Draft @ June 19, 2018

Mixin Network

A free and lightning fast peer-to-peer transactional network for digital assets.

TECHNICAL WHITE PAPER SUBJECT TO FURTHER REVIEW AND UPDATE

Contents

Motivation

Bitcoin has started a new era for financial resources management. For the first time, people regain the power to manage their properties in their own hands, to monitor how the resources being distributed, to rescue the economy from the control of the few.

Today, both professionals and the general public have accepted the excellent idea behind Bitcoin blockchain technology, and the user base of crypto currency is growing at a faster and faster pace.

Unfortunately, Bitcoin suffers from the fast growing usage. The most significant problems are the insufficient transaction capacity, slow confirmation and high transaction fee. Due to the inflexible high distributed nature of Bitcoin network, it's impossible to fix some critical flows. Rather than perpetually fix the original Bitcoin project, people prefer to invent some new projects.

Then Ethereum, Monero, Stellar, Cardano and many new blockchains have been invented in these years. Almost all of them are trying to fix the problems existing in Bitcoin while adding some new features to their projects. However, they can't rescue the original Bitcoin network from the fast growing pain, and can't help each other neither.

Fortunately, some Bitcoin believers choose to fix the Bitcoin network and they have proposed several excellent solutions. The most significant one is Lighting Network[0], which is a micropayment system built on Bitcoin network without any modifications to Bitcoin code.

Another interesting solution is the Liquid[1] project from Blockstream, which is a federated and two-way pegged sidechain alongside Bitcoin blockchain.

All these attempts have put forward the entire Bitcoin technology innovation without sacrifice the security and distributed nature of the original Bitcoin network. Similar solutions occur quickly in Bitcoin competitors, e.g. Raiden Network[2] on Ethereum.

In this paper, we try to propose a solution to empower all the popular distributed ledgers. We name the solution Mixin. It is not about creating yet another crypto currency or a competitor to any distributed ledgers.

Similar to what Lighting Network and Liquid do to Bitcoin blockchain, Mixin is a public distributed ledger to allow any public distributed ledgers to gain *trillions of TPS, sub second final confirmation, zero transaction fee, enhanced privacy and unlimited extensibility*.

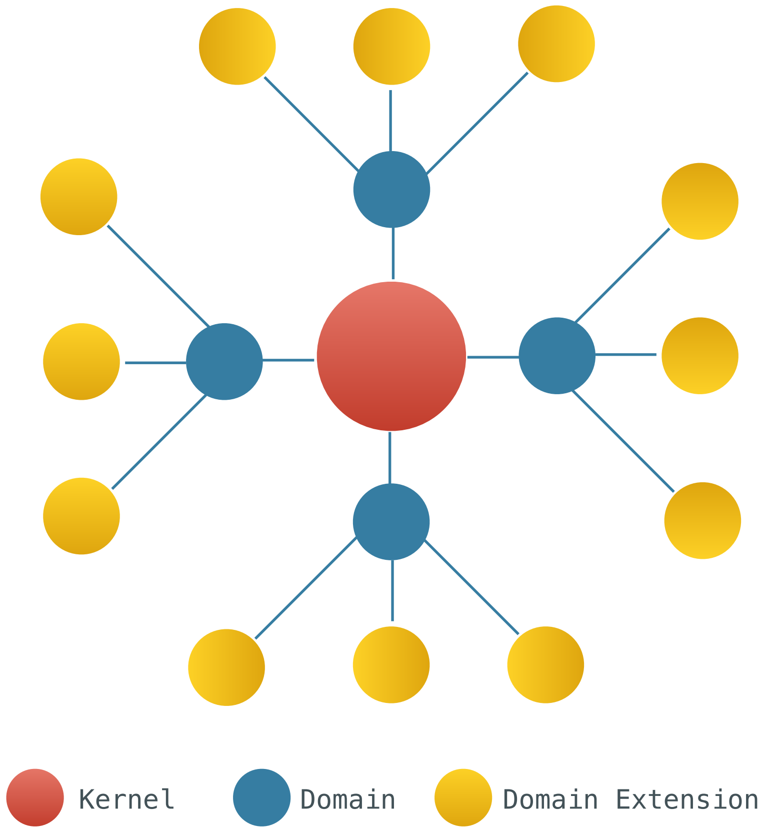
[0] Lighting Network https://lightning.network

[1] Blockstream Liquid https://blockstream.com/liquid

[2] Raiden Network https://raiden.network

Overview

Mixin is composed of a single theoretically permanent Kernel, many dynamic Domains and different multi-purpose Domain Extensions, to formulate an extended star topology in the high level.



