Hcash

TECHNICAL YELLOW PAPER

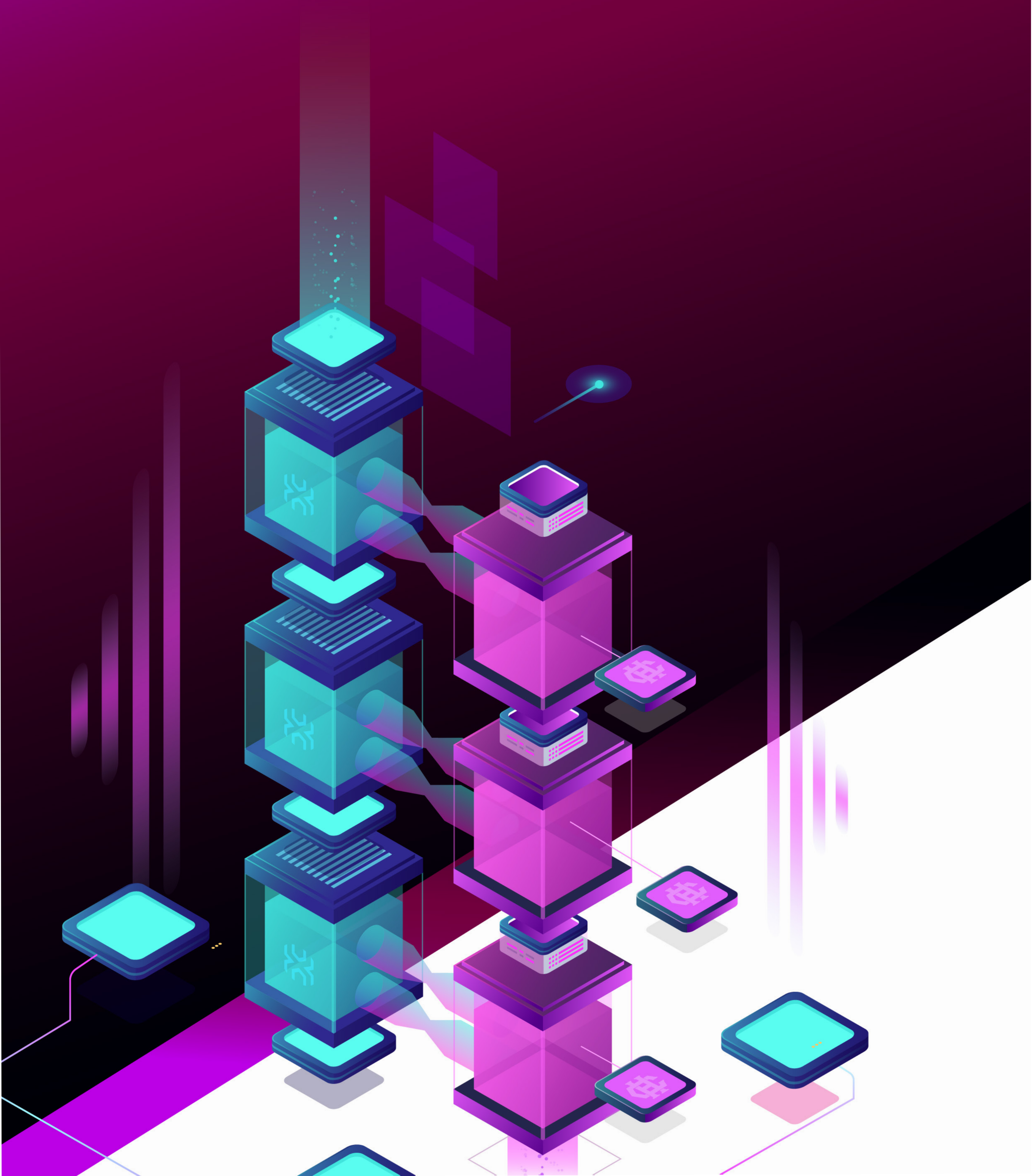


Table of Contents

1. Background

The UTXO-based blockchain system (e.g. Bitcoin [1]) and account-based blockchain system (e.g. Ethereum [2]) has opened the door of a brand-new world for us. Although the impressive success of Bitcoin and Ethereum has certainly proved the value of blockchain technology and its massive potential in the future, we see some inherent problems that need to be addressed.

The current status is that multiple blockchain co-existing in the blockchain world, however, they all face a similar problem that the information cannot freely circulate among the chains, thus creating "isolated blockchain islands". This has been a big inhabitor for the development of blockchain ecosystem.

Security technology based on hash algorithms and ECDSA elliptic curve cryptography provides some protection for Blockchain networks. The upcoming creation and implementation of quantum computers threatens almost all the cryptography which acts as the foundation of all current internet security systems. Thus, Quantum computing poses a huge threat to blockchain based cryptocurrency.

From its inception, HCASH 's vision has been to adopt and combine both blockchain technology, DAG side chain technology and advanced quantum-resistant signature scheme technology in order to create a safer and more efficient decentralised distributed ledger, and to achieve interoperability between Blockchains to create a fully trustless environment. In addition to these, elements such as a hybrid PoW+PoS consensus mechanism, smart contract functionality, privacy preserving mechanisms, and DAO will be added in increments to the network.

After a year of development, our team has made some breakthroughs, and this document will showcase them. This document should be considered as our Yellow paper version 1.0, and will contain information regarding our technical achievements and updates to our development roadmap.

As our technology and innovations evolve, we will continue to update this document and keep pace. For the latest information on HCASH and the progress of its technological development, please visit our official website at https://h.cash.

2. The Evolution of HCASH

According to our vision, HCASH will create a new platform which is able to be 'connected' to different blockchains (such as that of Bitcoin and Ethereum), thereby allowing value and information to circulate freely between systems, redefining the value of a blockchain.

In this yellow paper, we have focused on our main features, such as cross-chain technology, post-quantum cryptography, PoW+PoS consensus mechanism and smart contracts. To maintain our initial vision without affecting technological development, we propose the next generation of HCASH to have two chains running laterally, each serving different functions in the ecosystem. We call this ***"interlinked, bifocal" dual-token and dual-chain ecosystem***. In future, the development and advancement of technology will be achieved in the framework of two parallel chains, with a symbiotic relationship between them.

2.1 Parallel, dual-chain ecosystem

The proposed parallel, dual-chain ecosystem consists of an upgraded HCASH main chain (HyperCash) and a new chain - HyperExchange, which is derived from the main chain, with a different set of functions.

As the two core hubs in the HCASH ecosystem, the two chains can freely share their various technical advantages and resources from each other's ecosystem. HyperCash provides value tokens, helping maintain HyperExchange's system stability, while HyperExchange supplying HyperCash with more application scenarios. Hence, HyperCash represents the underlying blockchain technology for the HCASH ecosystem while HyperExchange solves inter-blockchain communication and brings more ecological advantages to the whole system. They are making joint efforts in bringing more resource and expanding the HCASH ecosystem.

According to the original plan, after the Hshare chain upgrades to version 2.0 (and is renamed to HyperCash), characteristics such as post-quantum cryptographic signatures and the hybrid PoW+PoS consensus mechanism will be achieved. Going forward, HyperCash will focus on the in-depth development of the underlying technology of blockchain. HyperExchange is the first system to achieve interoperability using the Blockchain Multi Tunnel (BMT) protocol, HyperExchange Axis, Indicator, smart contracts and other blockchain innovations. HyperExchange lays the foundation for building cross-chain distributed commercial applications in its ecosystem.

Together, HyperExchange and HyperCash build an interlinked, bifocal, dual-token, dual-chain ecosystem, with the vision proposed by HCASH basically completed.

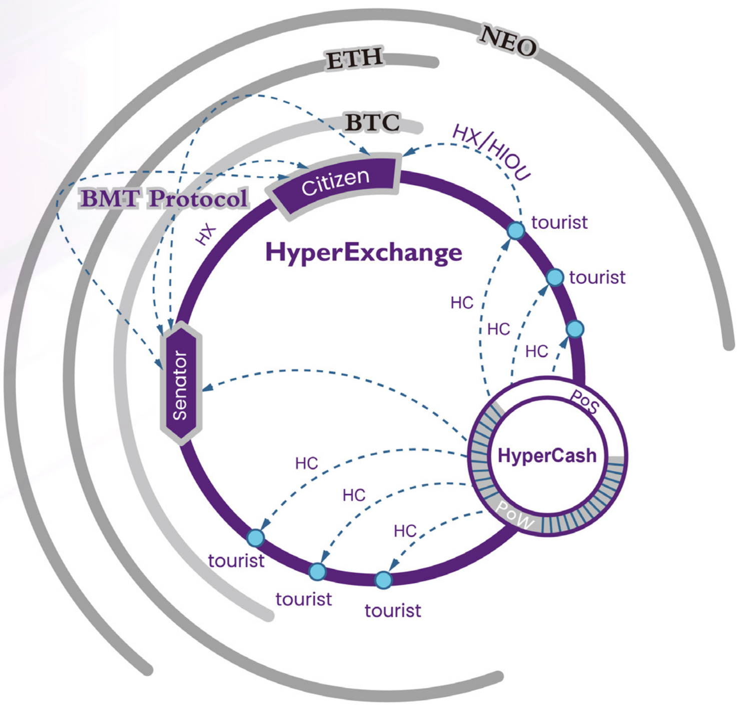


Figure 1: The HCASH parallel, dual-chain ecosystem diagram

2.1.1 HyperCash

Determining the right approach to building a chain that meets HCASH's project goals meant looking at two options:

1) Building a completely new chain from the ground up
OR
2) Forking from an existing open-source codebase

The primary considerations when choosing between the two, were whether the approach would ensure the implementation of functions proposed in the original vision, and whether the approach would ensure the delivery of these features in a timely manner.

HCASH found, that Decred (DCR), the public blockchain, shares many of the core values outlined in the principles of HCASH . Decred itself is a fork of Bitcoin, thus, if HCASH builds upon the Decred network, any innovations and updates implemented thereafter would be compatible, or require minimal modifications to be implemented in other main chain systems. Ultimately, this ensures a more open and cooperative environment.

Developing on top of the open-source DCR code, the HyperCash main chain has the following advantages:

1) Decentralised autonomous governance

It is well known that community governance in blockchain systems has always been a difficult task. For mechanisms similar to Bitcoin, the excessive control which miners have, restricts the development of the whole community, slows decision-making, creates a system where forking has become prominent. Additionally, mechanisms similar to ZCASH can become too centralised, which reduces participation from the community.

HyperCash will adopt the community governance mechanism proposed by DCR - the autonomous governance model.

DCR's technology supports the world's first successful direct-chain user activated vote, an innovative voting model that strengthens stakeholder autonomy and allows a seamless transition from one set of rules to another. Decentralised decision making and self-financing through this model have enabled DCR to build its strong and evolving community, while remaining relatively unaffected by third parties.

We believe that this community-driven autonomous governance model can adequately fulfil the needs of all HCASH stakeholders, ensuring the efficiency of decision-making, and ensuring more sustainable and inclusive development. This will achieve the 'autonomous' goal HCASH envisioned.

2) PoW+PoS hybrid mechanism

In the area of public blockchains, a prominent issue is the ongoing conflict between decentralisation and performance.

Blockchains which adopt a solely PoW consensus mechanism have done well from the aspect of decentralisation of consensus, however, the inability to scale to a large user base prevents it from being viable in commercial applications.

On the other hand, blockchains adopting a purely PoS consensus mechanism have a reputation of being too centralised, and are concentrated too heavily towards a small number of individuals who can afford a large amount of tokens.

The combined model of PoW+PoS hybrid consensus can merge the strengths and alleviate some of the weaknesses of both. Based on the benefits using both PoW and PoS, HCASH chose to adopt this mechanism.

3) Decentralised mining pools

If an individual user conducts PoS mining alone, they must constantly run their wallet on their system and be connected to the internet. If the wallet is not running, the user may miss the period when the network tries to verify their ticket (when the ticket is added to the ticket pool and when the ticket is selected as the ballot), therefore, the opportunity to receive a staking reward is missed.

Therefore, in the case where a user cannot or does not wish to run their wallet constantly, they can choose to join a stake pool.

Pools for most blockchains are centralised, and therefore possess security risks, and show a lack of transparency.

One of the biggest advantages of a decentralised mining pool is that the user is not required to transfer token ownership to the pool itself. The staking pool is built into the underlying protocol, the fees are pre-determined, and the pool has no access to the user's assets or mining rewards.

The upgraded HyperCash main chain will have decentralised mining pool functionality, in order to increase the security of assets and be more transparent about rewards.

4) The BLAKE256 algorithm

The BLAKE256 algorithm has been proven to be secure, with no effective attacks against it since its release in 2012.

Most blockchain projects, including Bitcoin use some version of the SHA2 algorithm family, which includes SHA256, SHA384, SHA512 and others. SHA2 can be regarded as SHA1 with increased security, however, there may still be vulnerabilities. Compared to BLAKE256, SHA2 is also inferior in terms of speed.

In terms of technical depth, BLAKE256 is comparable to SHA3, which possess the same security level, however, SHA3 is inferior when it comes to speed.

The HyperCash main chain therefore adopts the BLAKE256 algorithm to ensure the safety and efficiency of the system.

Based on the HCASH team's careful evaluation of the above situation, we have determined that forking a chain from DCR's open-source code retains several desired functions set out in our original vision, enabling us to confidently progress with project development.

HCASH believes that instead of building a new chain from the ground up, choosing to use an open-source codebase and developing on top of that is an advantageous method to ensure the faster development of other quality features in its chain. Additional to this, it would also create a more efficient and secure solution, and will foster closer relations between other Blockchain parties, as new implementations developed in HCASH can be more easily transferred to other networks.

