Ownership White Paper Draft V.1.0

Korda C., Kai Chen, Snow Zhao

***Abstract*—Ownership is a decentralized secure data exchange and data computation platform aiming to create a blockchain-based data ownership protocol on top of the OSI Model's Seven Layers of the Internet. By incorporating zero-knowledge proof, homomorphic encryption, and secure multi-party computation, Ownership delivers a secure and efficient data exchange and computation solution. Ownership's mission is to reshape the traditional competitive landscape defined by zero-sum games, build a new paradigm for data sharing and collaborative computing, and usher in a new era of business cooperation and economic prosperity.**

I. Introduction

A. Blockchain

In 2008, an article titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [1] was published under the pseudonym Satoshi Nakamoto[1]. This paper led to the creation of Bitcoin, the world's first decentralized digital currency. Bitcoin's decentralized nature is realized through blockchain technology, a form of distributed ledger.

Since the birth of the Internet in October 1969, there has been ongoing research on the feasibility of issuing a digital currency. However, it wasn't until 2008 when Satoshi created Bitcoin that the challenges of launching a decentralized digital currency were overcome. These challenges revolve around the nature of data, namely that data can spread easily and freely on the internet, making it difficult to manage and maintain data ownership. For example, if person A shares an image with person B, then person B can immediately share this picture with other people. Likewise, person A can continue to send the picture to other people as well. This model of free data flow works great for the spread of information, but not for digital currency. This is because in order for something to qualify as money, it must have the two basic attributes of not being able to be copied and not being able to be spent twice. Any digital currency must also satisfy these two conditions in order to qualify as money. The traditional model is to rely on a third-party intermediary to carry out transaction settlement in order to guarantee these conditions. The innovation of blockchain technology is that it allows any two individuals on the internet to transfer value to one another without the need for third-party institutions to intervene.

The Byzantine Generals' Problem is a fundamental problem in peer-to-peer information networks. In distributed computing, computers in the network reach consensus through information exchange, but system failures or erroneous information might compromise system consistency [2]. Aside from quantum communication, blockchain technology is also an effective solution to this problem. Blockchain technology is no longer simply a public ledger, but a public computer. This means that anyone in the world can not only store data on the immutable blockchain, but also run programs on the blockchain as well. The underlying logic of all programs running on the blockchain is unanimously approved by all users to ensure consistency, security, and fairness.

[1]Inventor of Bitcoin

B. Ownership

What is (ownership[2])?

1) Ownership Definition: From a legal point of view, ownership is the right of all persons to possess, use, profit and dispose of their property in accordance with the law. China's classical text *The Book of Lord Shang*[3] described ownership based on the theory of "Dingfen Zhizheng", the fixing of rights and duties to prevent disputes of ownership. During the Qin Kingdom's reformation, Duke Xiao of Qin inquired legalist statesman Yang Shang regarding how to govern his kingdom. Yang Shang replied:

> That a hundred men will chase after a single hare that runs away, is not for the sake of the hare; for when they are sold everywhere on the market, even a thief does not dare to take one away, because their legal title is definite. Thus if the legal title is not definite, then even men like Yao, Shun, Yu or Tang would all rush to chase after it, but if the legal title is definite even a poor thief would not take it.

To understand the idea of "Dingfen Zhizheng", one need only think of the difference between a wild rabbit chased by many and a caged rabbit chased by none. The reason one is chased by many and the other chased by none is because of the concept of ownership, which is precisely the meaning of "fen" in "Dingfen Zhizheng". If legal ownership is definite, then property cannot be legally obtained through brute force, and social order will be established. If ownership is not definite, then properties can be obtained through brute force, and society will descend into chaos. In effect, Shang Yang's governance principles are based on the basic idea of ownership [3]. This historical event shows that as early as the Spring and Autumn Warring States period, people understood the value and function of ownership.

In a society, ownership is the foundation of many human activities. The function of ownership is not to simply define property and wealth, but to provide basic social stability

[2]"ownership" refers to the general term while "Ownership" refers to this project

[3]A renowned Legalist text written during the Spring and Autumn Warring States period

and social order. The Coase Theorem[4], proposed by Nobel Prize Winning Economist Ronald H. Coase, states that explicit property rights is necessary for optimal resource allocation through market forces. Ownership in the traditional sense has been clearly defined and is well understood, but how ownership should be defined in the Age of the Internet remains an open question.

