51739 predictions on average. Hence, CryptoNets are accurate, secure, private, and have a high throughput - an unexpected combination in the realm of homomorphic encryption.

2 Neural Networks

The goal of this work is to demonstrate the application of neural networks over encrypted data. We use the term neural networks to refer to artificial feed-forward neural networks. These networks can be thought of as leveled circuits. Traditionally, these levels are called layers and are visualized as being stacked so that the bottom-most layer is the input layer. Each node of the input layer is fitted with the value of one of the features of the instance at hand. Each of the nodes in the following layers computes a function over the values of the layer beneath it. The values computed at the top-most layer are the outputs of the neural network.

Several common functions can be computed at the nodes. We have listed some of them here:

1. *Weighted-Sum* (convolution layer): Multiply the vector of values at the layer beneath it by a vector of weights and sum the results. The weights are fixed during the inference processes. This function is essentially a dot product of the weight vector and the vector of values of the feeding layer.

2. *Max Pooling*: Compute the maximal value of some of the components of the feeding layer.

[1] Available at `http://sealcrypto.codeplex.com`

3. *Mean Pooling*: Compute the average value of some of the components of the feeding layer.
4. *Sigmoid*: Take the value of one of the nodes in the feeding layer and evaluate the function $z \mapsto {}^{1}/_{(1+\exp(-z))}$.
5. *Rectified Linear*: Take the value of one of the nodes in the feeding layer and compute the function $z \mapsto \max(0, z)$.

Since homomorphic encryption supports only additions and multiplications, only polynomial functions can be computed in a straightforward way. Moreover, due to the increased complexity in computing circuits with nested multiplications, it is desired to restrict the computation to low-degree polynomials. The weighted-sum function can be directly implemented since it uses only additions and multiplications. Moreover, the multiplications here are between precomputed weights and the values of the feeding layer. Since the weights are not encrypted, it is possible to use the more efficient plain multiplication operation, as is described in Section 3.2.1. Some networks also add a bias term to the result of the weighted sum. To add this bias term a plain addition can be used since, again, the value of this bias term is known to the cloud.

One thing to note is that the encryption scheme does not support floating-point numbers. Instead, we use fixed precision real numbers by converting them to integers by proper scaling, although there are also other ways to do this (Dowlin et al., 2015). Furthermore, the encryption scheme applies all of its operations modulo some number t, which is why it is important to pay attention to the growth in the size of the numbers appearing throughout the computation, and to make sure that reduction modulo t does not occur. Otherwise the results of the computation might be unexpected. In our experiments, $5 - 10$ bits of precision on the inputs and weights of the network were sufficient in maintaining the accuracy of the neural network. All the numbers computed were smaller than 2^{80}, which guided us in selecting the parameters for the encryption scheme as seen in Section 3.2.5.

Max pooling cannot be computed directly since the max-function is non-polynomial. However, powers of it can be approximated due to the relation $\max(x_1, \ldots, x_n) = \lim_{d\to\infty} \left(\sum_i x_i^d\right)^{1/d}$. To keep the degree small, d should be kept reasonably small, with the smallest meaningful value $d = 1$ returning a scalar multiple of the mean pooling function. We will use this scaled mean-pool function instead of the max-pool function, as the sum $\sum x_i$ is easy to compute over encrypted data. The reason we use the scaled mean-pool instead of the traditional mean-pool is that we prefer not having to divide by the number of elements, although this could in principle be done (Dowlin et al., 2015). The only effect of not dividing is that the output gets scaled by a factor, which then propagates to the next layers.

The sigmoid and the rectified linear activation functions are non-polynomial functions. The solution of Xie et al. (2014) was to approximate these functions with low-degree polynomials, but we take a different approach here. We try to control the trade-off between having a non-linear transformation, which is needed by the learning algorithm, and the need to keep the degree of the polynomials small, to make the homomorphic encryption parameters feasible. We chose to use the lowest-degree non-linear polynomial function, which is the square function: $\mathrm{sqr}(z) := z^2$. It is interesting to note that Livni et al. (2014) have recently suggested a theoretical analysis of the problem of learning neural networks with polynomial activation functions and devoted much of their study to the square activation function.

AIn conclusion, to make a network compatible with homomorphic encryption some modifications are needed. Preferably, these modifications should be taken into account while training. The activation functions should be replaced by polynomial activation functions and the max pooling replaced by scaled mean pooling. For the sake of time-efficient evaluation, consecutive layers that use only linear transformations, such as the weighted-sum or mean pooling, can be collapsed.