HyperCash has more unique features through technology development compared to Decred:

Firstly, we'll begin with security. As security is a main concern on its network, HCASH selected quantum-resistant security technology as one of its key functionalities that will be active at launch. Although only a few Blockchain projects have acknowledged the threats that quantum computers pose, the emergence of quantum attacks in the long run is an inevitability.

After nearly a year of technological development, we have created a practical post-quantum signature scheme, based on BLISS. Details regarding this will be explained below.

Secondly, beyond security, HCASH is also focusing on developing and implementing privacy preservation techniques. In order to achieve better privacy protection, HCASH aims to do the following:

1) To develop an iteration of ZCASH's zero-knowledge proof protocol.

2) To effectively use the research provided through the Monash University joint Blockchain technology laboratory as the basis for the implementation of a new quantum-resistant ring signature scheme. Dr. Joseph Liu of the Joint Lab published a research paper in the field of 'ring signatures': as a privacy layer, circular signatures can be grouped by transactions, but only one person in the group is randomly selected for verification by signature, increasing the additional protection against the anonymity of transfers.

Further developments on post-quantum ring signatures are still in progress. Specific technical implementation details regarding these signatures and zero-knowledge proofs will be detailed below.

Acting as one of the two core hubs in the HCASH ecosystem, the HyperCash chain focuses on the in-depth development of the underlying base chain, as well as upgrading specific features in corresponding technologies, ASIC (application specific integrated circuit) Resistance and the HCASH AI Lighting Protocol (HAILP) has already been proposed in the roadmap for next phase development. The cross-chain functionality set out in the HCASH design vision will be achieved by the other core hub: HyperExchange.

2.1.2 HyperExchange

HyperExchange, acting as one of the two core hubs of the HCASH parallel symbiotic-chain ecosystem, takes the lead in implementing inter-blockchain information interconnection by implementing a Blockchain Multi Tunnel (BMT) protocol, a HyperExchange Axis, Indicators, Smart Contracts and more. HyperExchange will not only achieve cross-chain interconnection, but will enable Turing-complete smart contracts, laying the foundation for complex distributed business applications.

HyperExchange also has the following features:

1) RPPOM and decentralised mining pools

HyperExchange uses a ***RPPOM (Random Pool Proof of Multi-properties)*** decentralised consensus algorithm, which is a stochastic multi-asset equity pledging consensus algorithm, which will enable decentralised mining pools to operate.

There are four roles that exist within the HyperExchange ecosystem: Tourist, Candidate Citizen, Citizen, and Senator.

Tourist: Includes all users on the HyperExchange chain and all users on other chains that interoperate with the HyperExchange network.

Candidate Citizen: Ordinary users can pay 10,000 HX to become a Candidate Citizen (this fee will be burned). Candidate Citizens can accept the pledge of other users on the chain.

Citizen: A Citizen is a block generator and community governor of the HyperExchange chain, responsible for validating transactions and block generation. Citizens operate as decentralised mining pools on the chain. Candidate Citizens can become Citizens by providing multiple assets to stake on the chain (HSR/HC will receive a higher weighting), assets of which can be provided by themselves or supported by other users. If a Candidate Citizen do not have sufficient pledging assets, it is nearly impossible for him/her to become a Citizen.

Senator: A Senator is an asset manager and community administrator in the HyperExchange ecosystem. By consensus, Senator is responsible for managing assets on hot and cold multi-signature addressed on difference chains. And the cross-chain interconnection ultimately must be conducted by Senator's consensus. To become a Senator, a user must pay 500,000 HSR/HC, ensuring a significant financial stake is placed in the system, which will work to safeguard the networks integrity and stability by aligning the Senators interests with that of the network. The change of Senator is determined by Citizen votes.

By pledging, Citizens are selected to produce blocks and obtain block rewards. Tourists can also participate in mining and obtain mining rewards by pledging assets with Citizens. According to the RPPOM consensus mechanism, after new blocks being generated, the system will automatically follow the underlying protocol and distribute rewards to citizens and their supporters (users who pledged their assets with Citizens), according to their weighting.

This kind of pledging mechanism has the following advantages:

a. Collateral mining and rewards sharing are jointly verified by the entire network. After block rewards being generated, the internal underlying protocol will distribute those block rewards to Citizens according to their weights. There will be no uncertainties unlike those seen in centralised mining pools.

b. Similar PoS mining pool users are required to transfer their assets to their relevant pool, whereas the HyperExchange chain only stake assets, meaning, the users private key is still in their own custody, there is no need to physically transfer it to a Citizen, thus there will be no security problems and users can withdraw their staked assets at any time.

2) 100% Reserve Ratio

In order to ensure the stability and safety of the HCASH ecosystem, the HyperExchange reserve ratio will be kept at 100%. Each HIOU (HIOU: Refers to other digital assets, anchored on the HyperExchange. For example:HXBTC refers to the anchored Bitcoins on HyperExchange) has a real original chain asset (such as BTC, ETH, etc) that will be placed into hot or cold multi-signature addresses on the original chain. Those addresses will be managed by the RPPOM consensus mechanism. This ensures that all assets on the HyperExchange network will not be added or destroyed without reason, and that increases or decreases in assets correspond to users' deposits or withdrawals of assets on the original chain.

3) Efficiency

In accordance with the RPPOM consensus mechanism, the HyperExchange network produces a block every 5 seconds, producing significantly faster transaction confirmation speeds. This can be seen when compared to BTC which produces a block every 10 minutes, or LTC which produces a block every 2.5 minutes. The performance of a BTC or LTC asset transfer or trading on the HyperExchange network is approximately 120 times that of the BTC network and 30 times that of LTC.
HyperExchange’s theoretical TPS (transactions per second) reaches 10,000, which is enough to carry high-load transactions on multiple chains.

The specific comparison can be seen in the figure below:

Table 1: Mainstream Public Blockchain Performance Comparison Chart

Blockchain	**Block time**	**Size of block**	**Theoretical TPS**
HyperExchange	5s	20M	10000
BTC	10min	1M	7
ETH	17s	No upper limit (800 million gas)	22
EOS	1.5s	No upper limit	Millions
NEO	20s	No upper limit	1000
LTC	2.5min	1M	28

From the table above, we can see that the HyperExchange network has significantly improved performance over BTC, LTC, ETH, NEO and other chains when comparing against production rate, block size, and theoretical TPS. Furthermore, its block time is 5 seconds, and its theoretical TPS is 10,000, which is enough to meet the needs of high-frequency and high-capacity services.

Although there is a gap when comparing the theoretical million-level TPS of EOS to the HyperExchange, it should be pointed out that EOS block producers require a very stable network connection between servers, meaning they have high-performance server requirements. The block producing mechanism of EOS is that of 21 block producers, each of whom produce blocks in order. It hence has the risk of being exposed to DNS fraud and DDOS attacks.

In contrast, HyperExchange's server and network requirements are less demanding, more adaptable, and can easily interface with other blockchains. Also, the HyperExchange generation node is randomly selected from all Candidate Citizen who meet pledging requirements. This means that the HyperExchange's generation node is randomised and difficult to target. Therefore, the risk of network attacks can be circumvented to the greatest degree and partial nodes can be attacked completely without affecting the stability of the entire network.

See the HyperExchange White Paper for details.

2.2 Dual-token

HyperCash and HyperExchange are the two core hubs of the HCASH parallel symbiotic-chain ecosystem. The tokens produced by HyperCash and HyperExchange are HC and HX, respectively.

2.2.1 HC

HC is the asset that will be distributed through the HyperCash chain and a 1:1 conversion ratio from HSR will take place (refer to the HCASH Foundation's future official announcement regarding conversion details). HC is the value token that maintains the ecological stability of the HCASH network. HC can be used as an asset on the HyperCash chain to participate in the governance of the HyperCash chain, and can also be used as a pledged asset to participate in the production of HX tokens on the HyperExchange chain.

2.2.2 HX

HX is a functional token on the HyperExchange chain, its function will primarily be to maintain and invoke related functions within the HCASH ecosystem.

Specifics regarding the generation of HX tokens can be found in the HyperExchange White Paper.

3. HCASH Development Progress

3.1 Cross-chain

HCASH's core vision is to create a platform which links a variety of blockchain technologies, allowing trust-based value to flow freely across different blockchain systems. In our design, value circulation is not limited to blockchains, it also includes distributed ledgers based on blockchains and those which are based on other technologies, such as DAGs.

In the past years, our development team has made tremendous efforts to achieve the value interconnection between various blockchains through HyperExchange, one of the components of the HCASH dual-chain ecosystem. The team will continue to work on R&D as per the technology development roadmap. With existing cross-chain technology, we are striving to achieve the transfer of value between the various types of blockchain and non-blockchain based distributed ledger systems.

In this Yellow Paper, 'cross-chain' refers to the transaction of information, and therefore value between blockchains.

3.1.1 Cross-chain Technology Implementation

HyperExchange is the first system to achieve interoperability using the Blockchain Multi Tunnel (BMT) protocol, HyperExchange Axis, Indicator, smart contracts and other blockchain innovations. HyperExchange lays the foundation for building cross-chain distributed commercial applications in its ecosystem.

3.1.1.1 Blockchain Multi Tunnel Protocol

The Blockchain Multi-Tunnel Protocol is HyperExchange's communication protocol for point-to-point information transmission between blockchains.

When a user creates an account in HyperExchange, a corresponding tunnel account is generated according to the BMT protocol, and links it to the HyperExchange account. When a cross-chain transaction occurs, the corresponding data is encapsulated, packaged, and securely transferred through the BMT protocol.

3.1.1.2 HyperExchange Axis

The HyperExchange Axis is an important component of cross-chain communication. As a multi-asset decentralised ledger, it is responsible for recording, verifying and broadcasting cross-linked data, generating and destroying corresponding HIOU tokens, and completing the transfer of cross-chain assets. In HyperExchange's Axis, network security is ensured by consensus through 'Senators'. The binding relationship between the HyperExchange account and the corresponding tunnel account is verified by the Senator on the chain, and the cross-chain transaction of each asset is also verified by a Senator. Consensus is reached to verify and ensure that each asset chain is consistent with the status of the HyperExchange Axis.

3.1.1.3 Indicator

The Indicator is HyperExchange's client software. The Indicator is responsible for generating data in the HyperExchange ecosystem. Users use the Indicator to register HyperExchange accounts, become a part of the ecosystem, and take part in the development of the ecosystem by mining or validating cross chain transactions to obtain rewards from the ecosystem.

3.1.1.4 Smart Contracts

Trading activities such as multi-asset purchase orders on the HyperExchange are implemented via smart contracts, including:

Pending purchase order functionality rules:

- When creating a pending purchase order by implementing a smart contract, a pending purchase order is a new smart contract.
- Pending orders are able to be partially filled.
- Pending purchase orders are able to be cancelled.
- The matching deal and the withdrawal of the pending order are both transactions called by smart contracts.
- Buying pending orders, matching pending orders, and cancelling purchases all incur fees.

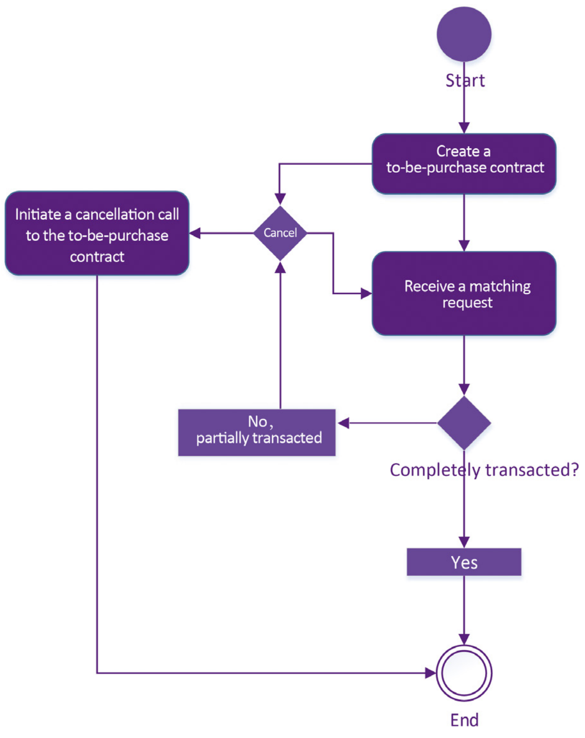

Start
Create a to-be-purchase contract
Initiate a cancellation call to the to-be-purchase contract
Cancel
Receive a matching request
No, partially transacted
Completely transacted?
Yes
End

Figure 2: Pending purchase orders

3.1.2 Cross-chain Process Flowchart

3.1.2.1 Initialisation Process

1) Open HyperExchange, perform initial asset allocation and basic parameter configuration.

2) Through consensus on HyperExchange, a Senator creates multi-signature accounts for BTC, LTC, and other blockchains. The multi-signature addresses are signed by all Senators and then broadcast to the HyperExchange.

3) Each Senator pays a margin of responsibilities specified by the HCASH Foundation for the maintenance of chain stability.

3.1.2.2 Account Creation

To complete a cross-chain transfer, a Tourist needs to create a HyperExchange account on the Indicator. The Indicator will provide a tunnel account option, and bind with the HyperExchange account.

In this way, when an asset (such as BTC) is recharged to a multi-signature account, HyperExchange will have the same amount of HIOU to the corresponding HyperExchange account after block confirmation. When the entire HyperExchange is initialised or a new Senator joins the HyperExchange network, the multi-signature account (in BTC, LTC or others) needs to be created or updated in the asset chain.

Account type includes:

1) HyperExchange Account

To begin, users are required to create a HyperExchange account. This account will allow the users to store and trade multiple assets on the HyperExchange, including HIOU, HX and other assets.

