Ixian DLT

A Peer-to-Peer Presence, Names and Electronic Cash system

Ixian Team
info@ixian.io
www.ixian.io

Version: 0.9.4 pre-release
Last updated: 12. February, 2025

1. Introduction

Communication between users on the Internet has come to rely almost exclusively on centralised services serving as trusted third parties to process and relay data to correct users. While the system works well enough for most communications, it still suffers from the inherent weaknesses of the trust based model.

User problems:

- Secure communication without using trusted third parties is not practically possible.
- Service providers can ban users/countries/regions and prevent them to communicate with each-other.
- Security model is questionable, most of the times claims made by service providers cannot be verified.

Service provider problems:

- Scaling requires a lot of infrastructure, which is associated with upfront heavy investment/costs to be able to support millions or billions of users and is a big entry barrier for indie devs and smaller dev houses.

These costs and communication uncertainties can be somewhat avoided in person by communicating directly, but no mechanism exists to make communications over the Internet without a trusted party. At the top level, communication between users usually relies on going

© Ixian 2025, All rights reserved.

through top level domain name servers first to resolve service providers' servers. User’s client then connects to the service providers' servers and verifies they are connected to the right server by checking with third party certificate authorities. The servers then actually process and relay user data to the recipient. This method requires a lot of trust that the data is not manipulated and that it will reach the correct recipient. Service providers can also ban or manipulate conversations of whole regions/countries, not just individual users or groups of users.

What is needed is a peer-to-peer platform based on cryptographic proof and proof of work instead of trust, allowing any two parties to communicate with each other without the need for a trusted third party. Services which enhance interaction between users and require secure and reliable communication could easily be implemented to allow true freedom of communication. In this paper, we propose a solution to resolving user addresses and allow communication between devices/users using a timestamp server similar to what Bitcoin Whitepaper originally introduced but with improved consensus algorithm and additional capabilities which allow devices/users to communicate directly. The system is reliable as long as honest nodes collectively control more processing power than any cooperating group of attacker nodes.

2. Presences

We define each user or device as a hash of their unique public key of an asymmetric cryptographic algorithm like RSA or ECDSA, which is what we call an IXI Address. Each user, in order to communicate with another user must know their IXI address, which resolves to the user's or relay server's IP address. The problem of course is that a list of presences needs to be stored somewhere for the time period where the user or device is online. A common solution is to introduce a trusted server which stores presences and other metadata, and responds to user's presence resolution requests. Every time a user's IP address changes or they go online/offline the trusted server must be contacted to update the information. Every time a user sends any type of data to another user (be it a simple message or larger data like video or files), this data is processed and in most cases stored by the service provider's server and then relayed to the recipient once they are online (or via push notifications).
The problem with these solutions is that reliability, security and privacy of all users' communications depends solely on the practices of a single company providing the service.

We need a way for the sender to resolve the latest recipient's address without relying on a centralised server and without the possibility of tampering with the presence or other important data. To accomplish this, presences must be publicly announced, and we need a system which will temporarily store these presences for the time that the address is online and respond to users requests.

© Ixian 2025, All rights reserved.

3. Names

In order to improve user experience of requiring to remember or transmitting long cryptographic addresses, we propose a method of attaching user's IXI Address or any kind of data to a custom user-defined name. These names contain more data than presences and must be stored by the system for longer periods of time and be available to the senders, even if the recipient is offline for a longer period of time.

Traditionally Domain Name System (DNS) and accounts were used as a solution. Domain Name System stores all domain names and with the help of different registrars keeps track of who owns and can modify the name data. Every time a new name is registered, it is checked with a central server that it's not already registered. Every time a name is modified, like new data is added or modified, the user must be authenticated by the domain name service provider, usually by using an account based system. Account access and operations are local to the service being used. The problem with this solution is that reliability and integrity of all accounts, domain names and their data relies solely on the company where the name is registered and in some cases other organisations higher in the tech stack chain.

We need a way for the users to register new accounts and names, and allow only owners of the name to modify data associated with it. The system must be incentive based, to give it an economic reason, to store name data, prevent spam and prevent or make it hard for a single entity to register all possible names and cause issues for real users. For our purposes only the latest valid update is the one that is relevant to the users. The only way to achieve this is to keep a record of all registered names associated with owner/management keys and other data records. To accomplish this without a trusted party, we need a system for the participants to agree on a single state of the names. The user which requests name data must be able to verify that this is the latest valid data at the time of the request.