2) Data Ownership: In July of 2017, Mark Zuckerberg announced that Facebook's monthly active users exceeded 2 billion, which is close to a quarter of the world's population and more than half of all internet users [4]. With the development of social networks, e-commerce, and mobile internet, the amount of data has increased exponentially. In fact, the amount of data that is being generated is so large that we need a new unit of measurement called the Petabyte (PB)[5].

In the age of big data, data is just as valuable an asset as land, labor, capital, and the aforementioned "rabbit". But how is data ownership defined? Data ownership can be understood through the framework of property ownership, and thus include having the right to control the data and claim the profits generated from the data. Control over data means being able to add, delete, change, and query the data. The principle of data ownership is that data belongs to those who have ownership over it, and data that is not owned by anyone are public resource.

One major difference between data and the "rabbit" described in the excerpt above is that data is composed of 0s and 1s in the computer network, which makes it extremely difficult to clearly assign ownership. The first difference is that data can be easily copied by nature, with the copies being the same as the original, while the "rabbit" in *The Book of Lord Shang* cannot be duplicated. Secondly, anyone with a copy of the data now has the same rights over the data as the original owner, and the original owner of the data loses control of this data. As a result of these two challenges, data owners are reluctant to exchange data with each other and must rely on a trusted third party for such exchanges. However, the profit-seeking nature of these centralized third parties is calling their neutrality and credibility into question. These trusted third parties are storing data in the absence of authorization, neglecting data privacy, and even engaging in data fraud.

According to Steven Levy's book *Crypto*[6], cryptographer and Turing Award winner Whitfield Diffie[7] put forth the concept of a "decentralized view of authority" in the 1960s with the aim of building cryptographic tools to solve the problem of data security during data exchange [5]. Bitcoin, the first decentralized digital currency in the history of mankind, has definitively shown the power of decentralization. In his first speech on Bitcoin [6], Andreas Antonopoulos[8] summarized the power of Bitcoin:

> Bitcoin is fundamentally different because with bitcoin you don't owe anyone anything and no one owes you anything. It is not a system based on that. It is a system based on ownership and no one can censor it, no one can seize it, no one can freeze it.

The Bitcoin system is built on blockchain technology, a technology that presents the solution to data ownership.

II. Ownership System

A. System Overview

On November 5, 2002, the famous novelist Neal Stephenson published *Cryptonomicon*[9], a novel thought by many to hold a "profound prophecy". This novel predicted the creation of cryptocurrencies in the beginning of the 21st century, and that cryptocurrencies would have a profound impact on our society. The invention of Bitcoin seems to confirm the author's prediction. The author even boldly envisions a "data sanctuary" in Southeast Asia - a place where encrypted data can be freely stored and exchanged [7].

The design of the Ownership System combines Diffie's "decentralized view of authority" with Stephenson's "data sanctuary". Diffie's "decentralized view of authority" became a reality with the birth of blockchain technology. The Ownership System's Ownership Blockchain incorporates Diffie's "decentralized view of authority" through its lottery-based consensus algorithm while the Ownership System's Ownership Engine implements Stephenson's "data sanctuary" through privacy-preserving data computation.

B. Account System

Ownership's account system is one of the core components of the entire system, responsible for data control within the Ownership System. Unlike the single-password single-account system used in general information systems, or the pure dual-account system (internal and external accounts) adopted by Ethereum, the Ownership account system implements a dual account multi-password verification model that supports authorized access control of the system's data.

1) Account System Functionality: The functionalities of the Ownership account system is shown in the following list. These functionalities serve as the starting point for the design of the Ownership account system.