2) Tunnel Account

When a user creates an account in HyperExchange, HyperExchange generates a corresponding tunnel account using the tunnel protocol, and binds it to the HyperExchange account. There is a free daily limit of 10,000 transactions for the HyperExchange account bound to the tunnel account. Any transactions after the limit is exceeded will require a Citizen’s verification & approval. (This 10,000 limit can be changed through a consensual agreement).

3) Hot and cold multi-signature account

Cross-chain assets will be stored in hot and cold multi-signature wallets on various asset chains. Through consensus, these hot and cold multi-signature wallets are created and managed by Senators.

3.1.2.3 Cross-chain recharge process

In addition to the HyperExchange account, users will also have a number of associated tunnel accounts. Through HyperExchange's bundled light wallet components of other asset chains, users can recharge from other asset chain addresses, or from centralised exchanges, to the tunnel account address associated with the HyperExchange account.

If a Citizen on the HyperExchange detects a transfer to the associated multi-signed recharge address, they wait for the blocks to reach a certain amount of confirmations. After this, the associated HyperExchange account will be indexed through the "from" address of the recharge, and credited corresponding amount of HIOU (HXBTC/HXLTC, etc) through consensus on the HyperExchange. Then the HIOU (for example, HXBTC/HXLTC, etc.) is frozen assets (block m) by default, and after the number of HyperExchange blocks reaches a certain height (for example, m+17), the 'frozen' status of HIOU assets, which are generated by consensus, becomes 'available' to be transacted.

Recharge on the HyperExchange is divided into the following sections:

1) The user recharges to the tunnel account associated with the HyperExchange account. This transaction confirmation depends on the confirmation time on the original asset chain. For example, BTC would require 6 blocks of confirmation, and LTC would require 12 blocks of confirmation.

2) Transfer of funds from the tunnel account to the corresponding multiple signature address is managed by Senator consensus. This transaction confirmation also relies on the confirmation time on the original asset chain. If the asset chain is BTC, it takes 6 blocks, for LTC, it takes 12 blocks.

3) On the HyperExchange, a Citizen generates corresponding HIOU, which is distributed to the user through consensus - this requires 17 confirmed blocks.

The underlying operation flow chart for Cross-chain recharge is as follows:

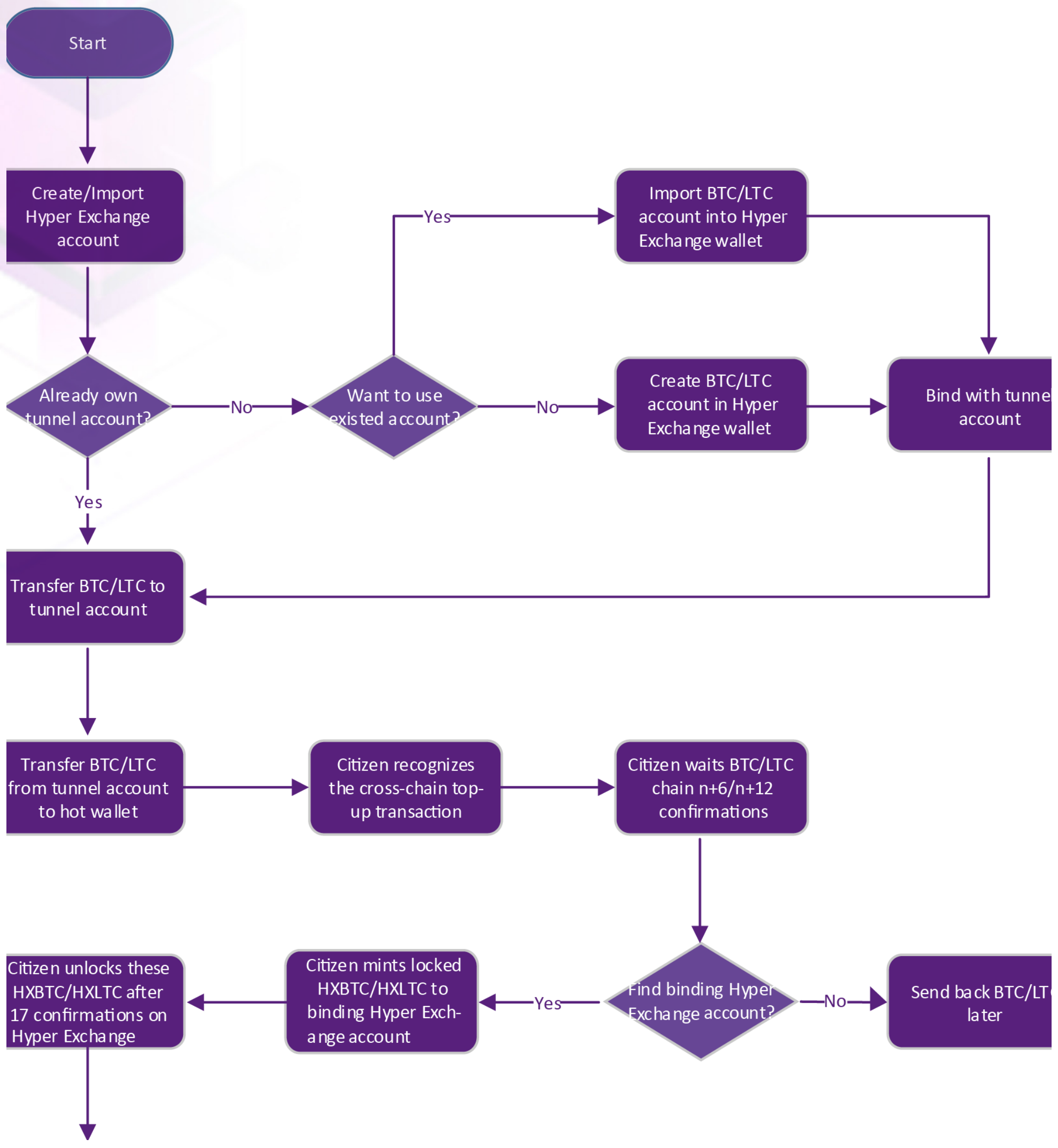

Start
Create/Import Hyper Exchange account
Already own tunnel account?
No
Want to use existed account?
Yes
Import BTC/LTC account into Hyper Exchange wallet
No
Create BTC/LTC account in Hyper Exchange wallet
Bind with tunne account
Yes
Transfer BTC/LTC to tunnel account
Transfer BTC/LTC from tunnel account to hot wallet
Citizen recognizes the cross-chain top-up transaction
Citizen waits BTC/LTC chain n+6/n+12 confirmations
Find binding Hyper Exchange account?
No
Send back BTC/LT later
Yes
Citizen mints locked HXBTC/HXLTC to binding Hyper Exch-ange account
Citizen unlocks these HXBTC/HXLTC after 17 confirmations on Hyper Exchange

Figure 3: Cross-chain recharge flowchart

During actual operation, Indicator will simplify the human to computer interaction by packaging the majority of processes. The user only needs to follow simple steps to complete cross-chain recharging.

3.1.2.4 Cross-link asset withdrawal process

The user initiates a withdrawal transaction, which includes the withdrawal addresses of other asset chains.

When the Senator receives the withdrawal transaction, the Senator will sign on this transaction and broadcast this signed transaction to the network. When it comes to a Citizen's turn to produce a block, the Citizen collected Senators' signatures and judges whether at least two thirds of Senators have signed the transaction. If the conditions are met, the transaction and all the collected signatures are packaged into a block. Otherwise, the transaction will not be packaged, but handled by next citizens.

After a transaction is packaged into a block, the corresponding HIOU (such as HXBTC, HXLTC, etc.), owned by the user is destroyed. After verifying, the Senator first determines whether there is enough balance in the multi-signature hot wallet. If sufficient, a withdrawal transaction from the hot multi-signature wallets to the cash withdrawal address on the asset chain is signed by the managing Senators, and the withdrawal is completed after satisfying the multi-signature condition. If the balance is insufficient, an asset recharge from the multi-signature cold wallet to the multi-signature hot wallet is required. A Citizen will initiate the transfer process after the assets in the hot wallet are determined to be sufficient.

Cross-link withdrawal is divided into two steps:

1) The user initiates a withdrawal request on the HyperExchange, the request awaits to be packaged by the Citizen (average 2.5 seconds), and waits for no less than two thirds of the Senator's signatures on the withdrawal request to be collected. If more than one-third of Senator's do not approve the request, the transaction is void.

2) When the packaged transaction collects no less than two thirds of signatures from Senators, it will generate a corresponding asset chain debit transaction. This transaction will be broadcasted by a Citizen to the corresponding asset chain for confirmation (needing to wait for the corresponding number of block confirmations). On the HyperExchange, a Citizen packages this withdrawal transaction. When the transaction is approved by the HyperExchange, the cash withdrawal transaction is completed.

The specific technical implementation process is:

After the user initiates the withdrawal application, a Citizen initiates the corresponding original transaction from the multi-signature address transfer to the user's target address, and the original transaction is packaged in the src_trx, and broadcasted to the HyperExchange.

After the Senator synchronizes to src_trx and then verifies, the src_trx signature is wrapped to the sig_trx and then Citizen conducts bookkeeping in HyperExchange.

When the entire network collects enough Senator signatures, a Citizen packages the collected signatures to generate a complete implementation transaction, and then calls the corresponding asset chain's transaction broadcast interface in the light wallet component built in the Indicator and broadcasts it to the corresponding assets chain.

The underlying operational flow chart of cross-chain withdrawals is as follows:

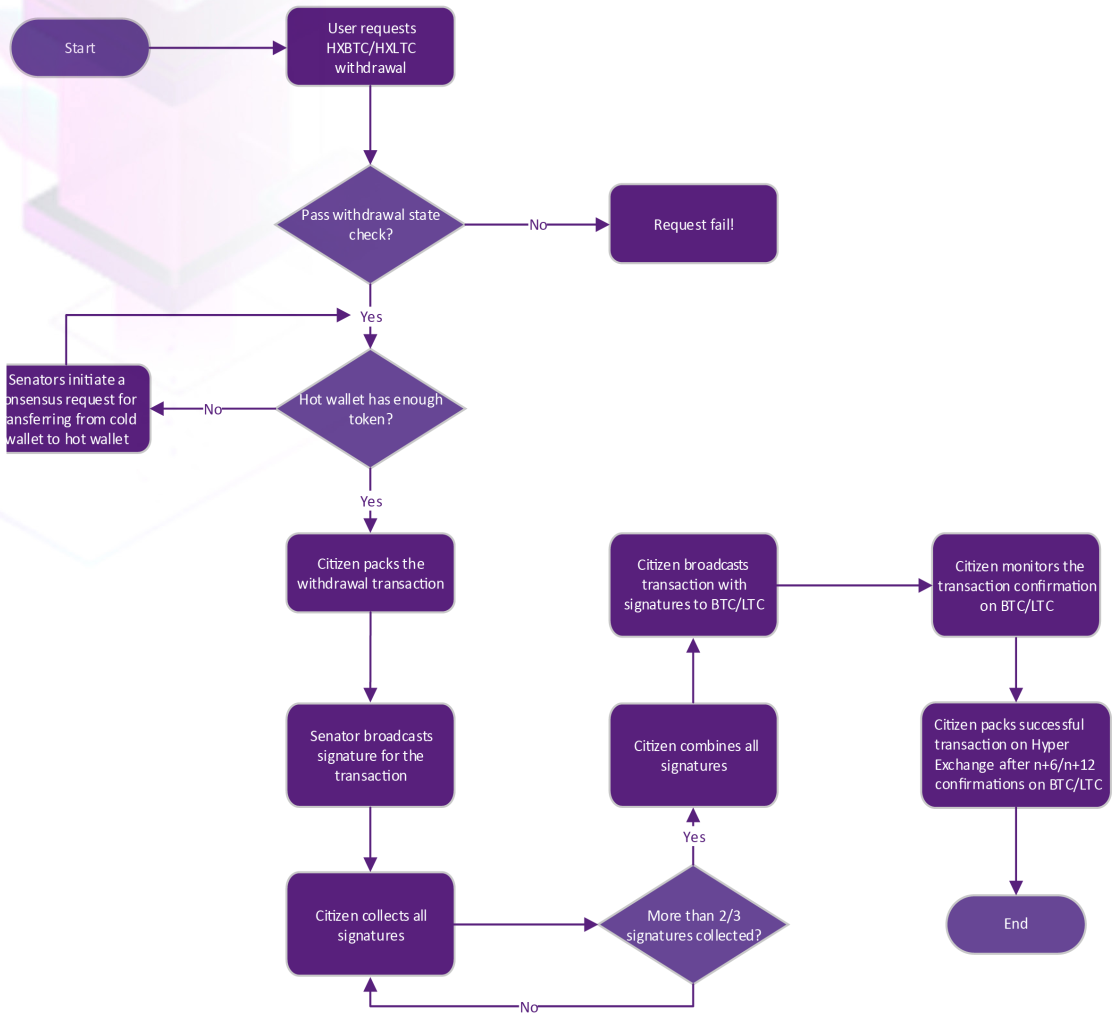

Start
User requests HXBTC/HXLTC withdrawal
Pass withdrawal state check?
No
Request fail!
Yes
Senators initiate a onsensus request for ansferring from cold wallet to hot wallet
No
Hot wallet has enough token?
Yes
Citizen packs the withdrawal transaction
Senator broadcasts signature for the transaction
Citizen collects all signatures
More than 2/3 signatures collected?
No
Yes
Citizen combines all signatures
Citizen broadcasts transaction with signatures to BTC/LTC
Citizen monitors the transaction confirmation on BTC/LTC
Citizen packs successful transaction on Hyper Exchange after n+6/n+12 confirmations on BTC/LTC
End

Figure 4: The cross-chain withdrawal flowchart

3.1.3 Sidechain asset security

3.1.3.1 Financial balance strategy

If assets in the hot multi-signature account exceeds 3 times of the limit, a fund balancing process is initiated to transfer the assets in the multi-signature hot wallet address to the multi-signature cold wallet address.

