IOVO Technical Paper, v. 0.1

January 18, 2018

Abstract

IOVO (Internet Of Value Omniledger) is a global open protocol for the storing and exchange of information of all types on possible nonmonetary quantitative data assets. The IOVO DAG (next-generation blockchain) serves as a scoring (rating) ledger by creating a universal global decentralised ledger of all data, which then publishes scoring rates on both individuals and institutional entities across a range of sectors. Rating scores within IOVO are solely accessible to the scored entity itself, and are revealed upon request only as answers to particular questions (IOVO acting as a zero-knowledge-proof platform).

IOVO allows users and companies to own their data and decide what is revealed and on what terms via their personal scoring data wallet. Being the first truly transparently public and globally universal personal and institutional decentralised database, IOVO serves as a global ledger on every type of knowledge about individuals and all entities. It stores and supplies all possible scoring features (protocols) for different market and social applications. It is encrypted, secure, distributed, democratic and above all human-centric. That said, IOVO has the capability to reach a broader scope than just people. It is a database of all possible scoring scales and information, serving as a base layer for future decentralised data market makers.

TABLE OF CONTENTS

1. Structure of IOVO

1.1 Addresses

The IOVO system consists of users and transactions carried out between them. Every user is assigned a pair of cryptographic keys | a public key which serves as an address and a secret key used for signing transactions.

1.2 Transaction DAG

The backbone of IOVO is a system of fast and free-of-charge transactions. The transactions are organized in the directed acyclic graph (DAG) structure, a new concept first proposed in cryptocurrencies like Dagcoin/ Byteball [1] and IOTA [2]. In IOVO we use a structure similar to the one used in IOTA (they call it a tangle). However we have extended it with scripts that allow users to create and run Ethereum-like smart contracts [3] and scorings – IOVO's primary feature, which serves as a base layer for all possible nonmonetary value applications.

The transaction DAG can be seen as a natural successor of blockchain first introduced in Bitcoin [4], however it has no chain and no blocks. Instead, every transaction is a node (site) in a directed acyclic graph in which an edge A ! B means that transaction A directly approves transaction B. Unlike blockchain, in the case of a transaction DAG there is no need for miners to approve transactions. Instead each user, in order to issue a new transaction, must approve k_0 existing transactions[1]. For further details about the transaction DAG, please refer to [2].

2. Fundamental operations in IOVO

2.1 Sending currency and messages

The most basic type of transaction in IOVO is a transaction with value and a message. The user specifies how much IOVO-coins (IOVO currency) he wants to include in a transaction. He can also add a short (say, 128B) message to it. Since all the transactions are kept public, this message will be visible to all the network, so this can be used, as a unmodifiable ledger to write. On the other hand, users can easily use the same feature to send private messages to other

[1] We can think of $k_0 = 2$, but this parameter can be adjusted to the network capabilities.

users by encrypting the message with the public key of the recipient. Then only the recipient can decrypt the message with his secret key.

This basic type of transaction (with value and message only) is called a standard transaction and can be sent without any fee | users are "paying" for the transaction through approving other transactions in the DAG. Of course, transactions with 0 value are also admissible, so one can use IOVO as a platform for fast and free money transfers and messaging.

2.2 Scoring others

The main feature underlying all of IOVO's innovative applications is the scoring mechanism. Each scoring is a user-generated protocol[2] which collects partial scores about the target user and computes from them the user's final score. In the simplest case, a scoring can be a numeric-value scale in some category. Let's say there is an eBay-like application built on top of IOVO. Following each transaction, the buyer gives a partial score to the seller. Then the final score will simply be made up of the mean of all the partial scores. However, we can imagine much more complicated types of scorings, e.g.:

An insurance company runs on top of IOVO and every car incident involving a particular driver is recorded as a partial score in an IOVO transaction. Then the final score is the driver's insurance premium.

A bank (or a system of banks) runs on top of IOVO and every loan taken and every installment paid are recorded as a partial score in an IOVO transactions. Then the final score is the client's creditworthiness.