3 Homomorphic Encryption

Encrypting data is a prominent method for securing and preserving privacy of data. Homomorphic encryption (HE) (Rivest et al., 1978) adds to that the ability to act on the data while it is still encrypted. In mathematics, a *homomorphism* is a *structure-preserving* transformation. For example, consider the map $\Phi : \mathbb{Z} \to \mathbb{Z}_7$ such that $\Phi(z) := z \pmod 7$. This map Φ preserves both the additive and multiplicative structure of the integers in the sense that for every $z_1, z_2 \in \mathbb{Z}$, we have that $\Phi(z_1 + z_2) = \Phi(z_1) \oplus \Phi(z_2)$ and $\Phi(z_1 \cdot z_2) = \Phi(z_1) \otimes \Phi(z_2)$ where $\oplus$ and $\otimes$ are the addition and multiplication operations in $\mathbb{Z}_7$. The map Φ is a ring homomorphism between the rings $\mathbb{Z}$ and $\mathbb{Z}_7$.

In the context of homomorphic encryption, we will be interested in preserving the additive and multiplicative structures of the *rings* of plaintexts and ciphertexts in the encryption and decryption operations. Since the first such encryption scheme was introduced (Gentry, 2009), there have been many advances in this field (see e.g. Naehrig et al. (2011); Gentry et al. (2012a,b); López-Alt et al. (2012)).

Technically speaking, (fully) homomorphic encryption allows for an arbitrary number of addition and multiplication operations to be performed on the encrypted data. For the sake of efficiency, we will instead use a weaker variant of this idea often called leveled homomorphic encryption, where the parameters of the encryption scheme are chosen so that arithmetic circuits of (roughly speaking) a predetermined depth can be evaluated. In our case this amounts to knowing the structure of the neural network, including the activation functions. The particular encryption scheme that we employ is YASHE', described in Bos et al. (2013).

3.1 Description of the method

The encryption scheme of Bos et al. (2013) (also López-Alt et al. (2012)) maps plaintext messages from the ring $R_t^n := \mathbb{Z}_t[x]/(x^n+1)$ to the ring $R_q^n := \mathbb{Z}_q[x]/(x^n+1)$. See Appendix A.1 for a brief introduction to rings and their properties. The encryption scheme chooses random polynomials $f', g \in R_q^n$, and defines $f := tf' + 1$. The public key h is defined to be $h := tgf^{-1}$, while f is the secret key. Since not every element in R_q^n is invertible, these steps are iterated until the corresponding f has an inverse and h can be computed.

A message $m \in R_t^n$ is encrypted by computing

$$c := \left[\lfloor q/t \rfloor m + e + hs\right]_q$$

where e and s are random noise polynomials in R_q^n, with coefficients of small absolute value. We use the notation $[a]_q$ (resp. $[a]_t$) to denote the reduction of the coefficients of a modulo q (resp. t) to the symmetric interval of length q (resp. t) around 0. Decrypting is done by computing

$$m := \left[\left\lfloor \frac{t}{q} fc \right\rceil\right]_t .$$

Here the product fc is first computed in R_q^n, the coefficients are interpreted as integers, scaled by t/q, and rounded to the nearest integers. Finally they are interpreted modulo t.

Two ciphertexts c_1 and c_2, with underlying messages m_1 and m_2, can be added together in R_q^n to yield the encryption of $m_1 + m_2$. This works because

$$\begin{aligned} c_1 + c_2 = &\lfloor q/t \rfloor (m_1 + m_2) + (e_1 + e_2) \\ &+ h(s_1 + s_2) , \end{aligned} \tag{1}$$

which decrypts to $m_1 + m_2 \in R_t^n$.

To multiply two messages we first compute

$$\left\lfloor \frac{t}{q} c_1 c_2 \right\rceil = \lfloor q/t \rfloor (m_1 m_2) + e' + h^2 s_1 s_2 \tag{2}$$

where e' is a noise term that under the right conditions is still small. It is easy to see that the term above decrypts to $m_1 \cdot m_2$, but under the secret key $f^2 \in R_q^n$. Using a process called *relinearization* (see e.g. (Brakerski & Vaikuntanathan, 2011; Bos et al., 2013)), it is possible to modify the result so that it will be decryptable under the original secret key.

3.2 Practical considerations

The first thing to note is that the method described above works as long as the noise terms appearing in the encryptions of m_1 and m_2 are small enough. Otherwise the decryptions might not yield correct answers. The security level of the system depends on the parameters n, q, t, and the amount of noise added. The maximum amount of noise that a ciphertext can have and still be decryptable depends on the parameters q and t.