4. Blockchain and Transactions

The solution we propose begins with a blockchain based decentralised peer-to-peer network, similar to what was pioneered by Bitcoin Whitepaper. Blockchain based systems generally solve the issue of having all participants agree on a single state/history without a central authority. Management operations for registering and updating names can be done by using blockchain-based transactions as carriers for commands.

5. What is Proof-of-Collaborative-Work (PoCW)?

For our blockchain based solution, we've decided on a variation of Proof-of-Work (PoW), combined with some Proof-of-Stake (PoS) concepts. We call our solution Proof-of-Collaborative-Work (PoCW).

© Ixian 2025, All rights reserved.

In a classic PoW a processor intensive calculation is performed for every new block and nodes/miners compete by generating the correct solution for each new block. This results in unpredictable block times and higher potential for network forking among other things.

In PoCW a processor intensive calculation is performed periodically every few blocks and can be used for generating or signing multiple blocks. Nodes attach their signatures collaboratively to each block in order to reach the required signature threshold and PoW, and accept the block as valid and fully signed. Advantage of this approach is more consistent block times and heavily reduced chain forking potential.

6. Block Signatures

Blocks can only be signed with PoW attached to the signature. To create a valid PoW solution, hash of one of the most recent blocks must be combined with hash of the public key, on which proof of work is then performed. Proof of Work involves scanning for a value that when hashed, such as with SHA3-512, the hash begins with a certain number of zero bits, determined by the average work performed in the last blocks. After a valid solution is found, it is combined with block hash which will be signed. This combination is then signed and published to other nodes, along with the signature and corresponding public key.

Once the block is signed with a sufficient amount of signatures, it is considered as accepted by the node and the next block can be generated.

7. Signature freezing

After a block has been fully signed, nodes continue to collect signatures for such blocks until a few new blocks are generated. Each newly generated block freezes one of the recent block's (next in line) signatures by hashing all of the signatures attached to the block. This allows the network sufficient time to correctly synchronise signers on blocks.

8. Signing transaction

Every newly generated block rewards signers of a historic block. These rewards are distributed with a signing transaction, which is the first transaction of every block. All fees collected in the historic block are distributed to every signer according to their PoW solution difficulty and registered in this transaction. This incentivizes node operators to participate in transaction validation and generating new blocks.

© Ixian 2025, All rights reserved.

9. SuperBlocks

In order to improve syncing (bandwidth/cpu/storage) of end-user clients and to better support PoCW functionality for end-users, we propose a special type of block called superblock. Superblocks are generated periodically and don't contain any transactions other than the signing transaction. They are connected to each other (every superblock contains the hash of the previous superblock) and they contain a list of all block hashes since the previous superblock and other full state hashes.

10. Steps to run the network

A. New presences and transactions are broadcast to all nodes.
B. Each node rebroadcasts presences and transactions to other nodes.
C. Each node syncs with other nodes in the network and builds upon a valid block tip with most proof of work and signatures.
D. Each node periodically creates proof-of-work for one of the last blocks, packs it into a presence packet and sends it to the network.
E. Each node listens for all valid transactions received from other nodes and clients and collects them into a block.
F. Eligible nodes which are determined by deterministic random function and time generate a new block, sign it with the proof-of-work signature obtained in step D and send the block to other nodes.
G. Nodes conditionally accept the block only if the block data and all of its transactions are valid and not already spent. Nodes wait for enough signatures to be attached to the block.
H. Nodes express their acceptance of the block by adding a proof-of-work signature obtained in step D and sending the signature to other nodes.
I. When enough signatures with enough proof-of-work are attached to the block, the block is accepted by the nodes. Number of signatures which can be attached to a block is limited to keep the system scalable.
J. Eligible node generates the next block, using the hash of the accepted block as the previous hash and freezing the signatures of the block that was accepted a few blocks ago, signs it with the proof-of-work signature obtained in step D and sends the block to other nodes.