The specific technical implementation plan is:

A senator initiates a financial balance process, after the chain is packaged, another Senator will perform verification. After successful verification, the signature of the transaction will be broadcast and then packaged by Citizen on the HyperExchange.

When a Citizen collects no less than two thirds of a Senator’s signatures, the original chain transaction of HIOU will be confirmed, and finally be broadcasted to the original chain to complete the process of financial balancing.

3.1.3.2 Dynamic balancing of funds

A dynamic balancing process of funds is executed every 10,000 blocks, and the total assets of the multi-signature hot wallet are adjusted to the set range, according to the number of assets in said multi-signature hot wallet.

The aforementioned financial dynamic balancing transaction is created by the Citizen who generated the block. After being broadcasted, the transaction will be signed by no less than two thirds of all Senators.

3.2 Post-quantum

Currently within the blockchain systems represented by Bitcoin, SHA-256 hash calculations and ECDSA elliptic curve cryptography serve as the most basic security protection along the Bitcoin network.

The approaching of quantum computers threatens almost all cryptography, which, among other things, is the foundation for internet security. Quantum computing is also a huge threat to blockchain-based cryptocurrencies. In particular, the quantum algorithm, Shor [Sho99] for computing discrete logarithms, can break the ECDSA signature scheme used by Bitcoin.

When this happens, one can easily derive the private key from the public key presented in Bitcoin transactions. Therefore, when a transaction is broadcast to the network, it is at risk before being placed on the blockchain. In particular, attackers may intercept the transaction, obtain the public key published with it, and compute the corresponding private key. Then, the attacker can modify the transaction content and generate a valid signature of the modified transaction. If the new transaction is placed on the blockchain before the original one, the attacker can steal the bitcoin from the original output address.

The HCASH project aims to build a secure, efficient, robust and reliable decentralised system. Among all the features HCASH is designed to present, the security against quantum computing is one of the most prominent. Many signature schemes have been proposed and are presumed to be quantum-secure. HCASH incorporates two of the most popular post-quantum signature schemes, BLISS [DDLL13] and MSS/LMS [BDH11, LM95]. BLISS is currently the most efficient post-quantum signature scheme with the smallest public key and signature sizes. For MSS/LMS, its security hypothesis is weak (that is, its security only relies on the security of the underlying hash function). If the underlying hash function is secure, then the solution is secure. [ABL+17]. Both schemes have been thoroughly analysed and evaluated by the academic community with respect to their security.

There are two problems to be solved in implementing BLISS and MSS/LMS signature:

1) The disadvantage of BLISS is its difficulty to implement in a secure way, as the discrete Gaussian sampling in the BLISS algorithm is susceptible to side channel attacks [PBY17].
2) Although BLISS and MSS/LMS signatures are comparably small among all post-quantum signature schemes, they are still larger than the traditional ECDSA signatures. Large signatures will increase the size of the transactions and reduce the TPS, and the amount of payments the system is able to provide.

After nearly a year of technological development, the HCASH team has made breakthroughs for the first problem. We have found a countermeasure against side-channel attacks on BLISS, by incorporating multiple side-channel-proof techniques in our implementation. In response to the second problem, to alleviate the negative impact of long signatures, we propose a new segregated witness scheme which can be used to solve the problem – related results will be described in detail below.

We will introduce our post-quantum scheme from five aspects: first, we will give a brief review of the post-quantum signature schemes in the literature, and explain the reason for choosing BLISS and MSS/LMS for HCASH; then we will demonstrate the BLISS scheme, and present our countermeasures against the side-channel attacks on BLISS, the following part describing the MSS/LMS scheme, next we will give a brief of the block transmission protocol. Finally, we conclude the exploration of post-quantum Signature Schemes in Hcash.

3.2.1 Post-quantum Signature Schemes

Many public-key signature schemes have been proposed in literature, and are presumed to be quantum-secure. Generally speaking, existing schemes can be classified into four categories:

1) Hash-based signature schemes, whose security relies on collision resistant hash functions, including MSS [Mer89, DOTV08], LMS [LM95, MCF17], XMSS [BDH11], SPHINCS [BHH+15] and NSW [NSW05];

2) Multivariate-polynomial-based signature schemes, whose security relies on Multivariate quadratic polynomial mapping on finite fields (trapdoor single function), including RAINBOW [DS05].

3) Code-based signature schemes, whose security relies on error-correcting code theory, including CFS [CFS01] and QUARTZ [PCG01];

4) Lattice-based signature schemes, whose security relies on difficulty problem on lattice, including GVP [GPV08], LYU [Lyu12], GLP [GLP12], BLISS [DDLL13], DILITHIUM [DLL+17] and NTRU [MBDG14];

Table 2 presents the signature sizes and public-key sizes of the schemes [ABL+17, BDH11, dOL15]

Type	Name	Security (bits)	PK Size (kb)	Sig. Size (kb)
I	LMS	128	0.448	22.624
	MSS	128	0.256	30.752
	XMSS	100	13.568	15.384
	SPHINCS	128	8	328
	NSW	128	0.256	36
II	GPV	100	300	240
	LYU	100	65	103
	GLP	100	12	9
	BLISS	128	7	5
	DILITHIUM	138	11.8	21.6
III	CFS	83	9216	0.1
	QUARTZ	80	568	0.128
IV	RAINBOW	160	305	0.244

After taking the public-key and signature sizes into consideration, we found that the most practical options are lattice and hash based signature schemes. The most significant advantage of hash-based schemes is having provable security, at least in the random oracle model. Currently, the most effective quantum-attack against the hash-based schemes is Grover's searching algorithm [Gro96]. As a result of the Grover's algorithm, the quantum security level of a hash-based scheme is only half of the classical security level. Therefore, at the same quantum security level, lattice-based schemes have advantage in signature and public key sizes.

MSS was developed by Ralph Merkle in the late 1970s, based on one-time-signatures (OTS) such as Winternitz (WOTS). The security of MSS depends only on the security of hash functions, i.e. collision resistance and second pre-image resistance. LMS is one of the many variations of MSS which improves MSS with respect to its security and efficiency. LMS was proposed by Leighton and Mechali in 1995, and has gone through a number of revisions based on the analysis by Katz [Kat16, Kat] and Fluhrer [Flu17]. We decide to use the most updated LMS algorithm [MCF17] as the hash-based signature in HCASH. We use SHA3 as the underlying hash function, which is strong in collision resistance and second pre-image resistance.

Compared to other existing post-quantum signature schemes, the BLISS algorithm has the shortest signature and public key sizes of all. The security of BLISS relies on the hardness of the NTRU problem, and the assumption that solving this problem is equivalent to finding a short vector in the NTRU lattice. The disadvantage of BLISS is its difficulty to implement in a secure way, as it is susceptible to side channel attacks [PBY17].

DILITHIUM is another lattice-based signature scheme which has higher levels of security, though at the cost of larger public keys and signatures, in comparison with BLISS. Currently there are no side-channel attacks on DILITHIUM, which makes it seemingly more secure than BLISS. However, DILITHIUM was proposed only recently, therefore its security in implementation is not as thoroughly analysed as that of BLISS. Considering our countermeasures against the revealed side-channel attacks on BLISS, and the advantage in efficiency of BLISS, together with the significant impact of the public key and signature size on the performance of blockchain based cryptocurrencies, we think BLISS is a better choice compared to DILITHIUM.

HCASH implements both MSS/LMS and BLISS schemes, taking the advantage of the efficiency of BLISS and the security of MSS/LMS. To address the side-channel attacks on BLISS, we incorporate many side-channel-proof techniques in our implementation.

3.2.2 BLISS Algorithm

The BLISS signature scheme is an improvement to the LYU [Lyu12] scheme, which is done by replacing the distribution of the signature, i.e. discrete Gaussian distribution, with a bimodal Gaussian distribution. This modification significantly reduces the reject sampling rate, which is used to force the signature distribution into a fixed Gaussian distribution, so as to eliminate any information leakage by the distribution of signature. The improvement is explained by Fig. 5.

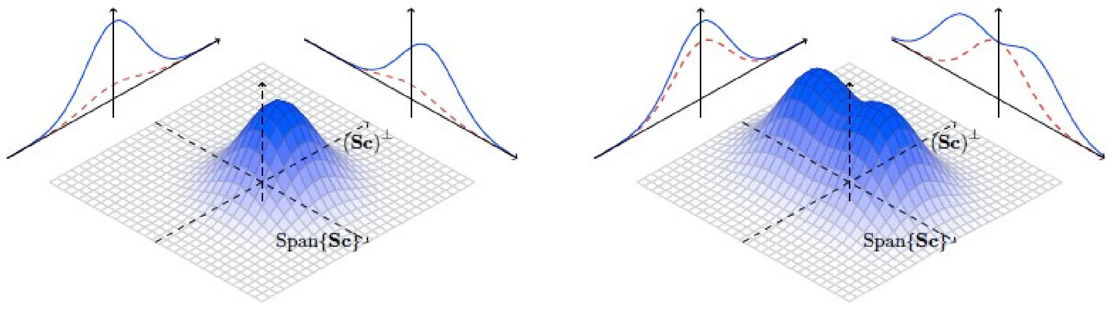
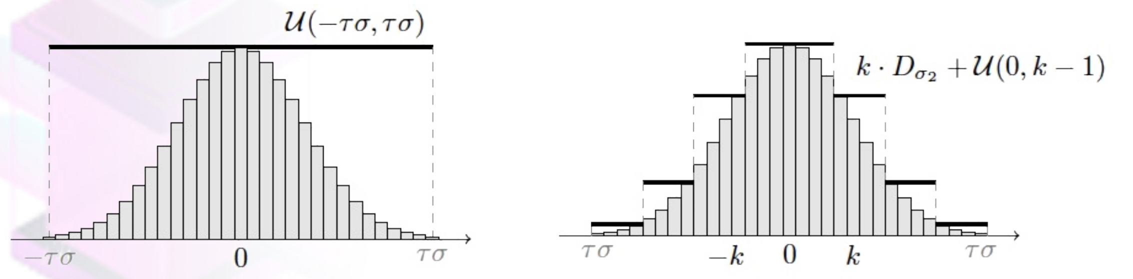


Figure 5: Improvement of Rejection Sampling with Bimodal Gaussian Distributions

Here we give a brief explanation of the basic idea of BLISS.

BLISS proposes an efficient discrete Gaussian sampling algorithm. Before that, all known algorithms to sample according to a distribution statistically close to a discrete Gaussian distribution on a lattice required either long-integer arithmetic at some point, or large memory storage. The Gaussian sampling algorithm proposed in BLISS can efficiently sample discrete Gaussians without resorting to large precomputed tables, nor evaluations of transcendental function. First, algorithms are proposed to sample according to a Bernoulli distribution with bias of the form exp(x/f) and 1/cosh(x/f)) without actually computing transcendental functions. Then, BLISS builds an efficient distribution based on the binary discrete Gaussian distribution, and applies rejection sampling to this distribution to obtain the target discrete Gaussian distribution. This procedure is explained by Fig.6.



(a) from uniform distribution (repetition rate ≈ 10)

(b) from our adapted distribution (repetition rate ≈ 1.47)

Figure 6: Sample from Discrete Gaussian by Rejection Sampling

The algorithm for sampling Bernoulli distribution with bias of the form exp*(x/f)* is presented in Algorithm 1. Bernoulli distribution with bias of the form 1/cosh*(x/f)*) is sampled based on the fact that $\mathcal{B}_{1/\cosh(x)} = \mathcal{B}_{\exp(-|x|)} \oslash (\mathcal{B}_{1/2} \vee \mathcal{B}_{\exp(-|x|)})$ where $\mathcal{B}_a \oslash \mathcal{B}_b$ is defined as $\mathcal{B}_{a/(1-(1-a)b}$. The binary discrete Gaussian distribution is sampled by Algorithm 2. The final discrete Gaussian distribution sampling algorithm is presented in Algorithm 3.

Algorithm 1 Sampling Bernoulli Distribution with Bias exp*(x/f)*

```
Input: $x \in [0, 2^\ell)$ an integer in binary form $x = x_{\ell-1} \cdots x_0$
1: Compute $c_i = \exp(-2^i/f)$ for $0 \le i \le \ell - 1$
2: for $i = \ell - 1$ to 0 do
3:     if $x_i = 1$ then
4:         Sample $A_i$ according to Bernoulli distribution of probability $c_i$
5:         if $A_i = 0$ then return 0
6:     end if
7: end for
```

Algorithm 2 Sampling Binary Discrete Gaussian Distribution

```
Output: An integer $x \in \mathbb{Z}^+$ according to binary discrete Gaussian distribution
1: Generate a bit $b \leftarrow \mathcal{B}_{1/2}$
2: if $b = 0$ then return 0
3: for $i = 1$ to $\infty$ do
4:     Draw random bits $b_1, \cdots, b_k$ for $k = 2i - 1$
5:     if $b_1 \cdots b_k \neq 0 \cdots 0$ restart
6:     if $b_k = 0$ then return $i$
7: end for
```

Algorithm 3 Sampling Discrete Gaussian Distribution

```
Input: An integer $k \in \mathbb{Z}$
Output: An integer $z \in \mathbb{Z}^+$ according to discrete Gaussian distribution of deviation
    $k\sigma_2$
1: Sample $x$ by Algorithm 2
2: Sample $y$ uniformly in $\{0, \cdots, k-1\}$
3: Compute $z \leftarrow kx + y$
4: Sample $b \leftarrow \mathcal{B}_{\exp(-y(y+2kx)/(2\sigma^2))}$
5: if not $b$ then restart
6: if $z = 0$ then restart with probability $1/2$
7: Generate a bit $b \leftarrow \mathcal{B}_{1/2}$ and return $(-1)^b z$
```

BLISS also replaces the ring $\mathcal{R} = \mathbb{Z}$ from which the matrix elements are taken, using the NTRU ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(x^n + 1)$ instead. The size of the public key matrix $\boldsymbol{A}$ and private key matrix $\boldsymbol{S}$ are reduced to 1×2 and 2×1 respectively. The private key of BLISS consists of two polynomials $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ from $\mathcal{R}_q$, and the public key consists of one polynomial $\boldsymbol{a}_1$ (the other element of the matrix $\boldsymbol{A}$ is a constant $q - 2$). The signature consists of a pair of polynomials $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, together with a challenge $\boldsymbol{c}$, which is a sparse polynomial with coefficients from {0,1}.