The topology above may lead to the concern that Mixin is a central controlled network, but that's definitely not true when you look into how the Kernel works.

Mixin Kernel itself is a high performance distributed ledger and its sole responsibility is to verify asset transactions. That said, the single permanent Mixin Kernel is a distributed network just like Bitcoin network as a whole.

Although Mixin Kernel verifies asset transactions, it doesn't make out any assets. All assets flow through the Kernel by Mixin Domains.

Each Mixin Domain is also a distributed ledger, whose job is providing assets for the Mixin Kernel. The assets may be those on Bitcoin, Ethereum or any other blockchains, even central organizations like banks.

While Mixin Domain is a component to provide assets for Mixin Kernel, the Kernel itself is also a component in the Mixin Domain to verify and govern its assets.

Unlike most existing gateway based solutions, Mixin Kernel and Domains are all public available distributed ledgers, no central authorities.

From the Kernel to Domains, Mixin Network is all about assets and transactions. The Mixin Domain Extension is where magic happens, it may be Ethereum contracts, EOS contracts , a distributed exchange on somewhat trusted instances, or anything possible.

Mixin Kernel

The core of Mixin Network is Mixin Kernel, it's a fast asynchronous Byzantine fault tolerant directed acyclic graph to handle unspent transaction outputs within limited Kernel Nodes.

Ghost Output

Mixin Kernel utilizes the UTXO model of Bitcoin to handle transactions, and CryptoNote[0] one time key derivation algorithm to improve the privacy, since there is no address reuse issue any more. We call the one time key as Ghost Address and the output associated with it as Ghost Output.

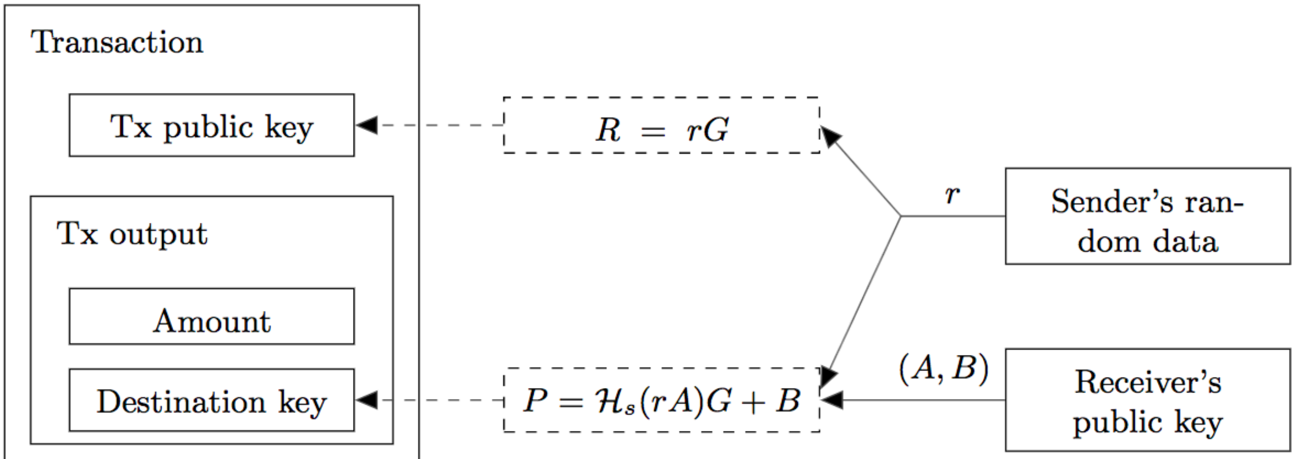
In the algorithm, each private user key is a pair (a, b) of two different elliptic curve keys, and the public user key is the pair (A, B) of two public elliptic curve keys derived from (a, b).

When Alice wants to send a payment to Bob, she gets Bob's public user key (A, B) and derives at least three Ghost Addresses with some random data, this ensures at least three different Ghost Outputs will be created for Bob.