Moreover, the author of a scoring can also impose some parts of the score which do not necessarily come from IOVO transactions (e.g., some initial scores are given to the users, which is useful in the case where an institution moves its business to IOVO and wants to transfer the data which it has already gathered about its users. What is important, there is no single formula for computing the final score of a user | the scoring author can arbitrarily create all the rules and publish them as the scoring manifest.

It is worth noting that the score part of the transaction is free of charge, as well as its value and message. We will sometimes refer to the transactions containing a partial score part as scoring transactions.

2.2.1 Notation

We will denote all the scorings by S^1; S^2; : : :. The final score of user U in scoring S^i will be denoted by $S^i(U)$, and all the partial scores for user U in scoring S^i will be denoted by $S_1{}^i(U)$; $S_2{}^i(U)$; : : :. The partial scores are broadcast in the encrypted form using target user's public key pk_U :

$$E_j{}^i(U) = Enc_{pkU}\,(S_j{}^i(U))$$

Everyone can create a scoring and publish all its rules as a manifest. More on this will be covered later in the paper.

Therefore only U can compute his ownfinal score from the public information.

2.3 Script

Every IOVO transaction can have an Ethereum-like script attached to it. It means, in particular, that a transaction can, for example, create a contract or call up another contract.

Since the script can run long, script transactions are much more expensive for the network to handle. Therefore, a transaction with script requires k_S confirmations in the DAG (where k_S is some number greater than k_0 and proportional to the script length).

2.4 Requesting a score

The main building block of the IOVO system is the operation of requesting a score. In this operation, user A wants to know $S^i(B)$ | the score of user B in scoring S^i. This score depends on all the partial scores ($S_1^i(B)$; $S_2^i(B)$; : : :) that appear in the transaction history in an encrypted form ($E_1^i(B)$; $E_2^i(B)$; : : :). The only way to reveal the score is to evaluate the function $Reveal_{sk_B}$ ($E_1^i(B)$; $E_2^i(B)$; : : :), which can be done only knowing the secret key sk_B. Therefore the whole procedure of acquiring a score will consist of the following phases:

A collects all the partial scores ($E_1^i(B)$; $E_2^i(B)$; : : :) from the network.

A broadcasts a transaction Request(A; B; $E_1^i(B)$; $E_2^i(B)$; : : :) with C coins attached to it.

When B notices a Request transaction addressed to him, he (locally) computes his score using formula (s;) = $Reveal_{sk_B}$ ($E_1^i(B)$; $E_2^i(B)$; : : :), where s is the computed score and is a zero-knowledge proof that it was correctly computed on the given set of partial scores ($E_1^i(B)$; $E_2^i(B)$; : : :). The proof is important, since it is a guarantee, that B used all available data ($E_1^i(B)$; $E_2^i(B)$; : : :) and did not use any fake partial scores.

B broadcasts a transaction Claim(A; B; Enc_{pk_A} (s);) to claim the reward of c coins. Score s is encrypted with a public key of A to avoid other users seeing the result (recall that all the transactions are publicly visible).

The Request transaction can be implemented as a time-locked Ethereum-like smart contract | it transfer c coins to B only if B will broadcast a correct Claim transaction within some given time period t. Then the Claim transaction will be in fact sent to Request contract (it will call a contract function). In case the Claim transaction is not issued within t time, the money will automatically go back to A.

Thanks to the mechanism of requesting a score, IOVO users are incentivized to actively participate in the network | they will claim the reward for revealing their score only in cases where they react to a request within the timeframe window t. This is important since active users approving other transactions are necessary for the stability and security of the currency.

Moreover, to further increase the contribution of these “revealing users" to the network, the system might be implemented in such a way that calling a contract function (as in the case of sending a Claim transaction) requires approving $k_C > k_0$ other transactions. This can be done, e.g., using an Ethereum-like gas system | every low-level command in a contract code burns some gas which must be paid for (the currency for this payment is the number of other transactions to approve). Hence, complex transactions like Claim (contract calls) will require more other transactions to approve, while standard transactions (money transfers between users) will remain simpler (i.e. require only k_0 other transactions being approved).