BLISS reduces the signature size by compressing the polynomial $\boldsymbol{z}_2$ without compromising the security. In particular, BLISS replaces $\boldsymbol{z}_2$ by $z_2^\dagger$ which equals $\lfloor \boldsymbol{u} \rceil_d - \lfloor \boldsymbol{u} - \boldsymbol{z}_2 \rceil_d$, where $\lfloor \cdot \rceil_d$ means dropping d bits from each coefficients of the polynomial. Compared to $\boldsymbol{z}_2$, the coefficients of $z_2^\dagger$ are much smaller, thus easier to compress by Huffman codes.

The BLISS scheme has been further optimised in efficiency, and the improved version is called BLISS-B [Duc14]. BLISS-B replaces the product of polynomials $\boldsymbol{S} \cdot \boldsymbol{c}$ by a procedure GreedySC($\boldsymbol{S}$,$\boldsymbol{c}$), which effectively computes $\boldsymbol{S} \cdot \boldsymbol{c}'$ where $\boldsymbol{c}'$ differs from $\boldsymbol{c}$ in the sign of a subset of the nonzero coefficients. GreedySC($\boldsymbol{S}$,$\boldsymbol{c}$) efficiently derives a $\boldsymbol{c}'$ such that $\|\boldsymbol{S} \cdot \boldsymbol{c}'\|_2$ is minimised, reducing the reject sampling rate by a considerable factor. The algorithm for computing GreedySC($\boldsymbol{S}$,$\boldsymbol{c}$) is presented in Algorithm 4.

Algorithm 4 GreedySC

```
Input: a matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n]$ and a binary vector $\boldsymbol{c}$
Output: $\boldsymbol{v} = \boldsymbol{S}\boldsymbol{c}'$ for some $\boldsymbol{c}' \equiv \boldsymbol{c} \bmod 2$
1: $\boldsymbol{v} \leftarrow \boldsymbol{0} \in \mathbb{Z}^m$
2: for $i \in \mathcal{I}_{\boldsymbol{c}}$ do
3:     $\zeta_i \leftarrow \mathrm{sgn}(\langle \boldsymbol{v}, \boldsymbol{s}_i \rangle)$
4:     $\boldsymbol{v} \leftarrow \boldsymbol{v} - \zeta_i \cdot \boldsymbol{s}_i$
5: end for
```

The final algorithms of the BLISS scheme are presented in Alg. 5, Alg. 6 and Alg. 7.

3.2.2.1 BLISS Against Side-channel Attacks

BLISS has been demonstrated to be vulnerable to side-channel attacks [EFGT17, PBY17]. A major part of the vulnerability comes from the discrete Gaussian sampling which plays an important role in lattice cryptography [MW17]. Here are the advanced BLISS algorithms that resist side-channel attacks.

Algorithm 5 BLISS Key Generation

```
Output: Key pair $(\boldsymbol{A}, \boldsymbol{S})$ such that $\boldsymbol{AS} = q \bmod 2q$
1: Choose $\boldsymbol{f}, \boldsymbol{g}$ as uniform polynomials with exactly $d_1$ entries in $\{\pm 1\}$ and $d_2$ entries
   in $\{\pm 2\}$
2: $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2)^T \leftarrow (\boldsymbol{f}, 2\boldsymbol{g} + 1)^T$
3: if $N_\kappa(\boldsymbol{S}) \geq C^2 \cdot 5 \cdot (\lceil \delta_1 n \rceil + 4\lceil \delta_2 n \rceil) \cdot \kappa$ then
4:     restart
5: end if
6: $\boldsymbol{a}_q = (2\boldsymbol{g} + 1)/\boldsymbol{f} \bmod q$ (restart if $\boldsymbol{f}$ is not invertible)
7: Output$(\boldsymbol{A}, \boldsymbol{S})$ where $\boldsymbol{A} = (2\boldsymbol{a}_q, q - 2) \bmod 2q$
```

Algorithm 6 BLISS Signature Algorithm

```
Input: Message $\mu$, public key $\boldsymbol{A} = (\boldsymbol{a}_1, q-2) \in \mathcal{R}_{2q}^{1\times 2}$, secret key $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2)^T \in \mathcal{R}_{2q}^{2\times 1}$
Output: A signature $(\boldsymbol{z}_1, \boldsymbol{z}_2^{\dagger}, \boldsymbol{c})$ of the message $\mu$
1: $\boldsymbol{y}_1, \boldsymbol{y}_2 \leftarrow D_{\mathbb{Z}^n,\sigma}$
2: $\boldsymbol{u} = \zeta \cdot \boldsymbol{a}_1 \cdot \boldsymbol{y}_1 + \boldsymbol{y}_2 \bmod 2q$
3: $\boldsymbol{c} \leftarrow H(\lfloor \boldsymbol{u} \rceil_d \bmod p, \mu)$
4: $(\boldsymbol{v}_1, \boldsymbol{v}_2) \leftarrow \mathsf{GreedySC}(\boldsymbol{s}, \boldsymbol{c})$
5: Choose a random bit $b$
6: $(\boldsymbol{z}_1, \boldsymbol{z}_2) \leftarrow (\boldsymbol{y}_1, \boldsymbol{y}_2) + (-1)^b \cdot (\boldsymbol{v}_1, \boldsymbol{v}_2)$
7: Continue with probability $1/\left(M \exp\left(-\frac{\|v\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \boldsymbol{z}, \boldsymbol{v} \rangle}{\sigma^2}\right)\right)$ otherwise restart
8: $\boldsymbol{z}_2^{\dagger} \leftarrow (\lfloor \boldsymbol{u} \rceil_d - \lfloor \boldsymbol{u} - \boldsymbol{z}_2 \rceil_d) \bmod p$
9: Output $(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{c})$
```

Algorithm 7 BLISS Verification Algorithm

```
Input: Message $\mu$, public key $\boldsymbol{A} = (\boldsymbol{a}_1, q-2) \in \mathcal{R}_{2q}^{1\times 2}$, signature $(\boldsymbol{z}_1, \boldsymbol{z}_2^{\dagger}, \boldsymbol{c})$
Output: Accept or Reject the signature
1: if $\|(\boldsymbol{z}_1 \mid 2^d \cdot \boldsymbol{z}_2^{\dagger})\|_2 > B_2$ then Reject
2: if $\|(\boldsymbol{z}_1 \mid 2^d \cdot \boldsymbol{z}_2^{\dagger})\|_\infty > B_\infty$ then Reject
3: Accept iff $\boldsymbol{c} = H(\lfloor \zeta \cdot \boldsymbol{a}_1 \cdot \boldsymbol{z}_1 + \zeta \cdot q \cdot \boldsymbol{c} \rceil_d + \boldsymbol{z}_2^{\dagger} \bmod p, \mu)$
```

Much effort has been devoted in designing a discrete Gaussian sampling algorithm with high efficiency and security. Another source of information leakage comes from the rejection sampling, which causes the program execution and memory access to depend on the random data.

First, we implement the Bernoulli sampler with probability in the form of e^x in constant-time way. The Bernoulli sampler is used in the discrete Gaussian sampling algorithm utilised in BLISS [DDLL13], and the execution procedure depends on the bits of x. Specifically, the sampler invokes a table lookup for each bit of value 1 in x. We eliminate this potential source of leakage by forcing the program to execute the lookup regardless of the bit value.

Secondly, we prevent the attackers from perceiving the sampled $\boldsymbol{y}$ by splitting $\boldsymbol{y}$ into the sum of two independently sampled $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Then we compute $\boldsymbol{Ay}$ by first computing $\boldsymbol{Ay}_1$ and $\boldsymbol{Ay}_2$ and take the addition. We carefully select the standard deviation and other parameters of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ such that the statistical distance $\Delta(\boldsymbol{y}, \boldsymbol{y}_1+\boldsymbol{y}_2)$ is negligible according to Theorem 3.1 in [Pei10]. Due to the overhead brought by the protection techniques, the signature generation procedure is three times slower than the unprotected BLISS, which is still fast enough.

Therefore, the scheme we propose here has security and applicability at the same time.

3.2.3 The MSS and LMS Algorithms

The hash-based signature schemes start from a one-time signature scheme (OTS), that is, a signature scheme where each key pair cannot be used to sign more than one message. Currently the most efficient OTS scheme is the Winternitz scheme (WOTS) [BBD09]. The core idea of WOTS is to iteratively apply a function on a secret input, whereas the number of iterations depends on the message to be signed.

The functions are members of the function family

$$F(n) = \{f_k := \{0,1\}^n \to \{0,1\}^n\} \mid k \in \{0,1\}^n$$

Let $f_k^0(x) := x, f_k^1(x) := f_k(x)$ and $f_k^i(x) := f_{f_{k-1}(x)}(x)$.

The WOTS is parameterised by the hash size n, and the compression level $w \in \mathbb{Z}^+, w > 1$.

WOTS key generation. To generate a pair of key for WOTS, choose a random value $x \in \{0,1\}^n$. Let ℓ be computed as follows

$$\ell_1 = \left\lceil \frac{n}{\log(w)} \right\rceil, \quad \ell_2 = \left\lfloor \frac{\log(\ell_1(w-1))}{\log(w)} \right\rfloor + 1, \quad \ell = \ell_1 + \ell_2$$

The secret key sk consists of ℓ uniformly randomly chosen bit-strings of length n.

The public key pk is computed from sk as follows: pk₀ = x, pk$_i$ = $f_{\mathsf{sk}_i}^{w-1}(x)$ for 1 ≤ i ≤ ℓ.

WOTS signature generation. To sign an n-bit message $M = (M_1, \cdots, M_{\ell_1})$ given in base-w representation, i.e. $0 \leq M_i \leq w-1$. First compute the checksum

$$C = \sum_{i=1}^{\ell_1} (w - 1 - M_i)$$

and represents C in base w as (C_1,⋯ ,$C_{\ell 2}$). Then we set $B = (b_1, \cdots, b_\ell) = M \| C$. The signature is computed as $\sigma = (\sigma_1, \cdots, \sigma_\ell) = (f_{\mathsf{sk}_1}^{b_1}(x), \cdots, f_{\mathsf{sk}_\ell}^{b_\ell}(x))$.

WOTS signature verification. To verify the signature, first computes the basew string B as described above, then check that

$$(f_{\sigma_1}^{w-1-b_1}(x), \cdots, f_{\sigma_\ell}^{w-1-b_\ell}(x)) = (\mathsf{pk}_1, \cdots, \mathsf{pk}_\ell)$$

The Merkle signature scheme (MSS) works with any cryptographic hash function and any one-time signature scheme. MSS uses a Merkle tree to link $N = 2^H$ OTS public keys together, where H is the height of the tree. The Merkle tree root is then used as the public key of MSS. The algorithm for computing the Merkle root is called Treehash, see Algorithm 8.

To generate a signature of a message M, take the secret key of the next unused OTS public key, and generate a signature of OTS. The signature of MSS consists of the OTS signature, the OTS public key, and an authentication path to prove the existence of the OTS public key in the Merkle tree. To verify the signature, first check the validity of the OTS signature against the OTS public key, then verify that the OTS public key is a valid leaf of the Merkle tree with the authentication path.

Algorithm 8 Threehash

```
Input: Height $H > 2$
Output: Root of the Merkle tree
1: for $j = 0, \cdots, 2^H - 1$ do
2:   Compute the $j$th leaf: $\text{NODE}_1 \leftarrow \text{LEAFCALC}(j)$
3:   while $\text{NODE}_1$ has the same height as the top node on STACK do
4:     Pop the top node from the stack: $\text{NODE}_2 \leftarrow \text{STACK.pop}()$
5:     Compute their parent node: $\text{NODE}_1 \leftarrow g(\text{NODE}_2 \| \text{NODE}_1)$
6:   end while
7:   Push the parent node on the stack: $\text{STACK.push}(\text{NODE}_1)$
8: end for
9: Let $R$ be the single node stored on the stack: $R \leftarrow \text{STACK.pop}()$
10: return $R$
```

The MSS private key consists of 2^H OTS secret keys. Storing such a huge amount of data is not feasible for most practical applications. Space can be saved by using a deterministic pseudo random number generator (PRNG) and storing only the seed of that PRNG. Then each one-time signature key must be generated twice, once for the MSS public key generation and once during the signing phase. In addition to the private key size, using a PRNG for the one-time signature key generation has another benefit. It makes MSS forward secure as long as PRNG is forward secure.

Computing the Merkle root and an authentication path from a PRNG seed is a computation consuming work. Meanwhile, saving the entire Merkle tree in the memory is efficient but causes large memory usage. To leverage the storage and computation cost, various Merkle tree traversing algorithms are proposed. The main purpose of Merkle tree traversal is to sequentially output the leaf values of all nodes, together with the associated authentication path. Currently the most efficient Merkle tree traversal algorithm is one proposed by Szydlo in [Szy04]. See Algorithm 9 for details.

LMS proposed by Leighton Micali is a modification to the MSS with respect to the security and efficiency. Here we briefly introduce the modification to the secret key, the public key and the signature respectively.