The three Ghost Outputs threshold delivers better privacy, it also forces the outputs random amounts.

[0] CryptoNote https://cryptonote.org/whitepaper.pdf

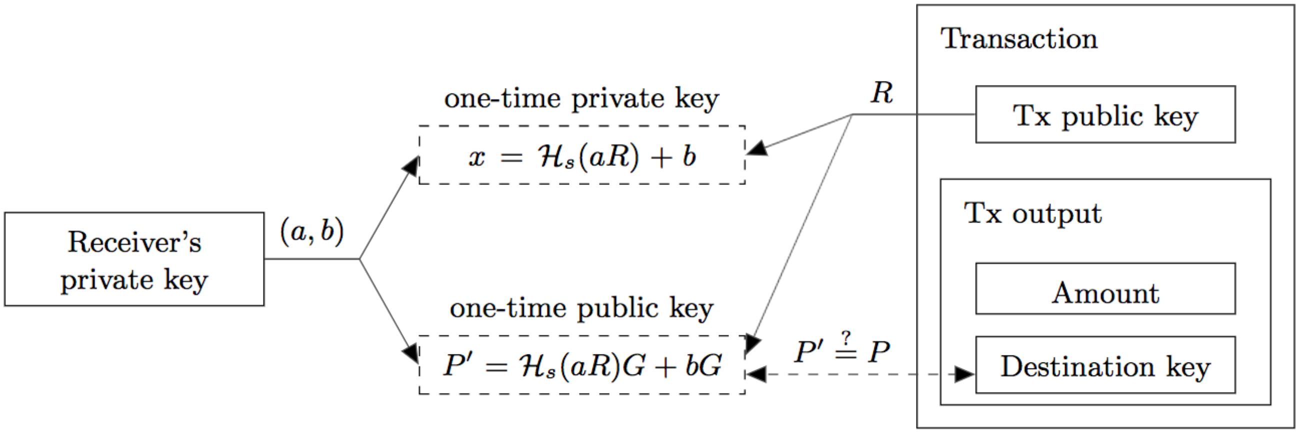
After derived the Ghost Addresses, Alice will sign the transaction with CryptoNote algorithm.



CryptoNote transaction sign

Note that, to improve privacy, Alice is forced to pick random UTXOs as the transaction inputs. After the transaction signed, Alice sends it to the Mixin Kernel.

Only Bob can recognize his transactions due to the Ghost Address feature, he can decrypt the output information with his tracing key (*a*, *B*).



CryptoNote transaction verify

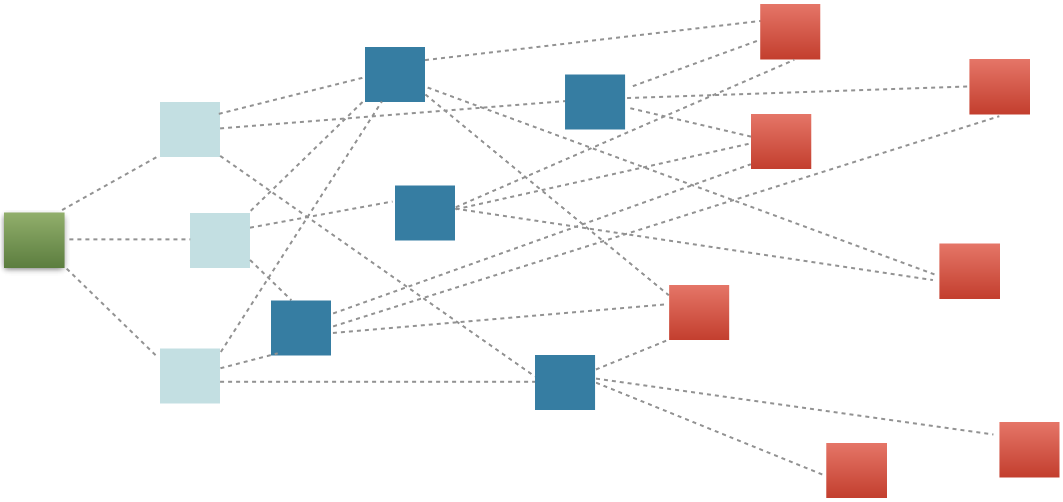
If an exchange wants to have a transparent address to disclose all its assets information publicly, it can just publish its tracing key (*a*, *B*) so that everybody can recognize all its transactions but can't spent them without the secret key *b*.