2.5 Search engine

The mechanism of requesting a score allows users to generate arbitrarily complex search queries through the network. Imagine user A wants to find all network users U whose scoring $S^i(U)$ satisfies some condition ($S^i(U)$), e.g., $S^i(U) > T$. Since all the partial scores are encrypted, A does not know in advance which users

will satisfy the condition. He can surely send a Request transaction to every other user in the network, but then he will be obliged to pay every requested user, even those not satisfying the condition. Instead, user A can send a modified transaction:

$$\text{CondRequest}(A; B; E_1^i(B); E_2^i(B); : : : ;)$$

which is a conditional request | one can claim the reward only if he reveals a correctly computed score $S^i(U)$, such that it satisfies the condition ($S^i(U)$). By doing this, one can create arbitrarily complex search queries and pay only the set of users which satisfy the filtering rules".

2.6 New scoring creation

At any given time, every user can create a new scoring by publishing its manifest. A manifest is a transaction whose script contains a set of rules that specify what partial scores are allowed and how to compute the final score from the partial scores | among others it contains the code of Reveal function. Every time a user refers to a scoring S^i, it is referred by the hash of the transaction containing its manifest.

Purposely, there are no restrictions on the rules of creating a manifest | any manifest (written correctly in the IOVO scripting language) can be published. In particular, the author can freely set the rules for paying for revealing the scoring, e.g., he can introduce a fee for every scoring revealed which is paid to him. It is, however, in his interest to introduce such rules so that users are more likely to use his scoring.

3. Economics of IOVO

By design, everything in IOVO is free | there are no fees paid to miners or founders. Instead, all the financial aspects are relegated to the authors of scorings | the only fees that the user is required to pay are those introduced by the author of a scoring, and the user can't be forced to use a particular scoring. The free-of-charge transactions in IOVO are possible thanks to an innovative transaction DAG technology. It outperforms traditional blockchain technology, which is limited by its restricted block size and need for miners. The security and fluency of DAG technology is broadly discussed by the authors of IOTA [2].

3.1 Earning with IOVO

In the most probable scenario, the scorings will be created in such a way that there will be a fee for requesting a score. The user who reveals his score will be paid from this fee. Moreover, many requests will be directed to some narrow set of most valuable users in some category. Therefore users are incentivized to actively participate in the network in order to claim the rewards for revealing their score, they are incentivized to maintain the highest possible scores, to be included in as many possible narrowed search requests as they can.

4. Security

4.1 Transaction DAG

The topology of transaction DAG in IOVO is identical to the IOTA tangle, so all the security properties described in [2] hold also true for IOVO. In particular, the process directing incoming transactions can be modeled as a Poisson process with rate λ. In this model, after some aDApptation period, the total number of tips (transactions with no confirmation) will converge to oscillate around some number

$$L_0^{(k)} = \frac{k}{h}\,;$$

where h is the average time a device needs to perform the calculations required to issue a transaction and k is a parameter of the network which shows how many other transactions must be approved by the incoming transaction. The typical duration of the aDApptation period can be derived to be

$$t_0 \;\; 2{:}84 \;\; h \ln L_0$$

The authors use this model to prove that the tangle is resistant to double spending attacks, where a user tries to spend coins that were already spent in another transaction. Furthermore, they claim that using the MCMC (Markov chain Monte Carlo) tip selection algorithm makes the tangle secure against a parasite chain

attack in which an adversarial user secretly builds his own subtangle and eventually publishes it, outpacing the official version of the tangle.

4.2 Scorings

Scorings in IOVO are constructed in a way that guarantees their security and integrity. Firstly, all the partial scores are encrypted with a public key from the target user, so only he can encrypt it and no one can get any information from them. Furthermore, the final scores are computed locally, so no information from this computation is leaked to the network. Finally, the correctness of the computed score can be easily verified using the zero-knowledge proof (and, by design, no other information from this computation is leaked).

REFERENCES:

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

Nakamoto, S, *Bitcoin: A Peer-to-Peer Electronic Cash System*

Popov, S. *The Tangle*, iota.org/IOTA_Whitepaper.pdf

Ribero, Y. and Raissar, D. *Dagcoin whitepaper*, 2015