As for the one-time **secret key**, LMS prepend security fields including typecode, I and q as a prefix to the plain secret key $\mathbf{x} = (x_0, \ldots, x_{\ell-1})$ in MSS, and the final secret key is encoded as

$$\text{typecode}\|I\|q\|x_0\|\ldots\|x_{\ell-1}$$

Algorithm 9 Logarithmic Merkle Tree Traversal

```
1: Set $s = 0$
2: for each $h \in [0, H-1]$ output $\mathrm{AUTH}_h$
3: for all $h$ such that $2^h$ divides $s+1$ do
4:     Set $\mathrm{AUTH}_h$ be the sole node value in $\mathrm{STACK}_h$
5:     Set startnode $= (s + 1 + 2^h) \oplus 2^h$
6:     $\mathrm{STACK}_h.initialize(\text{startnode}, h)$
7: end for
8: for $i$ from 1 to $2H-1$ do
9:     Let $\ell_{\min}$ be the minimum of $\mathrm{STACK}_h.low$
10:    Let $focus$ be the least $h$ such that $\mathrm{STACK}_h.low = \ell_{\min}$
11:        $\mathrm{STACK}_h.update$
12: end for
13: Set $s = s + 1$
14: if $s < 2^H$ go to line 2
```

Hence, the secret key in LMS comsumes 24 bytes more than that of MSS. When it comes to the **public key** (i.e., the Merkle root), a similar pre-padding is employed, The public key of one LMS is encoded as

$$\text{sigtype}\|\text{otstype}\|I\|\nu$$

where sigtype and otstype are both typecode, and ν is the origin public key for MSS.

A LMS **signature** σ is formatted as

$$q \| \sigma_{ots} \| \text{sigtype} \| A_s$$

where $\sigma_{ots} = \text{otstype} \| C \| \sigma_0 \| \ldots \| \sigma_\ell$ is the one-time signature, and $A_s = \{a_i\}_{i=0}^{\ell-1}$.

In MSS, the one-time public key is $\mathbf{y} = (y_0, y_1, \ldots, y_\ell)$, and it can be derived from σ_{ots} and the message M if σ_{ots} on M is signed by the paired secret key of $\mathbf{y}$. In LMS, the one-time public key $\mathbf{y} = (y_0, y_1, \ldots, y_\ell)$ to verify the one-time signature σ_{ots} part of σ is compressed as

$$\mathbf{y} = \text{typecode} \| I \| q \| H(I \| q \| \text{D\_PBLC} \| y_0 \| y_1 \| \ldots \| y_\ell)$$

while the counterpart in MSS stores each y_i publicly. Every time σ_{ots} needs verifying, the one-time public key candidate $\mathbf{z} = (z_0, z_1, \ldots, z_\ell)$ can be derived from σ_{ots} = $(\sigma_0, \sigma_1 \ldots, \sigma_\ell)$, and then check the extended digest of $\mathbf{z}$ as $H(I \| q \| \text{D\_PBLC} \| z_0 \| z_1 \| \ldots \| z_\ell)$ against the corresponding component $H(I \| q \| \text{D\_PBLC} \| y_0 \| y_1 \| \ldots \| y_\ell)$ in the public key.

Both LMS and MSS are based on the Merkle tree. Other intermediate routines such as updating the authentication path $A_s = (a_0, a_1, \ldots, a_\ell)$ through traversing the tree, are the same. The major difference is the compression of the one-time public y in the σ_{ots} component.

3.2.4 Block Transmission Protocol

Although BLISS and MSS/LMS signatures are comparably small among all the postquantum signature schemes, they are still larger than the traditional ECDSA signatures. Large signatures will increase the size of the transactions and reduce the amount of payments the system is able to provide. To mitigate the negative effects exerted by the large signatures, we design a new block transmission protocol based on the idea of segregated witness.

Simply speaking, the idea of segregated witness is to separate the signatures from the transaction body. Specifically, the transaction signature is not taken into calculation of the transaction ID. We further improve this idea by allowing users to transmit only the transaction IDs when passing a block. In this way, the communication cost of transmitting blocks decreases significantly. On receiving a block, a node checks if he/she has already obtained (and saved into the memory pool) all the transactions whose IDs are in the block. If he/she misses some transaction, he/she will ask the peer who gave him/her the block for the missing transaction. Fig. 7 (see below) is an example of the transmission procedure of a newly generated block.

3.2.5 Conclusion

HCASH supports multiple postquantum signature schemes, and is compatible with traditional ECDSA signatures. We innovate a new communication protocol based on segregated witness. This protocol relieves the pressure caused by long signatures on the network, and implements anti-quantum signature. It will provide a high-level security protection scheme for HCASH users in a future quantum environment.

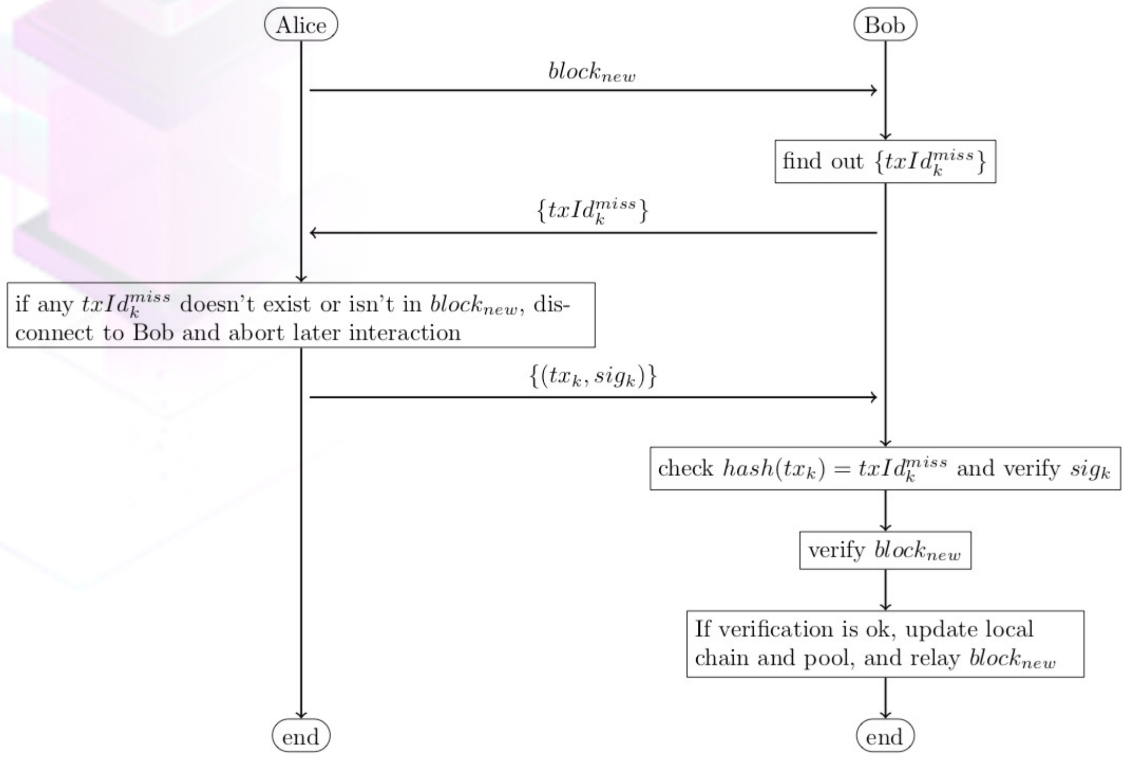

Alice
Bob
$block_{new}$
find out $\{txId_k^{miss}\}$
$\{txId_k^{miss}\}$
if any $txId_k^{miss}$ doesn't exist or isn't in $block_{new}$, disconnect to Bob and abort later interaction
$\{(tx_k, sig_k)\}$
check $hash(tx_k) = txId_k^{miss}$ and verify sig_k
verify $block_{new}$
If verification is ok, update local chain and pool, and relay $block_{new}$
end
end

Figure 7: Broadcast of a new block

The block transmission protocol based on segregated witness will significantly cut down the communication cost brought by large postquantum signatures, and this protocol can be generalised and transplanted into other blockchain and cryptocurrency systems.

The post-quantum signature has been implemented on the upgraded HyperCash main chain, and will also be implemented on HyperExchange. At that time, the two main chains of HCASH will both have post-quantum features, making HCASH one of the few cross-chain ecosystems with extremely high security in the blockchain field.

3.3 Smart Contracts

The HCASH Ecosystem allows users to process complex cross-chain transactions and asset verification through Turing complete smart contracts in the HyperExchange chain, providing developers with support for DAPP development, laying the foundation for cross-chain distributed business applications.

To bring more ecosystem builders to develop DAPPs on our platform, HyperExchange is committed to building the infrastructure needed for developers to build decentralised applications, and to creating an environment compatible with different programming languages. This enables developers who have skills with various development stacks to connect quickly and develop DAPPs conveniently and efficiently on the platform.

3.3.1 Fully Compatible Development Language

- The bytecode specification of HCASH is Turing complete and custom designed for smart contracts, this bytecode is the implementation specification for smart contract virtual machines. The system also uses a compiler implementation that provides static-type high-level programming languages such as C#, Java, TypeScript, etc. to generate smart contract bytecode.
- HCASH also continuously supplements SDKs in various languages to facilitate developers to port to different platforms, thus continuously attracting DAPP developers from different industries.
- HCASH simplifies the preparation needed by developers through a well-designed API and native smart contracts, enabling development on the platform to begin with ease.
- HCASH includes some basic libraries for common numerical operations, as well as string operations. In addition, it also provides some built-in function libraries such as on-chain queries, transactions, and so on. These libraries can be invoked within smart contracts.
- After the smart contract is deployed on the chain, it can be executed directly by users or by other smart contacts, and in turn, used to execute other smart contracts or built-in native contracts. The smart contract can also be used to deposit and/or withdraw assets.
- In different industries, there are different technical needs and focuses when developing DAPPs. For instance, different technologies are required to develop decentralised social networks, decentralised storage systems, decentralised games, and decentralised financial services. We will gradually form our own DAPP standard after a comprehensive exploration of both the DAPP industry and Turing complete smart contracts, and will apply these standards into native contracts to make it more convenient for developers to iterate their DAPP development cycle. Ultimately, we would like to see blockchain applications extend their reach to internet users, so that all can experience the values and advantages that blockchain technology can bring.
- In future, HCASH will improve the efficiency of DAPP development by launching an IDE with debugging features, allowing developers to identify problems faster when debugging errors.

3.3.2 Lite Node, Lite Storage

For existing mainstream main chains, the lack of “resource isolation” seems to be a major issue when connections are made to a smart contract. This causes congestion, which inevitably happens along with the expansion of an ecosystem which does not scale well. For example, congestion occurs on the Ethereum network as a result of popular crowdfunding campaigns which cause a sharp increase in the number of transactions at a given time.

Each time a smart contract is executed on the HyperExchange blockchain, a standalone lightweight execution environment is initialised by executing the corresponding contract bytecode on the chain; When accessing data on the chain, native API can also be utilised without invoking all data. In future development, HCASH will continue to improve related technologies, and ultimately create an independent operating environment for each DAPP.

Each smart contract has its own independent state stored as "storage". The execution of any contract transaction which results in a data change in the state storage area of the smart contract ("storage" change), the contract does not retain the full backup of all historical "storage", but only saves the current state of the "storage" and the changed parameters in each change of the "storage".

With this setting, when the user wants to obtain the execution result of a smart contract, the current value of the "storage" can easily be obtained without reading all data, which greatly reduces the user's workload and the data storage required by a node. This saves system resources, improves the processing efficiency of the system, and also calculates the gas on demand according to the actual change of "storage". At the same time, the user can also restore or roll back the historical change value of the "storage".

3.3.3 Concurrency Model To Improve Efficiency

By distributing smart contract transactions to parallel pipelines which are executed simultaneously, HCASH can increase its ability to execute a large number of contracts concurrently, reducing overall transaction execution time.

- For smart contracts which need to be executed within one block time, they are arranged to be executed in different pipelines. The pipeline here refers specifically to the contract processing unit, meaning that different pipelines can simultaneously perform contract executions via different threads.
- Pipelines are divided into serial pipelines and parallel pipelines. Different parallel pipelines can be executed simultaneously while serial pipelines cannot be executed in parallel with any other pipelines. There can be only one serial pipeline at any given time.
- After a pipeline has been initiated, the smart contract transactions are executed in serial.
- A smart contract in a serial pipeline is executed first, and contracts in a parallel pipeline are to be executed subsequently.
- The list of accounts or contract addresses that can be read or modified is called the 'list of dependent addresses'. Smart contracts can be pre-executed by the client to obtain the list of dependent addresses. Normal transactions also have a list of dependent addresses. Conflicting transactions that rely on this address list cannot be placed in different pipelines.
- For transactions that rely on an address list that is not attached to the transaction, transaction fees are increased once it enters the serial pipelines.
- When the list of dependent addresses on the contract statement conflicts with the list of executed dependent addresses, the execution fails, a penalty fee is collected and the empty transaction is consolidated.
- Through parallel execution via pipelines, the TPS, in most cases, can be greatly increased (by a factor of at least 4 times).

3.4 PoW+PoS Hybrid Consensus

3.4.1 Flaws in existing consensus mechanisms

A strong cryptocurrency protocol should strive to provide a more reasonable incentive structure under which the interests of different participants in the system are maintained healthy operation of the system.

Achieving consensus in a community has always been a difficult problem to solve. As we all know, the struggle to upgrade Bitcoin's PoW based consensus protocol has negatively affected the development of the community over the past two to three years, while the over-centralised governance model adopted by Zcash and similar projects has certainly deterred active participation of community members. Currently, none of the decision-making process used by existing projects can preserve the system's ability to adapt to rapidly changing technologies and accommodate the ideological differences among stakeholders without compromising the integrity of blockchain's key value proposition, that is, decentralization.