1. Subject of Ownership
2. Unique Data Controller
3. Smart Contract Authorization Controller
4. Multi-party Signature Access Authorization

2) Account System Components: Ownership's account system implements a dual account multi-password verification model that consists of four main components: the account lock system, account lock source file, account lock port, and Ownership wallet. First, the account lock system grants authorization for transactions or data that need to be

[4] An economic theorem that describes the economic efficiency of an economic allocation or outcome in the presence of externalities

[5] A unit of data measurement equivalent to 10 to the 15th power bytes of data

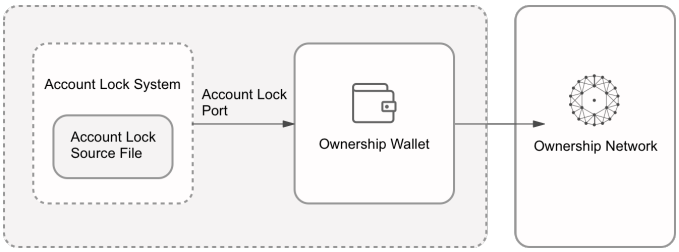
[6] A novel written by Steven Levy published in 2002

[7] Renowned cryptographer and security expert, inventor of public key encryption

[8] Security expert and author, Bitcoin core developer

[9] A novel by Neal Stephenson published in 2002

authorized. The account lock source file is the data source that enables the account lock system. It consists of six variable groups, and serves as a unique ID: account system version, account authentication type, account public key, account private key, account expansion data, and operation sequence number. Ownership wallets are nodes in the Ownership public blockchain, and is responsible for functionalities such as initiating data signing and creating and verifying transactions. The account lock port is the communication interface between the account lock system and the Ownership wallet, a framework for signature and encryption.



III. Consensus Algorithm

A. The Need for a New Blockchain

This section lays out the justification for a radically new blockchain. The success of Bitcoin has raised public awareness of the value of blockchain technology. Since the introduction of the Bitcoin blockchain, there have been many innovations such as colored coins, smart contracts, and new consensus algorithms. However, the technology still faces some fundamental challenges, including limitations in performance, high energy consumption, risks of forking, and a tendency to centralize.

The Bitcoin blockchain is currently experiencing some major issues. The Bitcoin blockchain produces a block every ten minutes, and it takes an hour or more to confirm a transaction. In addition, the proof-of-work (PoW) consensus algorithm requires a large amount of wasted computing power and electricity. Moreover, the rise of large mining pools is leading to a centralization of authority, in opposition to the decentralized intent of cryptocurrencies. While innovative solutions such as side-chain technology and lightning network have been designed to overcome these issues, fundamental problems of computation power concentration and performance limitations are yet to be resolved.

Even the most active blockchain, Ethereum, requires 15 seconds to produce a block with throughput per second being in the single digits. In addition, there isn't necessarily just one globally-consistent Bitcoin blockchain. In fact, there often exists many forks of the blockchain containing the latest transaction data. For example, user A might see the block combination $\beta_1, \beta_2 ... \beta_k, \beta_{k+1}$ on the blockchain while user B sees the block combination $\beta_1, \beta_2 ... \beta_k, \beta^w_{k+1}, \beta^w_{k+2}$ on the blockchain. Only when the blocks k+1 and k+2 have been verified and added to the chain can the block be assumed to be consistent among all users. As a result of this uncertainty, the transactions in the latest blocks cannot immediately be assumed to be completed.

As blockchain technology moves beyond cryptocurrencies and into industry and consumer applications, the above challenges become serious limitations in widespread adoption of blockchain technology. Hence the need for a blockchain that is designed with real-world applications in mind, a blockchain that is democratic and fully decentralized, offers high computing performance, and protects data privacy.

B. Survey of Consensus Algorithms

This table compares the pros and cons of the most common consensus algorithms currently used. While each offer a unique value proposition, none solves the fundamental problems inhibiting the widespread adoption of blockchain technology.

Name	Principle	Examples	Advantages	Disadvantages
PoW	Solve cryptographic problem to obtain the right to record transactions	Bitcoin Litecoin Primecoin	Simple and Secure	Computationally and energy intensive Inefficient and hard to scale Mining pool centralization
PoS	Ownership of the cryptocurrency confers the right to record transactions	Peercoin Blackcoin Nxt	Energy-saving Highly efficient	Susceptible to security threats such as nothing-at-stake Must rely on a security deposit
DPoS	Cryptocurrency owners vote for delegates who record the transactions	BitShares	Requires fewer nodes to reach consensus Consensus is reached quickly	Low participation leads to concentration of power
PBFT	A majority of the nodes reach agreement	HyperLedger	Highly efficient	High cost of information propagation Not suitable for public blockchains Vulnerable to malicious attacks

C. Lottery Consensus Algorithm

1) Overview: There are three distinct roles in the Lottery Consensus Algorithm: the leader, the verifier, and the recorder. The leader is responsible for creating a new block. The verifiers are responsible for reaching consensus on the leader and the corresponding block. The recorder is responsible for verifying and storing the data [8].