11. Block and Block Signature Validity

In order to ensure good performance and scalability of the network, we propose using the following block and signature checks.

© Ixian 2025, All rights reserved.

Blocks cannot be generated less than 30 seconds apart to prevent malicious nodes from gaming the consensus mechanism
Required signer difficulty/proof-of-work is recalculated every 2 weeks, expecting at least 40000 blocks generated in the time period. The signature difficulty threshold for every block to be fully accepted requires 23,4% of the required signer difficulty.
Required signer count is recalculated every block from the last 10 blocks, offset by 7. The number of signatures threshold for every block to be fully accepted is 75%. 50% + 1 signers must be the same as in the previously frozen block.

Proof-of-work expires and cannot be used to sign new blocks, after the block that proof-of-work was done on is older than 30 blocks.

Maximum 2000 signatures can be present on a block, before it is frozen. Maximum 1000 signatures can be frozen on a block, where signatures with higher proof-of-work have priority.

To ensure network liveness when the number of block signers/signatures can't reach at least 75% signatures, a mechanism is used which activates after two hours of no new fully accepted block and allows blocks to be accepted by increasing PoW requirement depending on the number of missing signatures.

To accept a block signature, the following must be valid:

- Signature frozen block must not have more than the maximum number of signatures. Or if it has a max number of signatures, the signature with least PoW is removed and a new signature is added, provided that the new signature has more PoW than the signature with least PoW.
- Signature frozen block signatures most not deviate too much from locally observed signatures on the block
- Block must be valid
- Block must not already be signature frozen
- PoW solution must not be older than the max solution age.
- Work must be above minimum value.

11. Incentives

Incentives to run an Ixian DLT node and participate in creating and signing blocks are created through a reward mechanism, which ensures that the new coin distribution, transaction and name registration fees are distributed fairly to the node operators.

© Ixian 2025, All rights reserved.

When an elected node generates a block, they must include a special transaction as the first transaction, which specifies the recipients and the amounts of the reward. Total reward amount is the sum of the following:

- New coin distribution which serves as initial coin distribution and minimal inflation
- all transaction fees collected in the block
- portion of name registration fees - Every time a name is registered or extended a registration/extension fee denominated in IXI goes into a reward pool, part of which is released with every block to the signers for the time period of name registration.

12. Full History Nodes

Because full history of all transactions is not required for normal node operation, we introduce the concept of full history nodes. Full history nodes, apart from the normal node functionality, also contain ranges of historic transactions. Full history nodes will be further incentivized to keep historic data and respond to requests from users by being paid per correct response.

13. Adjusting transaction and name fees

Each node operator decides on the transaction and name fees themselves. A common lower limit is set under which nodes won't accept the transaction. Suppose we have a scenario with two nodes where node A has a fee set to 1 unit and node B has a fee set to 10 units. If a transaction is sent to the network with a fee of 10 units, it will get accepted into the next block. On the other hand if a transaction's fee is set to 1 unit, it will get accepted into the block, when it's node A's turn to generate it. If a transaction fee is set to lower than 1 unit, it will be rejected by both nodes and will never be accepted into a block.

14. Spam/DDoS mitigation

To mitigate client spamming of presences and requests as much as possible, each client must perform a periodic Proof-of-Work to interact with the DLT network. For actual DLT nodes this Proof-of-Work will be of higher difficulty than other peers.

Potential spam related to transactions and names can be mitigated by adjusting fees.

Mobile and other light-weight clients where PoW is not feasible should connect through a relay/overlay network (like Ixian S2 Network, proposed in a separate WhitePaper).

© Ixian 2025, All rights reserved.

15. Mitigating nodes which aren’t following rules

Nodes which aren’t following the rules as set out in this whitepaper and reference implementation should be blacklisted on a best effort basis (i.e. IP, PoW, IXI Address) for a time period. Considering PoW is a requirement to connect, initial blacklisting by PoW would be in most cases sufficient.

16. Glossary

* **Presence**: A user' s online status or availability
* **Name**: A custom identifier registered by a user
* **Electronic Cash**: Digital currency used for transactions on the Ixian DLT
* **Node**: Ixian DLT Node which participates in verifying transactions and other data, responding to requests and signing blocks.

17. References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

© Ixian 2025, All rights reserved.