Relying on a single PoW mechanism is energy intensive and poses the risk of being attacked. With the advent of mining pools, the possibility of becoming a victim to a 51% attack is ever increasing. By obtaining large amounts of computational power (usually through ASIC miners), the attacker(s) can obtain more than 51% of the network's computational power, allowing them to edit transactions, to double-spend or to perform a PoW Denial of Service (PoW-DoS) attack by refusing to include transaction records in the blocks it generates. An attacker may conduct any number of these attacks to benefit financially from double spending or to demand higher transaction fees from users through conducting a PoW-Dos. If an attacker wishes to disrupt or damage the network, it is likely that he will achieve his goal as double-spend or PoW-DoS may cause users to lose confidence in the network's protocol.

In addition, the priority of miners is to make a profit and thus, they will often place their own interests above that of the community. Take Bitcoin as an example, as the network becomes increasingly congested, transactions that pay less fees are not prioritised and remain in the mempool. As a result, transaction fees become higher, with miners' fees reaching .15% of the reward pool and increasing to .5% during periods of heavy congestion. For small transactions, this ultimately means that fees are too high and will negatively affect the commercial use of Bitcoin.

Although a PoS mechanism has its advantages, it cannot effectively solve the main risks faced by cryptocurrencies, such as over-centralisation. Due to economies of scale, the computing power and verification capabilities of large nodes will outperform amateur PoS miners. Additionally, with a PoS protocol the risks of a PoW network being attacked is reduced, but new centralization risks are introduced (large equity holders may attempt to control the system). Another major risk associated with a PoS mechanism is blacklisted 'contaminated' coins which substitute ordinary coins, that are orthogonal to PoS can be mitigated by mixing [3,4] or SNARKs [5,6].

As a result, we have been always searching for a better consensus protocol for blockchain.

3.4.2 PoW + PoS Hybrid Consensus

By learning from its predecessors and incorporating part of the philosophy of Decred, HCASH decided to adopt a hybrid PoW + PoS mining mechanism. We believe that a hybrid consensus mechanism will enable us to reap the benefits of each system, while minimising their disadvantages. In this way, security and effectiveness can be balanced while maintaining interest of participants and ensuring environmental sustainability.

A hybrid PoW + PoS mining mechanism can effectively maximise the engagement of community members. This innovative voting model reinforces stakeholder autonomy, allowing all stakeholders to participate in decision making, to decide whether to implement certain technologies or agreements, and whether the development team should use certain features. Most importantly, the mechanism adopted by HCASH is more advanced than the traditional voting method in the sense that it provides smoother execution. Once the vote is passed, all decisions will be recorded in the blockchain and enforced immediately, thus avoiding the excruciating process of achieving consensus among miners, mining pools, exchanges and wallet service providers. To ensure fairness, voting weight is based on stake, both in the number of tokens held and mining work produced. Any token holders can participate in voting.

As compared to a sole PoW mechanism, a PoW + PoS hybrid consensus mechanism incentivises PoS nodes to maintain more continual online presence nodes. It effectively stabilizes the topology of the network and helps prevent attacks to some extent. In addition, a hybrid consensus also grants token holders the power to generate blocks. If an attacker tried to implement a double spending attack, he would have to possess and control a large number of tokens. In addition, the validator selected by the relevant mechanisms determine which transaction should be included in the block, providing some security against attackers who attempt to extort or destroy the network by rejecting transactions.

Under a PoW + PoS hybrid consensus mechanism, all PoW generated blocks must first be verified by the individual PoS mechanism before becoming valid. That is, PoW is responsible for block generation and PoS is responsible for voting to determine the validity of the block. Both Miners and token holders participate in block generation, each of which check and balance each other. This helps eliminate potential monopolies and ensures better network security. Moreover, a hybrid mechanism can effectively manage hard forks through a voting mechanism.

In addition, a hybrid consensus mechanism protects the interests of new users by preventing exploitations from early investors.

3.4.3 Implementation of PoW + PoS hybrid Consensus Mechanism

Logic in a hybrid consensus mechanism works in the following way. Simply put, PoW miners create blocks and PoS miners confirm the validity of these blocks.

When comparing mining in a PoW + PoS hybrid consensus mechanism with a PoW only network:

Under a hybrid consensus mechanism, PoW miners generate blocks in a certain period of time at the same block height, and PoS miners vote to select which block is valid. Therefore, the block that is generated first may not necessarily be used, and, if PoW miners violate the interests of other community users, their block rewards will be taken.

Whereas when relying solely on a PoW mining mechanism, miner A will obtain block rewards and have the right to record transactions in the block if he is the first to calculate the correct hash value and broadcast it on the chain at a certain block height. That is, if other miners verify miner A's hash value, then the efforts of other competing miners are deemed useless.

Through this we can see that a hybrid consensus mechanism can mitigate potential centralisation risk apparent in a PoW only network. It ensures fair and healthy development within the blockchain ecosystem, and protects the security of the network. In practice, PoW + PoS is not just a simple combination of two mechanisms. There are complex technical implementation paths and voting logic behind them.

3.4.3.1 How a PoW + PoS Hybrid Consensus Works

The HCASH development team have implemented the Decred (and Bitcoin-NG) mining algorithm Blake256 R14 voting mechanism - a practical and flexible PoS solution - on the HyperCash main chain and have optimised relevant rules.

The implementation process is as follows:

1) The mining process starts with PoW, where miners compete to solve a mathematical problem. According to this implementation, the blocks being mined do not contain any transactions.

2) After this moment, the system will switch to PoS. Based on the information contained in the header, a set of random validators are selected to sign the new block, meaning that participants staking via PoS will vote to determine the validity of blocks generated by the PoW miner. Stakeholders who participate in PoS voting are called verifiers. Here, the chance of being selected as a verifier would depend on the number of tokens they hold. The more tokens a verifier has, the higher the probability of them being selected. (Transactions related with a Stake System are inserted into the UTXO set.)

Ticket Pool

To ensure that the voting process remains fair, the DCR mechanism is implemented for the ticket pool. All tickets are placed in the ticket pool. The system randomly selects the ticket and refunds the cost of a ticket once the candidate successfully casts a vote. Resulting block rewards are jointly acquired by PoW and PoS miners.

A fare adjustment is made every 288 blocks and the total number of votes in the ticket pool is capped at 40960.

Every 288 blocks, the total number of votes in the voting pool is adjusted to 40960.

Purchasing tickets

Only HC token holders can purchase tickets, further, they can only purchase them directly from their wallet. The total cost of a purchased ticket includes its price and fees. Here, fees are paid to PoW miners who will place new tickets into a newly mined block.

Memory pool

HC used to purchase a vote will be locked and cannot be withdrawn until that round of voting has passed. Purchased tickets first need to be consolidated and recorded in the block by miners to be selected as an immature ticket. If tickets are not consolidated, they remain in the mempool. The higher the voting fee paid by the participant, the higher chance the ticket will be selected by miners to enter the ticket pool. Since there is a limit to how many tickets a new block can record, this creates competition among ticket holders.

Immature tickets

A ticket becomes an immature ticket once the miners consolidate all tickets and record them on the blockchain. An immature ticket will not be selected while it's waiting to enter the ticket pool, additionally, ticket fees cannot be refunded at this time. Tickets will only remain in the memory pool until they become a verified ticket.

If there are too many tickets in the memory pool (mempool) that have not been consolidated by miners, after a certain period of time, the ticket price and fee will be refunded. This process is very similar to Bitcoin mining where, for example, a participant tries to conduct a transaction but the fee they paid was too low, thus not incentivising the miner enough to process it. During these situations, fees are refunded to the participant.

Votes

After the ticket enters the ticket pool, it will become a mature ticket waiting to be selected by the sytem to cast a vote and claim block reward. The selection process is subject to a Poisson distribution function. Put simply, the probability of being selected within 28 days is 50%, and the probability of being selected within 142 days is 99.5%. If not selected within 142 days, the ticket price will be refunded while the ticket fee is used for the miner's verification fee and is not refunded.

3) Once the selected verifiers have completed signing the block, it will form a complete block and be added to the blockchain. If some verifiers are not available to sign the block, they will be selected to sign the next block and a new set of verifiers will be selected until the current block receives an appropriate number of signatures. The transaction fee will then be distributed among the miner and the verifiers who participated in the signing.

4) After verification, there will be a process to indicate whether or not the previous regular transaction tree (containing coinbase and non-stake related transactions) is valid. The system will then switch back to PoW, inserting the PoS outcome into the new PoW block, and will set the flag (based on the verifying majority) for the validity of the previous block. If the previous block's regular transaction tree is verified, the network then inserts these transactions into the UTXO set, and restarts the whole process over again.

In general, the combination of a PoW + PoS is more like a lottery system; the verifiers purchase (lock) some HC in return for a ticket. The ticket gives holders/owners the opportunity to vote on whether to add a new block to the existing blockchain and selects whether or not the previous transaction tree (containing the coinbase and non-stake related transactions) is valid. In return, block rewards are distributed among PoS verifiers and PoW miners, and all participants will receive their purchased ticket amount back. If the validation fails, that is, the block cannot get enough votes, the block is discarded, and the transactions return and await to be included in another block.

3.4.3.2. PoS Decentralised Staking Pool

A potential problem with PoS is that token holders may place their assets in a centralised pool. The risk that stakeholders face when entrusting their assets to a centralised PoS staking pool is much greater than the risk that PoW miners face when they contribute their computing power to a pool. Loss of tokens through centralized pool staking is a risk that is difficult to avoid.

Another risk of a centralised PoS equity pool is that there is no guarantee that the administrator will distribute the staked tokens in accordance with the agreed rules.

In its new network, HCASH takes advantage of a decentralised PoS staking pool. Through the contract built into its underlying protocol, its network automatically distributes rewards to users after deducting pool fees. Under this model, the pool cannot obtain the user's HC tokens, nor can it obtain the user's staked assets. That is, rewards will be distributed among participants trustlessly.

Due to high administrative fees, this pool may not be suitable for small stakeholders. Thus, in this situation it may work in a stakeholders favour to send their tokens to a centralised pool that invests into the larger decentralised pool.

3.4.3.3 Token Distribution Ratio Under a PoW + PoS Hybrid Mechanism

The total supply of HyperCash will remain the same. In its first year, after the new block generation (on average one per minute), 6.4 HC will be produced. For every 12288 blocks, rewards will be reduced by 999/1000, this will last for 99 years with an average 2.5 minute block time. Considering the current state of the HCASH network and the security of its future public chain operations, its mining ratio (PoW / PoS) will be adjusted to 2:1, rather than 1:1 after its main chain (2.0) is launched. The distribution block reward ratio on the HyperCash main chain will be: 60% PoW, 30% PoS and 10% for the development team and to encourage community development (this 10% of the block reward will eventually equate to 5% of the total supply of HyperCash). The HCASH team will provide more information to its community relating to this through the releasing of other documents, and the updating of this yellow paper in the coming future.

3.5 Privacy Preserving Mechanisms

Ensuring a high level of privacy throughout the network has always been part of HCASH's original design vision.Throughout our development process, we proposed a solution to preserve the privacy of our users by means of implementing a post-quantum ring signature, ensuring confidential transactions, and through a zero-knowledge proof protocol.

Through advancements in technology, we are able to implement two-way encryption not only during asset transfers, but also in many other fields that require extremely high transaction privacy. Moreover, we can facilitate encrypted communication between blockchain and non-blockchain distributed ledgers (such as DAGs). For instance, the encrypted communication from a HCASH Client to a Byteball Client can be implemented.

3.5.1 Ring Confidential Transaction (RingCT)

Ring Confidential Transaction (RingCT) is the privacy-preserving algorithm. Nowadays, HCASH has proposed a more advanced RingCT (privacy preserving cryptographic technique), which is post-quantum secure, with the ability to achieve bidirectional encryption in the process of asset transfer and high demands on transactional privacy.

RingCT can provide user privacy in two aspects:

1) Anonymize the identity of the sender

2) Hide the transaction amount

A RingCT protocol consists of 3 technical parts, namely one-time linkable ring signature, homomorphic commitment scheme and range proof.

The one-time linkable ring signature allows spender to sign a transaction using his one-time secret key (corresponding to his one-time public key) and anonymize the transaction by picking other n-1 public keys from the blockchain. Anyone in the blockchain only knows that the transaction is generated by one of the n users (whose public keys are included in this transaction), yet no one can find out who this user is exactly (the probability of finding out the real sender is equal to a wild guess). HCASH use this technology to preserve the privacy of the sender. Since the sender is hidden, it also requires another way to detect if the sender has double-spent. The linkability property can help. In spite of the anonymity property, the linkable ring signature further allows anyone to detect if the sender has double-spent (by using the same secret key to generate two or more transactions). This property provides the necessary security to reject any transaction that is double-spent.

The second part is the homomorphic commitment scheme. Although linkable ring signature can anonymize the identity of the sender, the transaction amount is still opened to the public. In order to hide the transaction amount, a homomorphic commitment scheme is deployed. Similar to an encryption scheme, a commitment scheme allows one to hide a message. On the other hand, there is no decryption algorithm. In other words, only the person who makes the commitment can reveal the committed value by revealing the message and the randomness used in the commitment process. Anyone can then verify the correctness of the commitment. A commitment scheme is homomorphic if commitment of $m_1 \pm m_2$ can be obtained from commitments of m_1 and m_2 without knowing their values (with only the commitment of m_1 and m_2 known).

A transaction is consistent if the sum of input amounts equals to the sum of output amounts. To hide transaction amount while still allows a miner to verify transaction consistency, one could make use of a homomorphic commitment scheme. Specifically, all transaction amounts are stored inside a commitment. We assume the randomness used in the commitment scheme is transmitted from the sender to the receiver using a secure channel[1]. A miner can verify transaction consistency using the homomorphic property of the input and output commitments.