The three-step Lottery Consensus Algorithm works as follows:

1. The set of candidate leader nodes is randomly selected according to the distribution of θ's hash value
2. The verifiers reach consensus on the leader and the corresponding block through PBFT
3. The recorder node verifies the validity of the leader and confirm that the verifiers have reached consensus, and updates his own chain accordingly

2) Cryptographic Sortition: First, select the set of candidate leader nodes.

Parameter Description:
$r \geq 0$: Current block number
$s \geq 1$: Current consensus step of block r
S: Private key signature of node i
$Hash()$:Hash function, with randomly distributed results
p: Expected range of results of the hash function
θ^r Input for the hash function
θ^0: Specified in the Creation Block

For each block r, current consensus step s, and the θ value of the previous block, each node i in the rth block uses its own private key to sign and hash the block. If the hash value is less than the parameter p, the node i belongs to the set of candidate leader nodes. θ^0 is specified in the creation block. The calculation of the θ value is based on the current consensus stage and the node's own information, independent of the transaction information in order to guarantee that the θ value is random.

$$Hash(S_i(r, s, \theta^{r-1})) \leq p \tag{1}$$

Second, select a leader from the set of candidate leaders. The candidate leader nodes will collectively broadcast their current θ value and their proposed block r. Each candidate leader will verify the θ value of other candidate leaders and select the node with the smallest θ value as the potential leader. When 2/3 of the nodes in the verifier set reach consensus, the node i with the smallest θ value of the current block r becomes the leader, and the proposed block r is broadcasted to the whole network once it has been verified and consensus has been reached. The recorder node verifies that the leader node's signature is valid and that the set of verifiers has indeed reached consensus on the block, upon which the block is included into the recorder's own chain.

$$\theta^r := Hash(S_i(\theta^{r-1}), r) \tag{2}$$

3) PBFT: The Lottery Consensus Algorithm applies PBFT (Practical Byzantine Fault Tolerance) [9] in its consensus algorithm. In 1999, Miguel Castro and Barbara Liskov proposed PBFT to reduce the complexity of the algorithm from exponential to polynomial.

Parameter Description:
pre': Pre-prepare the message identifier
pre: Prepare the message identifier
v: The address of the node that broadcasts the message
n: The serial number of the block awaiting consensus
d: Summary of the block awaiting consensus
m: The current block awaiting consensus and its θ value
i: The address of the node that forwards the message

The three stages of PBFT are the following:

1. Pre-prepare: Broadcast the pre-prepared message to the set of verifier nodes [[pre-prepare,v,n,d],m]
2. Prepare: Broadcast the prepared message to the set of verifier nodes [prepare,v,n,d,i]
3. Confirm: Once f+1 of verifiers' pre-prepared messages are received and 2f + 1 prepared message are verified, the set of verifiers reach a consensus on the leader node and its proposed block.

When the set of verifiers cannot reach a consensus, an empty block r is generated, and each node recalculates its θ value. The purpose of re-conducting leader and verifier selection is to avoid being unable to reach a consensus as a result of a leader node being offline.

$$\theta^r := Hash(\theta^{r-1}, r) \tag{3}$$

IV. Economic Model

A. Internal Ownership Model

This section will show that the system is internally consistent. A key innovation of the Ownership System is that all of the participants in the system can be reduced to a single class of users. All users, whether firms performing computations on the Ownership Engine or coin-owners transferring coins in the network, are interacting with the Ownership Blockchain through a smart contract. As described in section 5 of this white paper, blocks are created and consensus is reached by this same group of users, meaning that the concept of dedicated miners do not exist in this system. The system selects a leader to create the block and chooses a group of verifiers to confirm the leader and the block proposed by the leader, therefore the users are also themselves the miners. The following lays out the incentive model for rewarding users to perform the roles of leader and verifier. The incentive for the leader and the verifiers of any given block can be computed mathematically.