When ciphertexts are added or multiplied, the noise in the resulting ciphertext is typically larger than in the inputs. Noise growth is particularly strong in multiplication. This essentially means that the parameter q should be selected to be large enough to support the increased noise, which necessitates choosing a larger n for security reasons.

If the computation to be performed is expressed as an arithmetic circuit with addition and multiplication nodes, the main limitation to using the scheme is the number of multiplication gates in the path from the inputs to the outputs. This number we refer to as the *level*. Keeping the level low allows for selecting smaller values for the parameters,

which results in faster computation and smaller ciphertexts. Note that the level is not the same as the degree of a polynomial to be evaluated, and instead behaves like the logarithm of the degree.

While keeping the parameters small improves performance for our tasks, we would like to make t large to prevent the coefficients of the plaintext polynomials from reducing modulo t at any point during the computation. To better understand this point, note that the atomic objects used in a neural network are real numbers. The neural network takes as its input a vector of real numbers and, through a series of additions, multiplications, and other real functions, it computes its outputs, which are also real numbers. However, the homomorphic encryption scheme works over the ring $R_t^n := \mathbb{Z}_t[x]/(x^n+1)$. This means that some conversion process between real numbers and elements of R_t^n is needed. We refer to such conversions as encodings (real numbers to R_t^n) and decodings (R_t^n to real numbers). If the coefficients of a polynomial in R_t^n are reduced modulo t after say, an addition, there is usually a problem with decoding it correctly to the sum of the real numbers, and instead the result is likely to be unexpected. This is why we need to keep track of how large the coefficients of the plaintext polynomials grow throughout the entire computation, and choose the parameter t to be larger.

To make the computations faster, it is also important to keep track of what parts of the data need to be secured. A common task that is repeatedly performed in the neural network is computing the weighted some of the inputs from the previous layer. While the data from the previous layer is encrypted, the weights are known to the network in their plain form. Therefore, when multiplying the data by the weights we can use a more efficient form of multiplication, described below.

3.2.1 Plain operations

In applying neural networks a common operation is to add or multiply some value, which is derived from the data with some known constant. The naive way to implement such operations is to first encrypt the constant and then perform the addition or multiplication operation. However, this process is both computationally intensive and adds a large amount of noise if the operation is multiplication. However, this is not necessary. Let $c = \lfloor q/t \rfloor m + e + hs$ be the encrypted message and w the known constant. Addition can be achieved by multiplying w by $\lfloor q/t \rfloor$ and adding that to c, which results in $\lfloor q/t \rfloor (m+w) + e + hs$. This is essentially just encrypting w with no noise and performing normal homomorphic addition.

For multiplication, even the scaling is not needed since $cw = \lfloor q/t \rfloor mw + e' + hs'$. This is very efficient, especially if w is a sparse polynomial. For example, if w is a scalar (as it would be in the scenario below), then this multiplication is computed in linear time in the degree of c, which is $n-1$.

3.2.2 Encoding

As we already discussed above, there is a mismatch between the atomic constructs in neural networks (real numbers), and the atomic constructs in the homomorphic encryption schemes (polynomials in R_t^n). An encoding scheme should map one to the other in a way that preserves the addition and multiplication operations. Such an encoding scheme can be constructed in several ways. For example, it is possible to convert the real numbers to fixed precision numbers, and then use their binary representation to convert them into a polynomial with the coefficients given by the binary expansion. This polynomial will have the property that when evaluated at 2, it will return the encoded value. Another option is to encode the fixed precision number as a constant polynomial. This encoding is simple, but might seem inefficient in the sense that only one coefficient of the polynomial is being used. In Section 3.2.4 we show how a batch of such instances can be encoded simultaneously to make use of the entire space. One problem with such a *scalar encoding* is that the only coefficient of the message polynomials grows very rapidly when homomorphic operations are performed

3.2.3 Encoding large numbers

As we have already explained, a major challenge for computing in this encryption scheme lies in preventing the coefficients of the plaintext polynomials from overflowing t. This forces us to choose large values for t, which causes the noise to grow more rapidly in the ciphertexts and decreases (with q fixed) the maximum amount of noise tolerated. Therefore, we need to choose a larger q, and subsequently a larger n for security reasons. One way to partially overcome this issue is by using the Chinese Remainder Theorem (CRT) (See Appendix A.2). The idea is to use multiple primes $t_1, \ldots, t_k$. Given a polynomial $\sum a_i x^i$ we can convert it to k polynomials in such a way that the j-th polynomial is $\sum [a_i \pmod{t_j}] x^i$. Each such polynomial is encrypted and manipulated identically. The

CRT guarantees that we will be able to decode back the result, as long as its coefficient does not grow beyond $\prod t_j$. Therefore, this method allows us to encode exponentially large numbers while increasing time and space linearly in the number of primes used.