Asynchronous BFT Graph

0. Section XIN - The Token for details

Each Mixin Kernel Node is required to pledge 10,000 XIN, due to the 500,000 XIN circulating supply[0], no more than 50 Kernel Nodes will exist. To prevent extremely central authority, the Kernel can only be booted with at least 7 Kernel Nodes.

All the Kernel Nodes make up a loose mesh topology, and they are responsible for transaction validation and persistence. Unlike blockchain, there are no blocks in the Mixin Kernel, all transactions will be exponentially broadcasted as soon as possible .



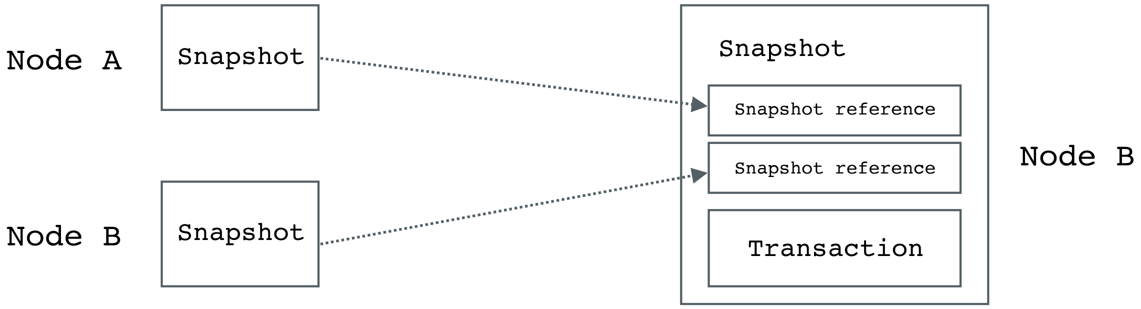
transaction flow when K = 20 and b = 3

A typical Mixin Kernel transaction finalization sequence goes as follows:

1. When Alice's signed transaction is sent to the Mixin Kernel with *K* (*7 <= K < 50*) nodes, *b* (*b > 1*) random nodes (*A*) will receive it.
2. Each node does the same transaction validation.
   1) Inputs are all unspent.
   2) Input and output amounts are in valid range.
   3) Verify the signature of each input.
   4) The total of input amounts equal to the total of outputs.
3. Each node will create a Kernel Snapshot with the validated transaction, and the snapshot is the base unit stored in the Kernel to construct a DAG. Each snapshot is composed of:
   1) The transaction as payload.
   2) Previous snapshot hash of this node.
   3) The node signature.
4. The signed snapshot will be broadcasted to another *b* random nodes (*B*) as soon as possible. After received the snapshot and validated with the same procedure in step 2, a new snapshot will be created immediately. This snapshot has the same payload as received snapshot, and the referenced snapshot hash is a pair of previous snapshot hash in this node and the received snapshot hash.
5. Steps 4 will be repeated until the node learnt that wether the transaction is approved or rejected by at least *2/3K* nodes. Since each snapshot referenced the parents up until the nodes group *A*,

it's easy for new nodes to learn that the previous snapshots are aware of the snapshots. So the procedure can avoid lots of redundant works.

6. In this procedure, a transaction can be averagely approved or rejected in about *K/b^2* rounds, considering the typical Kernel size, the latency may be within second with very high probability and guaranteed within seconds.



Due to the asynchronous BFT consensus, double spend is impossible. Because of the UTXO nature, snapshots order is irrelevant and high concurrency can be guaranteed in the DAG.

Punitive PoS

Each Mixin Kernel Node represent 10,000 XIN, which is approximate 2% of the network stake. The Kernel can only operate with at least 7 nodes joined, that's about 15% of the whole network stake.

The Kernel BFT consensus is secured by highly strict punitive PoS, that said, if a Kernel Node is recognized as an attacker, its all collateral will be recycled to the mining pool. The node will be identified as an attacker if it tried to broadcast an obvious double spend snapshot. A snapshot will be considered obvious when some of its inputs state have been validated by at least *2/3K* nodes.

The first time a node sends out attacking snapshot, its stake won't be recycled, but only be flagged by the network as potential attacker. The Kernel size will be temporally reduced to *K - 1*, and the reduction is invisible to the potential attacker.