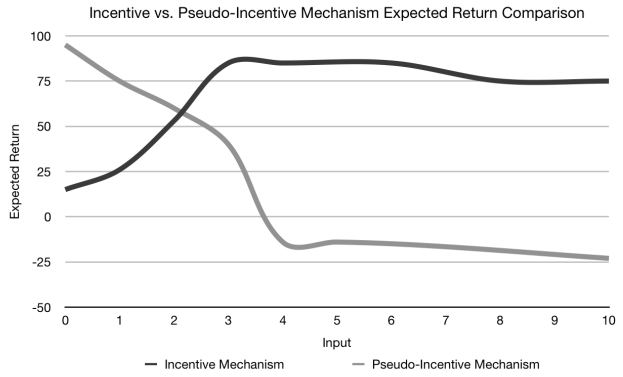
Parameter Description:
I^r: Total incentive for a given block
$\bar{f}^r$: Average fee (gas) of processing a smart contract in a given block
N^r: Number of contracts processed in a given block
I_s^r: Incentive given by the Ownership System for a given block
Putting everything together, total incentive for a given block:

$$I^r = \bar{f}^r \times N^r + I_s^r \tag{4}$$

By observing the above equation, the following four conditions need to be met to ensure that the Ownership System is internally economically consistent:
1. The total incentive I^r is sufficient to encourage users to create and verify blocks
2. The total incentive I^r is limited to dissuade purposeful mining behavior
3. The fee $\bar{f}^r$ is capped at a level to encourages system adoption and usage

4. System incentive I_s^r compensates for the potentially low $\bar{f}^r$ and/or N^r

We propose a pseudo-incentive mechanism that combines the economic models of public blockchains, the free model of P2P file download, and the traditional lottery revenue model. The pseudo-incentive mechanism is different from the free model or the model of giving users large incentives. For each individual user, the pseudo-incentive mechanism provides an opportunity to earn a relatively high return. However, centralization of computation power is avoided since even if the winning probability is increased by buying a large number of "lottery tickets" (shown as input in the chart below), the total expected return is reduced.



B. External Business Model

The goal of this section is to validate the external consistency of the Ownership System with the broader business model of profit-maximizing firms, with a specific focus on cooperative behavior, whether internal to an organization or externally across organizations. In deciding whether or not to cooperate, a department or a firm weighs the tradeoff between the marginal benefits and the marginal costs of such cooperation. In the current business framework, there exist abundant cases in which cooperation is mutually beneficial but does not materialize due to external factors such as high costs, high risk, and political issues. The following analysis shows that the Ownership System changes the economics of cooperative behavior so that profit-maximizing firms will choose to cooperate in cases where they heretofore did not.

First, lay out the following assumptions:

1. Firms are profit-maximizing, meaning that they will make decisions with the goal of maximizing profits for the firm and their shareholders
2. Political issues might not be the main cause of non-cooperation
3. Firms factor in all costs of cooperation, both financial and non-financial (i.e. as opportunity cost), into their profit-maximization function
4. The costs of cooperation can be so high that firms cannot benefit from cooperation

Next, denote the terms required for mathematical analysis:

P: Firm profit
R: Firm revenue
C: Firm cost
^{N}: Current framework
^{O}: Ownership System framework
$_c$: Cooperation
$_n$: Non-cooperation

Under the current model, profit-maximizing firms compare the following two options and choose the one that maximizes firm profit, P.
Profit with cooperation:

$$P_c^N = R_c^N - C_c^N \tag{5}$$

Profit without cooperation:

$$P_n^N = R_n^N - C_n^N \tag{6}$$

If firms choose not to cooperate, then it must be true that:

$$C_c^N - C_n^N > R_c^N - R_n^N \tag{7}$$

In other words, the marginal costs of cooperation are greater than the marginal benefits, therefore firms choose not to cooperate. In order for firms to change their behavior in such a situation, either the marginal cost of cooperation must decrease or the marginal benefit of cooperation must increase.