3.2.4 Parallel Computation

The encryption uses polynomials of a high degree. For example, in our case $n = 8192$, both ciphertext and plaintext polynomials can have degrees up to 8191. If the data is encoded as a scalar, only one out of the 8192 coefficients is being used, while all the operations (additions and multiplications) act on the entire 8192 coefficient polynomials. Therefore, the operations are slow due to the high degree, but the result contains only a single significant coefficient. Another application of the CRT allows us to perform Single Instruction Multiple Data (SIMD) operations at no extra cost Gentry et al. (2012b). Assume that t is selected such that $x^n + 1 \equiv \prod (x - \alpha_i) \pmod t$. In this case the CRT can be used to show that $R_t^n \cong \mathbb{Z}_t^{\times n}$. The isomorphism is explicit and easy to compute, which means that we can encode n values into a single polynomial, operate on this polynomial, and decode the n different results.

Note that we use here the CRT in an opposite direction to how we use it when encoding large numbers (Section 3.2.3). When encoding large numbers, we take a single number and break it into multiple small numbers that are being processed in parallel and joined together at the end. On the other hand, here we take multiple scalars and join them together to form a single polynomial. This polynomial is being processed as a single unit and only upon completing the computation is it broken into its components.

3.2.5 Parameter Selection

The main parameters defining the cryptosystem are the plaintext modulus t, the coefficient modulus q and the degree n of the polynomial modulus $(x^n + 1)$. To allow for encoding large enough numbers for the purposes of the network, we used two plaintext moduli and both of the CRT techniques described above. The values used are $t_1 = 1099511922689$ and $t_2 = 1099512004609$. They were selected so that their product is greater than 2^{80}, which is large enough for applying the network. Moreover, they are small enough so that with the coefficient modulus $q = 2^{383} - 2^{33} + 1$ and the polynomial modulus $x^{8192} + 1$ allow for computing the desired network correctly, i.e. so that the noise does not grow too large. Finally, the plaintext moduli are chosen such that

$$x^{8192} + 1 = \prod_{i=0}^{8191} \left(x - \alpha_i^{1,2}\right) \pmod{t_{1,2}}$$

In other words, the polynomial modulus breaks into linear components, which allows for optimal use of the SIMD technique described in Section 3.2.4.

4 Empirical Results

We have tested CryptoNets on the MNIST dataset (LeCun et al., 1998). This dataset consists of 60,000 images of hand written digits. Each image is a 28x28 pixel array, where each pixel is represented by its gray level in the range of 0-255. We used the training part of this dataset, consisting of 50,000 images, to train a network and the remaining 10,000 images for testing. The details of the network used are presented in Table 1. The accuracy of the training network is 99% (it mislabels only 105 out of the $10,000$ test examples).

4.1 Timing analysis

Since the network can accept batches of size 8192 (due to the choice of the degree $n = 8192$ in the encryption parameters) we timed the network on the first 8192 images of the test set to match the batch size. The latency of the network is governed by the time to process a batch while the throughput is also a function of the batch size. Therefore, we separated the report for these two parameters and also reported on the time per instance. These results are presented in Table 2.

Applying the network takes 570 seconds on a PC with a single Intel Xeon E5-1620 CPU running at 3.5GHz, with 16GB of RAM, running the Windows 10 operating system. Since applying the network allows making 8192 predictions simultaneously using the SIMD operations as described in Section 3.2.4, this PC can sustain a throughput

of $8192 \times 3600/570 \approx 51739$ predictions per hour. Encrypting the data takes 122 seconds and additional 0.060 seconds for every parallel instance to be encoded. Therefore, if 8192 instances are encoded, a throughput of 48068 instances per hour can be encrypted and encoded. Decrypting the data takes 5 seconds and additional 0.046 seconds to decode predictions for each instance. Therefore, a throughput of 77236 decryptions and decoding per hour is achievable with our setup.

4.2 Description of the Network

The network has two forms: the model that is the direct output of training, and the simplified version which is actually used for making predictions. The trained network has 9 layers, and the simplified version has 5 layers. A visualization of the latter is given in Table 1, though we will describe both of them here.