All other nodes will still broadcast to the flagged node, but won't consider their snapshot in stake votes. If further snapshots from the flagged node remain malicious, the Kernel will sign an snapshot with a transaction that will transfer all the flagged node's collateral to the mining pool.

The flagged node will be permanently removed from the Kernel and it will have some period to appeal to the Mixin Kernel Governance[0], which is voted by all XIN holders.

Trusted Execution Environment

Mixin Kernel is already an ABFT consensus DAG. To ensure further security, Kernel Nodes must run in Trusted Execution Environment[1]. Specifically, Mixin uses Intel SGX[2] as the TEE implementation.

[0] Section Governance for details

[1] Trusted Execution Environment https://en.wikipedia.org/wiki/Trusted_execution_environment

[2] Intel SGX https://software.intel.com/en-us/sgx-sdk/details

The TEE enforcement ensures three important security and trust factors in Mixin Kernel.

1. All Kernel Nodes should run the same consensus rule.
2. Mixin Kernel will be trusted due to the Intel SGX enclave, even when the Kernel is controlled by several earlier Kernel Nodes.
3. Distributed Domain communications will be much more secure.[0]

0. Section Kernel System Calls for details

The underlying logic for the TEE security is that Intel SGX is somewhat trusted for the Mixin system.

Note that, Mixin Kernel is secure by itself, as secure as existing BFT solutions. The mandatory Intel SGX just makes it better.

Light Witness

Mixin Light Node is a simplified payment verification (SPV) node to Mixin Kernel. It typically stores all its unspent outputs for easy account balance query.

If the Light Node is a XIN holder, it has the chance to act as a Light Witness. The Light Witness will actively monitor the Mixin Kernel, and they will be scheduled to vote automatically on the attacker appeal.

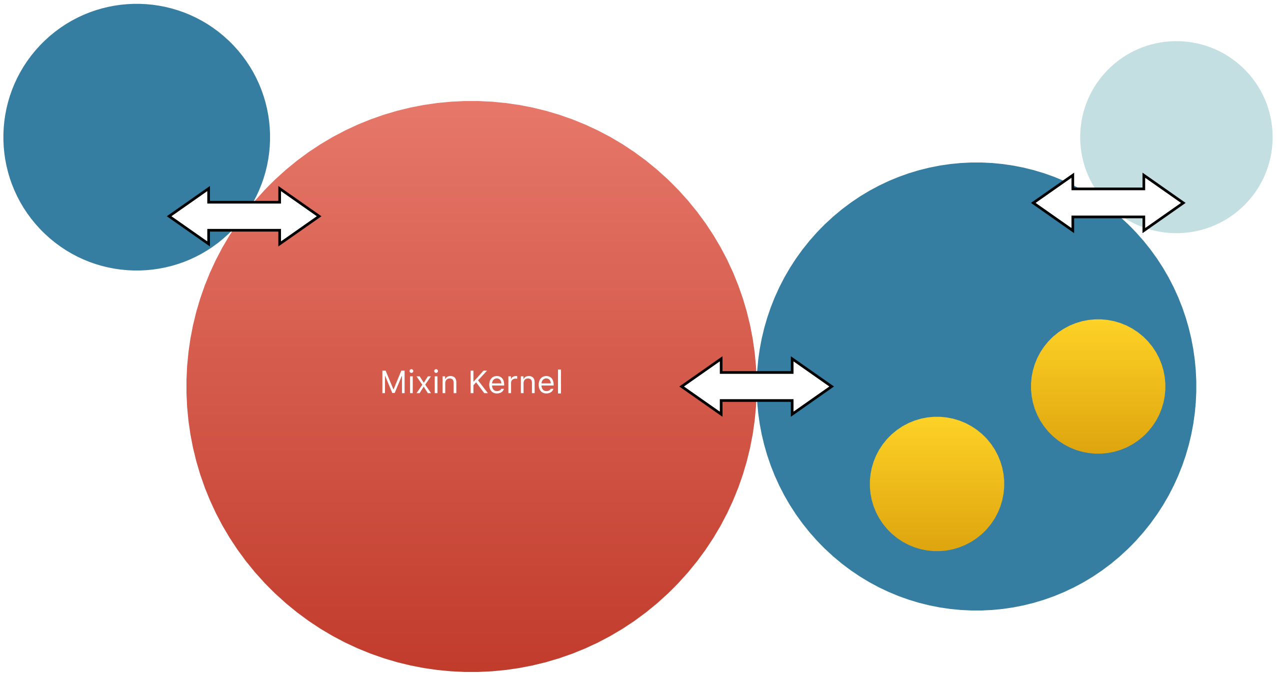
The Light Witness vote is weighted on their XIN stake. And the vote is mostly on the attacker node's network connectivity state to determine whether the attacker behavior is caused due to network delay.