The Ownership System Framework changes firm cooperative behavior by altering the underlying economic calculus of cooperation. For simplification, assume that the Ownership System Framework does not alter the revenue and cost calculations of the non-cooperating case. This implies that the only factors that are variable in the profit model laid out above are the revenue of cooperation R_c^O and the cost of cooperation C_c^O.
Assuming firms choose cooperation, then it must be true that:

$$R_c^O - C_c^O > R_n^O - C_n^O \tag{8}$$

The marginal revenue from cooperation is greater than the marginal cost of cooperation, therefore firms choose to cooperate. The revenue of cooperation R_c^O and the cost of cooperation C_c^O under the Ownership System Framework can be analyzed independently. The revenue of cooperation is impacted through two main channels, namely the Effective Effect and the Ownership Effect. The Effective Effect refers to the marginal increase in revenue of existing forms of cooperation resulting from a more efficient cooperation framework. For example, by guaranteeing data privacy, cooperating parties can now design a more effective co-marketing campaign based on valuable sensitive data that they were previously unwilling to share. The Ownership Effect is the general increase in value of all cooperation projects. The Ownership System enables new forms of cooperation that were heretofore economically unviable, widening areas of cooperation, creating synergies and

deepens trust, ultimately increasing the overall benefit from cooperation. Each new cooperation project not only has a direct impact on the revenue of the project itself but also indirectly impacts all other projects.

The two main components of the cost of cooperation C_c^O, namely financial cost and non-financial cost, are both decreased in the new framework. Financial costs, which include costs such as the labor, system, and computation costs of data acquisition, data verification, data processing, and other tasks, are reduced through system simplifications and process eliminations enabled under the Ownership System Framework. The new framework has an even bigger impact on the non-financial costs of cooperation, such as the time needed to establish mutual trust and the risks of data security and data misuse. The decentralized secure privacy-preserving computation engine resolves the above issues as a matter of default.

In conclusion, the Ownership System increases the revenue of cooperation and decreases the cost of cooperation, thus enabling new forms of cooperation and expands the set of scenarios in which profit-maximizing firms will choose to cooperate, both internally and externally.

V. Ownership Engine

A. Overview

The Ownership Engine is a platform for running decentralized privacy-preserving applications. The core purpose of the Ownership Engine is to enable multi-party collaborative computing while allowing all parties to retain full control over data ownership. The Ownership Engine consists of three components:

1. OVM (Ownership Virtual Machine): Smart contract execution engine, supports homomorphic encryption and other cryptographic primitives
2. OOP (Ownership Oracle Protocol): Standardized data exchange protocol governing secure smart contract data exchange execution
3. OAF (Ownership Application Framework): Decentralized privacy-preserving application development framework containing cryptographic function libraries

The Ownership Engine integrates blockchain technology with cryptographic technologies such as zero-knowledge proof, homomorphic encryption, and secure multi-party computation to enable the rapid development and deployment of privacy-preserving smart contracts and decentralized data computation applications.

B. Zero Knowledge Proof

Introduced by S.Goldwasser and C.Rackoff in the 1980s [10], zero-knowledge proof is a cryptographic methodology by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true. Zero knowledge proof is an effective cryptographic method of establishing a secure privacy-preserving protocol. To understand zero-knowledge proof, start by defining an interactive proof system.

Interactive Proof System: a pair of interactive machines [P,V], where P and V represent the prover and the verifier respectively, can be considered as an interactive proof system of language L if they satisfy the following requirements:

1. Machine V is polynomial time
2. Completeness: for any $x \in L$, there exist an honest prover P, so that the system outputs $x \in L$ when the verifier V completes the interaction with P
3. Soundness: for any $x \notin L$ and any prover P, the system output $x \in L$ with negligible probability when the verifier V completes the interaction with P

A zero knowledge proof system is an interactive proof system that meets the requirements of zero knowledge proof and must possess the following four attributes:

1. The verifier cannot obtain any information from the protocol
2. The prover cannot deceive the verifier
3. The verifier cannot deceive the prover
4. The verifier cannot simultaneously disguise him/herself as a prover in another zero knowledge proof system

Zero knowledge proof is particularly suitable for use cases which require privacy-preservation. Zerocash, a privacy-preserving version of bitcoin, is a classic example of the application of zero knowledge proof. Zerocash is the first blockchain system to integrate zero knowledge proof, providing full confidentiality in peer-to-peer payments. The system does not reveal the origin, destination, or amount of the transaction, while allowing authorized queries regarding transaction details through private keys.

The Ownership Engine provides a zero-knowledge proof security service layer at the architectural level for smart contracts and decentralized applications. This security service layer ensures privacy protection during data computation, such as for zero knowledge identification verification, transaction data security, and so on.