Here is a description of the network used for training:

1. *Convolution Layer*: The input image is 28×28. The convolution has windows, or kernels, of size 5×5, a stride of $(2, 2)$, and a mapcount of 5. The output of this layer is therefore $5 \times 13 \times 13$.
2. *Square Activation Layer*: This layer squares the value at each input node.
3. *Scaled Mean Pool Layer*: This layer has $1 \times 3 \times 3$ windows, and again outputs a multi-array of dimension $5 \times 13 \times 13$.
4. *Convolution Layer*: This convolution has a kernel size of $1 \times 5 \times 5$, a stride of $(1, 2, 2)$, and a mapcount of 10. The output layer is therefore $50 \times 5 \times 5$.
5. *Scaled Mean Pool Layer*: As with the first mean pool, the kernel size is $1 \times 3 \times 3$, and the output is $50 \times 5 \times 5$.
6. *Fully Connected Layer*: This layer fully connects the incoming $50 \cdot 5 \cdot 5 = 1250$ nodes to the outgoing 100 nodes, or equivalently, is multiplication by a 100×1250 matrix.
7. *Square Activation Layer*: This layer squares the value at each input node.
8. *Fully Connected Layer*: This layer fully connects the incoming 100 nodes to the outgoing 10 nodes, or equivalently, is multiplication by a 10×100 matrix.
9. *Sigmoid Activation Function*: This layer applies the sigmoid function to each of the 10 incoming values.

The sigmoid activation function is necessary for the training stage in order to get reasonable error terms when running the gradient descent algorithm. However, we don't have a good way of dealing with the sigmoid in the encrypted realm. Luckily, once we have our weights fixed and want to make predictions, we can simply leave it out. This is because the prediction of the neural network is given by the index of the maximum value of its output vector, and since the sigmoid function is monotone increasing, whether or not we apply it will not affect the prediction.

The other change that we make to the network is just for an increase in efficiency. Since layers 3 through 6 are all linear, they can all be viewed as matrix multiplication and composed into a single linear layer corresponding to a matrix of dimension 100 by $5 \cdot 13 \cdot 13 = 865$. Thus, our final network for making predictions is only 5 layers deep.

One obstacle to training networks using the square activation function is that, unlike the rectified linear and sigmoid functions, its derivative is unbounded. This can lead to strange behavior when running the gradient descent algorithm, especially for deeper nets it sometimes blows up or overfits. The overfitting issue can be partially resolved by the addition of convolution layers without activation functions (layers 4 and 5 in our network). This allows us to reduce the number of degrees of freedom in the output polynomial. However, for even deeper nets (10 to 20 layers) something else will be needed to aid in training.

4.3 Message sizes

The images consist of 28×28 pixels. Each pixel is encrypted as 2 polynomials (two values for t are used together with CRT to allow for the large numbers needed). Each coefficient in the polynomial requires 48 bytes and therefore, each image requires $28 \times 28 \times 8192 \times 2 \times 48$ bytes or 588 MB. However, the same message can contain 8192 images and therefore, the per image message size is only 73.5 KB. The response of the classifier contains only 10 values (for

Table 1: Breakdown of the time it takes to apply CryptoNets to the MNIST network

Layer	Description	Time to compute
Convolution layer	Weighted sums layer with windows of size 5×5, stride size of 2. From each window, 5 different maps are computed and a padding is added to the upper side and left side of each image.	46 seconds
1^{st} square layer	Squares each of the 835 outputs of the convolution layer	290 seconds
Pool layer	Weighted sum layer that generates 100 outputs from the 835 outputs of the 1^{st} square layer	195 seconds
2^{nd} square layer	Squares each of the 100 outputs of the pool layer	36 seconds
Output layer	Weighted sum that generates 10 outputs (corresponding to the 10 digits) from the 100 outputs of the 2^{nd} square layer	3 seconds

Table 2: The performance of CryptoNet for MNIST

Stage	Latency	Additional latency per each instance in a batch	Throughput
Encoding+Encryption	122 seconds	0.060 seconds	48068 per hour
Network application	570 seconds	0	51739 per hour
Decryption+Decoding	5 seconds	0.046 seconds	77236 per hour

the 10 possible digits) and therefore the message size is $10 \times 8192 \times 2 \times 48$ which is 7.5 MB or 0.94 KB per image, when 8192 images are encoded together. These numbers are summarized in Table 3.

It is interesting to put the message sizes used in comparison to natural raw representations of these messages. The size of the message depends on the representation used. For example, if each image is represented as an array of size 28x28 and each pixel is represented as a double precision floating point number, then the size of each image is approximately 6 KB, which is 12 times smaller than the encrypted version. More concise representation is possible if only a single byte is used to represent each pixel, which will bring the ratio between the encrypted size to the unencrypted size to 96. The sparsity of the data allows compressing the data even further, and indeed the compressed version of this dataset has an average of only 165 bytes per instance. Therefore, the encrypted version is $456\times$ larger than this compressed form. In conclusion, the encrypted data is one to three orders of magnitude larger than the unencrypted data. The exact factor depends on what is considered a natural representation of the data in its raw form.