All the Light Witness votes will be weight calculated with the Mixin Kernel Governance votes, to determine the final attacker appeal. If the appeal fails, the penalty will be final.

The Light Witness is incentivized to do these votes because they could get the mining reward if they do some excellent work.

Mixin Domain

Mixin Domain is a distributed ledger to provide assets for the Mixin Kernel. The assets may be those on Bitcoin, Ethereum or any other blockchains, even central organizations like banks.



- 🔴 Mixin Kernel, the ABFT DAG.
- 🔵 Mixin Domains, the distributed gateway to provide assets for Mixin Kernel.
- 🟡 Domain Extensions, could be smart contracts, trusted application, etc.
- ⚪ Trusted external sources, e.g. Bitcoin blockchain, bank API.

Kernel System Calls

Mixin Kernel offers some system calls to communicate with Domains, and it's the only way the Kernel and Domains can exchange state. The system calls are defined as standard JSON-RPC interfaces.

JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol. It is transport agnostic in that the concepts can be used within the same process, over sockets, over HTTP, or in many various message passing environments. It uses JSON (RFC 4627) as data format.

Currently Mixin Kernel only implements the standard HTTPS transport for the protocol, and the available calls are listed below.

`kernel_registerDomain`
Register the domain and waiting for the Kernel approval to connect. The call can also update the domain nodes. The registered domain will be forced to form a XIN stake based network between the domain nodes and the Kernel as a whole.

The domain registration is a governance behavior, and should relate to the domain nodes XIN stake. In the future, we hope to implement a more automatic domain management policy in Mixin Kernel. The policy upgrade should always be governed by all Kernel Nodes and XIN holders.

Parameters

1. `UUID` - A unique UUID that represents the domain among all other domains.
2. `Array` - Array of domain nodes' transparent public keys.

```
params: [“c6d0c728–2624–429b–8e0d–d9d19b6592fa”,
[“4b7a842ce6050c99450dc30b4e848c4eaffd33915653b472d900f47
d11722058”,
“b3aef7b3a998a593c157103d20f9cb17bdbd535f304b17c862e3b35b
108faeb8”]]
```

Returns

`String` - Indicate the registration request state, the value is one of `invalid`, `pending`, `denied`, and `approved`.

Note that, all Kernel System Calls should be forwarded to *b* known Kernel Nodes to ensure delivery.

Standard Domain Interfaces

A domain can only be registered to the Mixin Kernel if it implements all the Standard Domain Interfaces.

`domain_getKeyDerivationFunction`

Get the domain specific asset key derivation function, which is one of some key derivation methods in Mixin Kernel, and could be upgraded with governance.

The supported methods may also be improved to some sandboxed VM languages such as solidity.

Parameters

1. `UUID` - The global unique asset ID in the whole Mixin Network.

```
params: [“c6d0c728-2624-429b-8e0d-d9d19b6592fa”]
```

Returns

`Object` - The function name and parameters.

1. `method: String` - The function name, one of the predefined derivation function names in Kernel.
2. `params: Array` - The parameters should be used relative to the method.

`domain_associatePublicKey`

Associate a Mixin public key to the domain for an asset supported by the domain. The public key and domain asset association is the magic that will associate an external asset to the Mixin Kernel.

After public key associated with an asset, it will get an asset specific public key, e.g. Bitcoin public key.

Whenever the Bitcoin blockchain has an output to this public key, the domain will create a transaction to the Mixin public key.

This works because the Mixin Kernel and the Mixin Domain is also a Proof of Stake network. Besides the XIN collateral, there are also additional Intel SGX enforcement for all related functions.

After the domain create the asset transaction to the public key, the asset will be locked by both the Mixin Kernel and Mixin Domain. This result in the asset lightning transactions in Mixin Kernel.