C. Homomorphic Encryption

The topic of homomorphic encryption was first introduced by Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos in 1978 [11], but it was only in 2009 that Craig Gentry proved the first homomorphic algorithm [12]. Homomorphic encryption is a form of encryption that allows computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

Let E(m) be the encrypted cyphertext of m, if E(a) and E(b) are known, then anyone can obtain the cyphertext of $a \oplus b$ through some sort of operation denoted by $E(a) \otimes E(b)$, and homomorphism can be generally expressed as:

$$E(a \oplus b) = E(a) \otimes E(b) \tag{9}$$

where $\oplus$ and $\otimes$ represent the binary operations of plaintext and ciphertext spaces respectively.

Homomorphic encryption includes operations such as addition, subtraction, multiplication, and division. Achieving both

additive homomorphism and multiplicative homomorphism means being able to run all operations, a state known as algebraic homomorphism.

Homomorphic encryption is a key cryptographic technology for the development of blockchain technology and the adoption of blockchain-based applications. Due to current security concerns, users do not perform computations on sensitive data directly on the blockchain. Therefore, the introduction of a homomorphic encryption technology suited for practical use can relieve user concerns over data security and drive the adoption of blockchain-based data computation.

Although current homomorphic encryption technology is computationally-prohibitive for large scale commercial use on large datasets, it is invaluable for the deployment of smart contracts in specific security-sensitive use cases involving small data sets. The Ownership Engine has a native homomorphic encryption operator built into the Ownership Virtual Machine to implement homomorphic cryptosystems including Palllier, Benaloh, RSA, and ElGamal. With this homomorphic encryption-powered Ownership Virtual Machine, the Ownership Engine enables the rapid development and deployment of decentralized privacy-preserving applications.

D. Secure Multi-Party

The traditional method of completing a computation involving data from multiple sources/parties is to aggregate the data in a single location for centralized computation. While this method works for some cases, it fails in situations that lack a single entity with enough authority and credibility to obtain data from all participating parties. The following lists a couple of such situations:

1. Alice suspects that she might have a genetic disease and knows that Bob has a DNA database that has categorized a wide variety of genetic diseases. Alice wants to find out if she indeed has the disease, but the only way to do so is to send her DNA sequence to Bob so that Bob can perform a diagnosis. However, Alice is concerned about her privacy and does not want to disclose her own DNA information and diagnostic results. It appears as if she must choose between her health and her privacy, both of which are important to Alice. With secure multi-party computation, she can get a reliable diagnosis without compromising her privacy.

2. Company A is considering expanding into a certain market X, but is concerned that company B might also be planning to enter market X. Both Company A and Company B want to avoid competing in the same region. Therefore, they want to verify that their market expansion plans do not overlap, but do not want to reveal the specific markets that they are targeting. Revealing such sensitive information can be very costly since Company A can preemptively act on Company B's plans and vice versa. Moreover, should such information leak, a third competitor can also execute on those plans, or the real estate developer can charge distortionary rates. With secure multi-party computation, Company A and Company B can achieve their goal of not competing in market X without revealing their actual market expansion plans.

The above two examples share a common characteristic, namely that two or more parties want to engage in collaborative computation that requires the input of their private data, but none of the parties is willing to divulge their private data. The technical question becomes how to complete such computations while protecting the private information of all parties. This general problem is known as the secure multi-party computation problem.

Secure multi-party computation was proposed by the Turing Award winner A.C. Yao in the 1980s [13]. The main objective of secure multi-party computation is to complete the following computational task: In a trust-less distributed network, two or more parties can collaboratively perform agreed upon computations and retrieve the results of such computations, all while guaranteeing privacy-preservation. Secure multi-party computation has important applications in the fields of collaborative scientific computing, privacy-preserving database query, privacy-preserving data mining, privacy-preserving data analysis, and more.

Although O.Goldreich, S.Micali and A.Wigderson proposed a cryptographic computation protocol that can calculate any function [14], its real-world application is limited because the protocol runs a large number of zero-knowledge proofs which require users to transfer large amounts of data. Therefore, the key to improving the applicability of secure multi-party computation is to design protocols tailored to meet targeted use cases. The Ownership Engine categorizes the various use cases of secure multi-party computation and build computation protocols into the blockchain infrastructure in order to meet the privacy-preservation and data computation needs of different industries and use cases.