5 Discussion and Conclusions

The growing interest in *Machine Learning As a Service* (MLAS), where a marketplace of predictors is available on a pay-per-use basis, requires attention to the security and privacy of this model. Not all data types are sensitive, but in many applications in medicine, finance, and marketing the relevant data on which predictions are to be made is typically very sensitive.

Different methods can be used to protect the data. For example, the prediction provider and the data owner can encrypt the data while in transit using traditional cryptography. These methods are promising in terms of throughput, latency, and accuracy, but they require some way to establish trust between the cloud and the data owner. The provider also needs to guarantee the safety of the keys, and the safety of the data against attackers while it is stored in the cloud.

Another possible approach would be using secure *Multi-Party Computation* (MPC) techniques (Goldreich, 1998). Most MPC methods establish a communication protocol between the parties involved, such that if the parties follow

Table 3: Message sizes of CryptoNet for MNIST

	Message size	Size per instance
Owner $\rightarrow$ Cloud	588 MB	73.5 KB
Cloud $\rightarrow$ Owner	7.5 MB	0.94 KB

the protocol they will end with the desired results while protecting the security and privacy of their respective assets (Barni et al., 2006; Orlandi et al., 2007; Piva et al., 2008; Chen & Zhong, 2009). Barni et al. (2006) presented a method of this type, where the data owner encrypts the data and sends it to the cloud. The cloud computes an inner product between the data and the weights of the first layer, and sends the result to the data owner. The data owner decrypts, applies the non-linear transformation, and encrypts the result before sending it back to the cloud. The cloud can apply the second layer and send the output back to the data owner. The process continues until all the layers have been computed. In Orlandi et al. (2007) they also noted that this procedure leaks much of the information of the weights of the network to the data owner, and added a method to obscure the weights. The main difference between these methods and the method we describe in this paper is that in our method the data owner does not have to maintain a constant presence while the neural network is evaluated. For example, the data owner can encrypt the data and store it in the cloud in its encrypted form. The cloud can apply one or several networks to the data while the data owner is offline. Whenever the data owner wishes to read the predictions, it can retrieve the information and decrypt it, allowing the data owner to maintain a much simpler infrastructure. Moreover, since intermediate results are not shared, less information is leaked from the cloud to the data owner.

Graepel et al. (2013) suggested the use of homomorphic encryption for machine learning algorithms. They focused on finding algorithms where the training can be done over encrypted data, and therefore were forced to use learning algorithms in which the training algorithm can be expressed as a low degree polynomial. As a result, most of the algorithms proposed were of the linear discrimination type. Several authors also looked at nearest neighbor classification (Zhan et al., 2005; Qi & Atallah, 2008). However, linear classifiers and nearest neighbor classifiers do not deliver the same level of accuracy that neural networks are capable of delivering.

Aslett et al. (2015a,b) presented ways to train machine learning models over data encrypted with homomorphic encryption. They presented both simple algorithms, such as naive Bayes classifiers, as well as more involved random models such as random forests and some variations of it. Their work differs from our work in several major aspects: The models they propose work well on some tasks, but do not compete well with neural networks on tasks such as recognizing objects in images. They also had to use a unique coding scheme, in which values are compared to threshold before encryption, to allow the learning algorithm to work. CryptoNets imposes fewer requirements on the data owner, and allows the use of neural networks; however, it does not support training on the encrypted data.

Training neural networks over encrypted data is still possible. If all the activation functions are polynomials, and the loss function is polynomial too, back-propagation can be computed using additions and multiplications only. However, there are several challenges in doing so. Computational complexity is a major challenge. Even when trained on plaintext, neural networks are slow to train. Today, much of the effort in the field of machine learning goes towards accelerating this training process by using sophisticated hardware such as GPUs. However, adding homomorphic encryption to the process will make the process at least an order of magnitude slower. It is more likely that the slowdown would be much worse since the level of the computed polynomial is proportional to the number of back-propagation steps made, and therefore using leveled homomorphic encryption does not seem to be feasible. Another challenging aspect in the presence of encryption is the lack of ability of a data scientist to inspect the data and the trained models, to correct mislabeled items, to add features, and to tune the network.

The main contribution of this work is a method that enjoys the accuracy of neural networks with the simplicity of use of homomorphic encryption. By combining techniques from cryptography, machine learning, and engineering, we were able to create a setup in which both accuracy and security are achieved, while maintaining a high level of throughput. This work leaves much room for improvement, however. For example, the throughput and latency can be significantly improved by using GPUs and FPGAs to accelerate the computation. Another direction for further progress would be finding more efficient encoding schemes that allow for smaller parameters, and hence faster homomorphic computation.