Parameters

1. `String` - The Mixin public key.
2. `UUID` - Unique asset ID within the whole Mixin Network.

```
params:
[“4b7a842ce6050c99450dc30b4e848c4eaffd33915653b472d900f47
d11722058”, “c6d0c728–2624–429b–8e0d–d9d19b6592fa”]
```

Returns

`String` - The asset specific public key associated with the Mixin public key.

`domain_unlockAsset`

Unlock the asset and transfer out to external sources, this is similar to the withdrawal action in the crypto asset exchanges.

This does not mean the operation is controlled by some central authority, it just seems alike for general public.

The operation to unlock is somewhat similar to the associate function, it must be signed by both the Mixin Kernel and Mixin Domain to make it a valid snapshot acceptable by the network.

Parameters

1. `UUID` - Unique asset ID within the whole Mixin Network.
2. `String` - External asset specific public key.
3. `String` - The amount of asset to unlock.
4. `String` - The fee for external source transaction.

```
params: [“c6d0c728-2624-429b-8e0d-d9d19b6592fa”,
“15SdoFCiwaoUN4grnhPCoDWxWLcY6ZT68V”, “12.345678”,
“0.0005”]
```

Returns

`String` - The external sources transaction identifier, e.g. transaction hash.

The above three Domain Interfaces are mandatory for all domains to be approved by the Kernel. They communicate through the Intel SGX trusted transport layer, and all encrypted private keys are securely duplicated in all Kernel Nodes and Domain Nodes.

Domain Extensions

With a transaction only purpose Mixin Kernel, and Mixin Domains as assets provider and gateway to external blockchains or any other sources, Mixin has become the most sophistic and high performance distributed ledger to almost all digital assets.

However, people need smart contract, which is made popular since Ethereum. We allow Extension to Mixin

Domain, something similar to smart contract but with higher robustness, capability and performance.

Domain Extensions are some programs running in the Domain Virtual Machine secured by Enclave in Intel SGX, a popular and secure Trusted Execution Environment.

Due to the possibility to run the “smart contract” in a single computation unit, Domain Extensions can achieve many goals which are almost impossible in something similar to Ethereum.

1. Much higher performance and lower latency which is only limited by the hardware.
2. Non-deterministic transactions, e.g. trustable random number.
3. Interact directly with trusted external sources.

Besides these trusted applications, it’s also possible to run other popular distributed VM, e.g. Ethereum or EOS.

Attack Resistance

Due to the PoS and distributed nature of both Kernel and Domain Nodes, and enforced by Intel SGX, the key protection are almost guaranteed to be safe from leak.

Because of the highly distributed key duplication and secret sharing mechanism, the encrypted private keys are also guaranteed to be safe from loss.

Ideally, each asset should have many different distributed domains, these domains are governed by the Kernel and securely enforced by Intel SGX.

The associated keys can only be accessed from where it's generated in the Domain, this further improve the protection.

The Kernel will balance the assets in different Domains constantly to further prevent the asset loss due to almost impossible private key leak or loss in different domains.

We will prove that Mixin is safe for digital assets against different possible attack vectors.

To simplify the explanation, only Bitcoin will be used as a sample.

Key Association

Key association is the first step to grant a Mixin public key with Bitcoin access.

Every Mixin public key $M_{\texttt{pub}}$ will have a Bitcoin public key $B_{\texttt{pub}}$ associated, despite when would this happen, since it's irreverent to the security proof, how this association happen and be managed determines the key safety.

$B_{\texttt{pub}}$ is the public derivation of Bitcoin private key $B_{\texttt{priv}}$, so how $B_{\texttt{priv}}$ is generated defines the $B_{\texttt{pub}}$ correctness.

$B_{\texttt{priv}}$ is generated purely by the Mixin Domain itself, and it will transfer part of it to the Kernel to keep it by *(t-n)*-threshold secret sharing scheme. If the domain is trustable in this procedure, the association is absolutely secure.

Intel SGX will enforce the domain trustworthy, and even when Intel SGX itself is not safe, which is almost impossible, the following parts in this paper will also proves that the Bitcoin asset will also be secure in Mixin.

Deposit Attack

Deposit is the action when external assets flow into Mixin Kernel, this is the first step when some BTC joins Mixin.