VI. Use Cases

A. Health Care Data

As medical data migrate from paper records to electronic records, medical institutions and patients have accumulated a wealth of health care data. As data continues to accumulate, it becomes increasingly imperative that this data be analyzed and shared in order to improve health care services. The need to better leverage and apply existing medical and health care data is resulting from a few major developments:

1. Patients are demanding a better and more individualized end-to-end health care experience
2. The low-hanging fruits in pharmaceutical RD are gone
3. Patient concerns over data security and personal data privacy are increasing

In response to the above challenges, Ownership System provides a fundamentally new framework. Take the case of a pharmaceutical company that is developing a new drug and in need of patient data for clinical trials. Under the current regime, the cost of doing so is high, not least because patients have concerns over their privacy and the security of their data. With guaranteed privacy and data security as well as autonomous control over their data, patients will be a lot more likely to share/sell their data to the pharmaceutical company. The smart contract can also simultaneously handle the payment attached to the sell of the patient's data. This pharmaceutical company is thus able to obtain the data that it

needs at a much lower cost, from both a financial and non-financial perspective. This is a win-win situation in which the pharmaceutical company increases the efficiency of drug development by acquiring valuable patient data and the patient earns an income from their data without sacrificing privacy.

B. Other Use Cases

The above example is not the only use case of the Ownership System. In fact, the potential applications are only limited by imagination. Developers can develop various decentralized applications based on the Ownership Blockchain and/or the Ownership Engine. Developers can also decide whether to incorporate privacy protection and data security functionalities into their decentralized applications.

VII. VISION

The sharing of data can generate tremendous value for society. In today's age of big data, machine learning, and artificial intelligence, data is the "food" for machine learning algorithms, but this "food" is locked in the cellar of big company "landlords". The data is not sufficiently utilized because of concerns such as data security, privacy protection, data ownership and so on. Through the Ownership System, we hope to truly implement Neil Stephenson's vision of a "data granary" shared by all of mankind.

ACKNOWLEDGMENT

Here and now, we would like to extend our sincere thanks to all those who have helped us make this white paper possible. First and foremost, we are deeply grateful to the inventor of Bitcoin, Satoshi Nakamoto. His extraordinary work introduced blockchain technology to the world, set the foundation for a decentralized future, and served as the inspiration for the Ownership Project. Next, we want to thank MIT Professor and Turing Award recipient Silvio Micali for introducing a new democratic consensus algorithm, Algorand, with the aim of achieving true decentralization. Finally, this project would not have been possible without the support of our friends, particularly cryptographer and security expert Dr. Steven He and protocol guru Mr. Bryan Q.

REFERENCES

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

[2] Vukoli, Marko. "The Byzantine empire in the intercloud." *ACM SIGACT News* 41.3 (2010): 105-111.

[3] Duyvendak, Jan Julius Lodewijk. *The Book of Lord Shang*. Probsthain, 1928.

[4] Constine, Josh. "Facebook now has 2 billion monthly users and responsibility. TechCrunch, TechCrunch, 27 June 2017, techcrunch.com/2017/06/27/facebook-2-billion-users/. Accessed 17 July 2017.

[5] Levy, Steven. Crypto: secrecy and privacy in the new code war. London, Penguin, 2002.

[6] Antonopoulos, Andreas M. Blockchain vs. Bullshit: Thoughts on the Future of Money .

[7] Stephenson, Neal, and Jean Bonnefoy. *Cryptonomico*. Paris: Payot Rivages, 2000. Print.

[8] Micali, Silvio. "ALGORAND: the efficient and democratic ledger." *arXiv preprint arXiv:1607.01341* (2016).

[9] Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance." *OSDI*. Vol. 99. 1999.

[10] R.Riv est, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, Vol. 21, No 2, pp.120-126, 1978.

[11] Gentry C. Computing arbitrary functions of encrypted data[J]. *Communications of the ACM*, 2010, 53(3): 97-105.

[12] Goldwasser, Shafi, Silvio Micali, and Charles Rackoff. "The knowledge complexity of interactive proof systems." *SIAM Journal on computing* 18.1 (1989): 186-208.

[13] Yao, Andrew C. "Protocols for secure computations." Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982.

[14] Goldreich, Oded, Silvio Micali, and Avi Wigderson. "How to play any mental game." *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM, 1987.

DISCLAIMER