References

Agrawal, Rakesh and Srikant, Ramakrishnan. Privacy-preserving data mining. In *ACM Sigmod Record*, pp. 439–450. ACM, 2000.

Aslett, Louis JM, Esperança, Pedro M, and Holmes, Chris C. Encrypted statistical machine learning: new privacy preserving methods. *arXiv preprint arXiv:1508.06845*, 2015a.

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

Barni, Mauro, Orlandi, Claudio, and Piva, Alessandro. A privacy-preserving protocol for neural-network-based computation. In *Proceedings of the 8th workshop on Multimedia and security*, pp. 146–151. ACM, 2006.

Bos, Joppe W, Lauter, Kristin, Loftus, Jake, and Naehrig, Michael. Improved security for a ring-based fully homomorphic encryption scheme. In *Cryptography and Coding*, pp. 45–64. Springer, 2013.

Brakerski, Zvika and Vaikuntanathan, Vinod. Efficient fully homomorphic encryption from (standard. In *LWE, FOCS 2011, IEEE 52nd Annual Symposium on Foundations of Computer Science, IEEE*. Citeseer, 2011.

Chen, Tingting and Zhong, Sheng. Privacy-preserving backpropagation neural network learning. *Neural Networks, IEEE Transactions on*, 20(10):1554–1564, 2009.

Dahl, George E, Yu, Dong, Deng, Li, and Acero, Alex. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.

Dowlin, Nathan, Gilad-Bachrach, Ran, Laine, Kim, Lauter, Kristin, Naehrig, Michael, and Wernsing, John. Manual for using homomorphic encryption for bioinformatics. Technical report, Microsoft Research, 2015. `http://research.microsoft.com/apps/pubs/default.aspx?id=258435`.

Dwork, Cynthia. Differential privacy. In *Encyclopedia of Cryptography and Security*, pp. 338–340. Springer, 2011.

Eisenbud, David. *Commutative Algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media, 1995.

Gentry, Craig. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pp. 169–178, 2009.

Gentry, Craig, Halevi, Shai, and Smart, Nigel P. Fully homomorphic encryption with polylog overhead. In *Advances in Cryptology–EUROCRYPT 2012*, pp. 465–482. Springer, 2012a.

Gentry, Craig, Halevi, Shai, and Smart, Nigel P. Homomorphic evaluation of the aes circuit. In *Advances in Cryptology–CRYPTO 2012*, pp. 850–867. Springer, 2012b.

Goldreich, Oded. Secure multi-party computation. *Manuscript. Preliminary version*, 1998.

Graepel, Thore, Lauter, Kristin, and Naehrig, Michael. Ml confidential: Machine learning on encrypted data. In *Information Security and Cryptology–ICISC 2012*, pp. 1–21. Springer, 2013.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yan, Cortes, Corinna, and Burges, Christopher J.C. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

Livni, Roi, Shalev-Shwartz, Shai, and Shamir, Ohad. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pp. 855–863, 2014.

López-Alt, Adriana, Tromer, Eran, and Vaikuntanathan, Vinod. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1219–1234. ACM, 2012.

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

Orlandi, Claudio, Piva, Alessandro, and Barni, Mauro. Oblivious neural network computing via homomorphic encryption. *EURASIP Journal on Information Security*, 2007:18, 2007.

Piva, Alessandro, Orlandi, Claudio, Caini, M, Bianchi, Tiziano, and Barni, Mauro. Enhancing privacy in remote data classification. In *Proceedings of The Ifip Tc 11 23rd International Information Security Conference*, pp. 33–46. Springer, 2008.

Qi, Yinian and Atallah, Mikhail J. Efficient privacy-preserving k-nearest neighbor search. In *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*, pp. 311–319. IEEE, 2008.

Rivest, Ronald L, Adleman, Len, and Dertouzos, Michael L. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

Stehlé, Damien and Steinfeld, Ron. Making ntru as secure as worst-case problems over ideal lattices. In *Advances in Cryptology–EUROCRYPT 2011*, pp. 27–47. Springer, 2011.

Xie, Pengtao, Bilenko, Misha, Finley, Tom, Gilad-Bachrach, Ran, Lauter, Kristin, and Naehrig, Michael. Crypto-nets: Neural networks over encrypted data. *arXiv preprint arXiv:1412.6181*, 2014.