Since key association is proved secure, and all Mixin Domains are governed by Mixin Kernel. So if some BTC is submitted to the Kernel, it will be guaranteed to the correct $M_{\mathtt{pub}}$.

All Bitcoin deposits will also require a large enough domain finality threshold, that said, there must be at least 12 Bitcoin confirmations when the system will accept the asset.

Then the system have enough time to detect fraud domain action and will punish it without any Bitcoin loss.

The domain mandatory Intel SGX requirements will improve this further.

Fraud Domain or Key Leak

Mixin Kernel constantly balance the assets across all Domains according to their behavior and collateral amount. If a domain is fraud or hacked, the leaked key will only cause partial of the Bitcoin loss.

Also the Intel SGX will prevent fraud Domains from existing and keep hackers away in most cases.

Further more, Kernel and Domains will always load most Bitcoin into a multi signature $B_{\mathtt{mpub}}$, this is almost impossible get hacked, especially correctly and transparently distributed implemented.

Damaged Domain or Key Loss

Just like the fraud domain issue, domain damage or key loss will only affect a few Bitcoin assets.

Since the Mixin Governance will ensure the Domain is correctly implemented as a distributed system, it's almost impossible to have the domain damaged as a whole.

Compare to Exchange

Exchanges or other kinds of central managed Bitcoin solutions, they typically store most BTC in their cold storage.

The cold storage is actually some private keys which are never exposed to the Internet and managed by several people in the same firm.

In terms of security, if both Mixin and Exchanges implement the solutions correctly without any bugs, Mixin is considered much more safer and trustable.

Because Mixin multi-signature $B_{\texttt{mpub}}$ is guaranteed to be managed by many different parties, while exchanges have their keys kept by their own people.

Despite hackers, exchanges may have the chance to steal the money by themselves. This is much more harder or even impossible for Mixin.

The most important thing is that most exchanges can’t implement the solution correctly, which will expose bugs to hackers.

While Mixin solution is transparent, the code is open to all people to review and improve, this is how Linux is thought more secure than Windows.

No system is perfect, it does have issues, just like Bitcoin have been actually controlled and attackable by central mining pools. But through above factors, we have confidence that Mixin is secure enough, and much more convenient than any existing technologies.

Governance

We try our best to make Mixin Network just work without any governance, but there are still situations the program can’t handle.

XIN is the only stake to determine how the governance work on all the Mixin problems. The vectors that can be voted to governance are listed, and even this list are in governance.

1. Kernel Node penalty, mainly when double spend, or fraud asset.
2. Asset and Domain registration, determine which asset to be added to the Mixin Kernel. This may be programmed automatically in the future.
3. External asset assurance, e.g. how to recover when Bitcoin forks after the domain finality threshold.
4. Kernel development and upgrade. Determine some policy in the Mixin Kernel specification and upgrade procedure.
5. Community development, vote on community issues if critical.

XIN - The Token

XIN is the sole token used by many services in Mixin, especially full node collateral, the DApp creation and API calls.

To join the network as a full node, it should pledge at least 10,000 XIN token to establish the initial trust.

Every DApp creation will cost some XIN for one time, the cost is determined by the resources the DApp claimed to consume. The Mixin API calls from DApp may cost some XIN depends on the call type and count.

All the XIN penalties and fees charged by the network will be recycled to the mining pool.

1,000,000 permanent total XIN token is issued to the world at one time, and 400,000 of them have been successfully distributed to holders from 25/11/2017 to 25/12/2017 with rate 20 EOS/XIN.

50,000 XIN have been distributed to early Mixin Messenger adopters. 50,000 XIN are reserved for the development team.

The remaining 500,000 XIN will be the incentives for all Mixin full nodes and light nodes.

Conclusion

We have proposed the Mixin Network as a multi-layer distributed network, the core layer Mixin Kernel may be the best distributed transactional network due to the simplicity in the ABFT directed acyclic graph design. The Mixin Domains layer is quite extensible without any overhead to the Mixin Kernel performance.

We also have thorough security proof that when managing external blockchain assets, Mixin is secure for daily usage compared to almost any existing cold storage solutions.

The most important thing is that Mixin isn’t inventing any new things, and all technologies described in this paper have been mature parts in existing projects.

The Mixin Messenger app have proved that this paper is feasible to be implemented in real world, unlike most other projects that has beautiful new theory but impossible to make to daily usage.