Zhan, Justin Zhijun, Chang, LiWu, and Matwin, Stan. Privacy preserving k-nearest neighbor classification. *IJ Network Security*, 1(1):46–51, 2005.

A Commutative Algebra

Many of our results rely on concepts in commutative algebra that might be unfamiliar to some readers. In this section we provide some background on the concepts used in this paper. We refer the reader to Eisenbud (1995) for a comprehensive introduction to the field.

A.1 Rings

A commutative ring R is a set on which there are two operations defined: addition and multiplication, such that there is $0 \in R$ which is the identity element for addition and $1 \in R$ which is the identity for the multiplication operation. For every element $a \in R$ there exists an element $-a \in R$ such that $a + (-a) = 0$. Furthermore, the following hold for every $a, b, c \in R$:

$$\begin{aligned} a\,(bc) &= (ab)\,c\,; \\ a\,(b+c) &= ab + ac\,; \\ (a+b)\,c &= ac + bc\,; \\ a + (b+c) &= (a+b) + c\,; \\ a + b &= b + a\,; \\ ab &= ba\,. \end{aligned}$$

Since all the rings we discuss in this work are commutative rings, we use the term "ring" to refer to a "commutative ring".

Several rings appear in this work. The set $\mathbb{Z}$ of integers is a ring, as is the set $\mathbb{Z}_m$ of integers modulo m, whose elements can be thought of as sets of the form $\{i + am : a \in \mathbb{Z}\}$, where i is an integer. When we write $k \in \mathbb{Z}_m$, we mean the set $\{k + am : a \in \mathbb{Z}\}$. Conversely, we say that $k \in \mathbb{Z}$ represents, or is a representative of, this element of $\mathbb{Z}_m$.

The set $R[x]$ of polynomials with coefficients in a ring R is itself a ring. In this work we deal a lot with the ring $\mathbb{Z}_m[x]$ of polynomials with integer coefficients modulo m. Finally, the set $\mathbb{Z}_m[x]/(x^n+1)$, whose elements can be thought of as sets of the form $\{p(x) + q(x)(x^n+1) : q(x) \in \mathbb{Z}_m[x]\}$, where $p(x) \in \mathbb{Z}_m[x]$, is a ring. When we write $r(x) \in \mathbb{Z}_m[x]/(x^n+1)$ we mean the set $\{r(x) + q(x)(x^n+1) : q(x) \in \mathbb{Z}_m[x]\}$, and conversely say that $r(x)$ represents, or is a representative of, this element of $\mathbb{Z}_m[x]/(x^n+1)$. The polynomials with coefficients in some fixed set of representatives of elements of $\mathbb{Z}_m$, and of degree at most $n-1$, form a complete set of representatives of elements of $\mathbb{Z}_m[x]/(x^n+1)$. To simplify the notation, we refer to the ring $\mathbb{Z}_m[x]/(x^n+1)$ as R_m^n.

A.2 Chinese Remainder Theorem (CRT)

An element $p \in R$ is said to be prime if for every $f, g \in R$ it is true that if p divides fg, then p divides at least one of f and g. The Chinese Remainder Theorem states that $R \cong \prod_i R/(p_i)$ when the $p_1, \ldots, p_n$ are distinct primes. This should be interpreted as follows: An element $r \in R$ is uniquely represented by elements $r_1, \ldots, r_n$ such that $r_i \in R/(p_i)$. This allows breaking $r \in R$, which might be large (in some sense), into n "small" values $r_1, \ldots, r_n$. At the same time, it is also true that every $r_1, \ldots, r_n$ such that $r_i \in R/(p_i)$ has a unique $r \in R$ that represents it. This allows us to pack n "small" values $r_1, \ldots, r_n$ into a single large value $r \in R$.

The Chinese Remainder Theorem can be written in an explicit "constructive" form. The transformation from R to $\prod_i R/(p_i)$ is the easier one, and is simply given by sending r to the the sets $\{r + qp_i : q \in R\}$ for each i. In the other direction, given $r_1, \ldots, r_n$, they can be mapped to $\sum q_i r_i$, where $q_i \in R$ are such that for every $j \neq i$, p_j divides q_i, and p_i divides $q_i - 1$. The values of the q_i can be computed as follows: First let $\hat{q}_i := \prod_{j \neq i} p_i$. Next, let $\hat{q}_i^{-1} \in R$ be such that p_i divides $\hat{q}_i \hat{q}_i^{-1} - 1$. This is always possible when the ideals (p_i) and (p_j) are coprime (Eisenbud, 1995), which is the case when R is the ring of integers, or a polynomial ring over integers modulo a prime number. Finally, let $p_i := \hat{q}_i \hat{q}_i^{-1}$.