The remaining task of this part is to combine linkable ring signatures and homomorphic commitments to hide both sender address and transaction amount. The HCASH team managed to achieve it in an innovative way. Looking ahead, this is possible if a commitment to 0 is a valid public key of the linkable ring signature scheme whose private key can be derived from the randomness used in the commitment; while it is impossible to derive the private key of the commitment of any non-zero value.

Firstly, the linkable ring signature is extended to support a ring of the following form: $\{(pk_{11},pk_{12}), (pk_{21},pk_{22}), \ldots, (pk_{n1},pk_{n2})\}$ (to make it easier to understand, we assume that each stakeholder only has two wallets and therefore two public keys), where the signer needs to use both sk_{j1},sk_{j2} to sign, for some j belongs to $\{1,\ldots,n\}$.

[1] This can be done by appending the encryption of the randomness under the public key of the receiver to the transaction.

Secondly, we combine the homomorphic commitment scheme with the extended linkable ring signature scheme. To facilitate discussion, we assume each address is of the form (pk, cn), where cn is the homomorphic commitment of amount v of that address. Assume the real spender is (pk_π, cn_π), and he chooses n-1 other addresses and arranged them as (pk_1, cn_1), (pk_2, cn_2), …, ($pk_{\pi-1}$, $cn_{\pi-1}$), ($pk_{\pi+1}$, $cn_{\pi+1}$), (pk_n, cn_n). Assume the output is (pk_{out}, cn_{out}).

Thirdly, compute new "public keys" pk'_i = **eval**(cn_i, cn_{out}, -), where **eval** is the evaluation function of the homomorphic commitment scheme, and $\mathbf{pk_i}$ is the commitment of $\mathbf{v_i - v_{out}}$. Now, $\mathbf{pk_i}$ is a valid public key if $\mathbf{v_i - v_{out}}$ =0 thanks to the special property of the underlying homomorphic commitment scheme and the extended linkable ring signature scheme discussed above.

Finally, the spender computes the extended linkable ring signatures based on {(pk_{11}, pk_{12}, pk'_1), (pk_{21}, pk_{22}, pk'_2), …, (pk_{n1}, pk_{n2}, pk'_n)}. The rationale is that the spender knows the secret key of (pk_{i1}, pk_{i2}, pk'_i)because $\mathbf{pk'_p}$ is a commitment of 0 if the transaction is consistent. Otherwise, the spender will not be able to compute the secret key of $\mathbf{pk'_i}$ and thus unable to compute the required extended linkable ring signature.

The above design is vulnerable to integer overflow/underflow attack when the scheme is extended to support multiple inputs and outputs. Specifically, the above protocol only ensures the sum of committed input values equals to that of the committed output values. A malicious spender, for example, can create output commitments of 100 and -99, and a transaction is valid if he is using an input with value 1. Furthermore, in cryptography, we work in a finite field of known order (usually **a** prime **p**) and thus **-99** is regarded as **p-99**, a large value.

To prevent this type of attack, the spender also needs to provide a proof that each output commitment is within a specific range, say, between 0, 2^{64} -1. Therefore, the final part of a confidential transaction is range proof that a committed value is within a given range.

3.5.2 Quantum Resistant RingCT

The RingCT protocol does not provide quantum resistant security (post-quantum security). Therefore, in the existence of quantum computer, one can easily forge the signature and the system will be cracked down. In order to provide post-quantum security, we use lattice-based cryptography to construct the whole RingCT protocol (we call it Lattice RingCT). This designed adopted by HCASH team can enhance the security in the transaction and preserving the privacy in post-quantum era.

We adopted the scheme in the previous paragraphs and [7] as the foundation. It provides a construction of one-time linkable ring signature and homomorphic commitment scheme. Yet it can only support 1 wallet as input and one wallet to output (and thus does not require range proof). We first outline the details here:

Let $R_q = Z_q[x]/f(x)$, where $f(x)$ is a polynomial of degree n. The public parameter is $A_0' \in R_q^{1\times(m-1)}$ and is used to commit to a scalar message $m \subseteq Dom_m \subseteq R_q$, where Dom_m is a domain of that message, as follows: $Com_{A_0'}(m,\ S_0) = A_0' \cdot S_0 + m$. The properties of the homomorphic operations are also defined as:

$$\begin{aligned}
Com_{A_0'}(m_1, S_0)\ Com_{A_0'}(m_2, S_0') &\equiv Com_{A_0'}(m_1, S_0) + Com_{A_0'}(m_2, S_0')\ mod\ q \\
&= Com_{A_0'}(m_1 + m_2, S_0 + S_0') mod\ q \\
Com_{A_0'}(m_1, S_0) \ominus Com_{A_0'}(m_2, S_0') &\equiv Com_{A_0'}(m_1, S_0) - Com_{A_0'}(m_2, S_0')\ mod\ q \\
&= Com_{A_0'}(m_1 - m_2, S_0 - S_0') mod\ q
\end{aligned}$$

where $m_1, m_2 \in R_q$ are ring elements and $S_0, S_0' \in R_q^{(m-1)\times 1}$ are $(m-1)$-dimensional vectors of ring elements. The integers $m_1, m_2 \in Z$ are encoded in binary as coefficient vectors $m_1 = (m_{j,0},\ \dots,\ m_{j,l-1}, 0, \dots 0) \in \{0,1\}^n$ where $m_j = \sum_{i=0}^{l-1}(m_{j,i} \cdot 2^i)$ with $m_{j,i} \in \{0,1\}$ and $j \in \{0,1\}$.

Note: Our RingCT v1.0 follows from the linkable ring signature scheme construction from [7], where an additional function – called $L2RS.Lift$- is used to lift ring elements from $R_q^{1\times m}$ to $R_{2q}^{1\times m}$. In the following description of our RingCT we will take use of that function to lift some elements as required.

The construction of the Lattice RingCT v1.0 algorithm consists of the following five algorithms (*Setup, KeyGen, Mint, Spent, Verify*):

1) *PubParams ←Setup(λ)*: On input security parameter λ, this algorithm and outputs the public parameters, A_0' and H_0', where $H_0' = (h_{0,1}, \dots,\ h_{0,m-1}) \leftarrow R_q^{1\times(m-1)}$ is a vector of $m-1$ ring elements (Note, that the hash function is used to create the ring signature in step 4. See [7] for more details)

2) $(A_{in},\ S_{in})$ *←KeyGen (PubParams)*: On input public parameters, it outputs the input wallet (IW) pair of keys $(A_{in},\ S_{in})$, where $A_{in} \in R_q^{1\times m}$ is the public-key (or one-time address) and is the private-key. The commitment of zero is defined as $a'_{1(in)} = A_0' \cdot S_0\ mod\ q \in R_q = Com_{A_0'}(0, S_{0(in)})$.

3) (cn', ck')←$Mint(A_{in},\ a_{in})$: On input a valid one-time address A_{in} as well as an input amount $a_{in} \in \{0,1\}^n$ to create a coin cn_{in}', this algorithm chooses a coin-key $ck_{in}' \in Dom_{S_0}$ from the domain of S_0, where every component is chosen uniformly and independently with coefficients in $(-2^\gamma,\ 2^\gamma)$. It computes a commitment on the input amount a_{in} with randomness ck_{in}' as $cn_{in}' = Com_{A_0'}(a_{in},\ ck_{in}')$ and it returns (cn_{in}', ck_{in}') An account constitutes $(a'_{1(in)},\ cn_{in}') \in R_q \times R_q$.

4) $(TX,\ \sigma_{L'}(\mu)) \leftarrow Spend(\mu, OW)$: This algorithm consists of the following steps:

a. A new coin for the output wallet (OW) is created by the spender. It generates $ck_{out}' \in Dom_{S_0}$, where every component is chosen uniformly and independently with coefficients in $(-2^\gamma,\ 2^\gamma)$, then it is computed $cn_{out}' = Com_{A_0'}(a_{out}, ck_{out}')$, where a_{out} denotes the output amount. The new OW is set as $(a'_{1(out)},\ cn_{out}') \in R_q \times R_q$.

b. A transaction string $\mu \in \{0,1\}^*$ defines the ring signature message.

c. The list of the public keys of the underlying ring signature is constructed as $L' = \{(\hat{a}'_{1(in),i}, cn'_{in,i})\} \in R_q \times R_q$ for $1 \leq i \leq w$ with w being the size of the ring signature. Its components are produced as:

i. $\hat{a}'_{1(in),i} = a'_{1(in),i} + cn'_{in,i} - cn'_{out} = Com_{A'_0}(a_{in,i} - a_{out}\ ,\ S_{0(in),i} + ck'_{in,i} - ck'_{out})$

ii. $cn'_{in,i} = Com_{A'_0}(a_{in,i},\ ck'_{in,i})$.

d. We call the *L2RS.Lift* function from [7] to lift *L'* from $R_q^{1\times m}$ to $R_{2q}^{1\times m}$ to:

i. $L' = \{(L2RS.Lift(A'_0, \hat{a}'_{1(in),i}),\ L2RS.Lift(A'_0, cn'_{in,i}))\} = \{(\hat{A}_{1(in),i}, CN_{in,i})\} \in R_{2q}^{1\times m} \times R_{2q}^{1\times m}$

for $1 \leq i \leq w$.

ii. The private-key of user π is given as

$S''_{in,\pi} = (S_{in,\pi}, CK_{in,\pi}) \in R_q^{m\times 1} \times R_{2q}^{m\times 1}$, where:

- $S_{in,\pi} = (S_{0(in),\pi} + ck'_{in,\pi} - ck'_{out}) \in R_{2q}^{m\times 1}$
- $CK_{in,\pi} = (ck'_{in,\pi}\ , 1) \in R_{2q}^{m\times 1}$.

e. The algorithm creates the ring signature $\sigma_{L'}(\mu)$. (For more details see [7], Algorithm 5).

f. We set the transaction $TX = (\mu,\ L',\ OW)$.

g. This algorithm ultimately outputs TX and $\sigma_{L'}(\mu)$.

5) *(Accept/Reject)* ← *Verify*$(TX,\ \sigma_{L'}(\mu))$: This algorithm verifies the signature (according to Algorithm 6 in [7]) and returns either *Accept* or *Reject*.

This construction as stated supports one-IW to one-OW and thus in this case the range proof [8] is not needed.

The next step is to extend it to multiple input/output wallets. We are working towards this part in the lattice-based setting.

The last step is to give a lattice-based range proof for all committed values. There are several approaches to achieve this goal. The naive way is to construct a 1-out-of-2 ring signature to conduct the range proof. A more efficient way is to deploy the bullet-proof technique [9] and convert it into the lattice-based setting. We are investigating the best solution for this part.

One-time Public Key Construction

The one-time public key of the transaction receiver may be given to the sender by the receiver. However, this implies that the sender has to contact the receiver for each transaction and this may be undesirable. On the other hand, publishing the one-time public key may jeopardize receiver privacy. To cope with this problem, we are going to design a new kind of stealth address generation.

More concretely, the receiver publishes a set of long-term public keys. Each time a sender wishes to send a transaction to that receiver, he/she chooses a random subset of the set of long- term pubic keys and used them to generate a new pseudorandom public key. The randomness used in the choosing of the random subset is determined by a non-interactive key exchange between the sender and the receiver and thus does not need to be transmitted. Finally, the receiver can recover this randomness to re-compute the random subset, and computes the private keys of the pseudorandom public key. This hides the identity of the receiver without requiring additional communication between the sender and receiver.

3.5.3 Zero-Knowledge Proof

Zero-knowledge succinct non-interactive arguments of knowledge (zk-snark) , commonly referred to as zero-knowledge proof, is another useful technology in providing transaction privacy. Specifically, zk-snark allows a prover to produce a short proof to attest to the fact that the prover knows a certain secret without actually revealing the secret. Zcash, a cryptocurrency regarded as offering a very high level of privacy protection, is the first successful use case of zk-snark.

However, there are still a number of limitations in existing zk-snarks. This translates to the following issues in Zcash, which is utilizing zk-snark technolohy. While zk-snark is efficient in terms of proof size and verifier's computation, it is inefficient in terms of prover's computation and parameter size. Specifically, the public parameter is about 1 GB in Zcash, and that it takes minutes for a powerful computer to generate a proof. As a result, it is virtually impractical to implement a secure standalone mobile wallet for Zcash.

As mobile devices have become prevalent and mobile payment is widely accepted, there is a pressing need to address the above issues if zk-snark is to be used in cryptocurrency. We are making encouraging progress in this regard. We estimated that the efficiency improvement should be sufficient to allow payment in the mobile scenario. Below we give a brief account of our finding, whose details can be found in [10]. Efficiency improvements are based on the following improvements. Firstly, zk-snark requires representing the statements to be proved in the form of arithmetic gates. At a high level, these statements are related to the building blocks of the coins, including hash functions, commitment schemes and pseudorandom functions. In Zcash, the choice of this building blocks is the commonly used hash function, such as SHA256. However, the circuit of these building blocks consists of nearly a million Boolean gates and thus the parameter and prover's consists of millions of elements and steps.

First, we suggest that building blocks can be represented with much fewer arithmetic gates. Consequently, the prover's effort and the parameter size is much smaller.
The second improvement comes from a better architecture. Two highlights include the use of a 3-ary search tree instead of a binary search tree, and the use of window-based modular exponentiations of windows size = 3. While typically a binary search tree is simpler, the fact that in group equipped with a bilinear map, it is possible to represent operations involving 3 elements with less arithmetic gate, and thus the additional improvement.

Combining the two tricks, it is hopeful that our zk-snarks based cryptocurrency will have a parameter size reduced by approximately 90%, and thus achieving our goal of implementing the system with a mobile client.

3.5.4 Conclusion

The RingCT protocol created by HCASH is based on a lattice signature scheme (we call is lattice RingCT). A lattice RingCT implements post-quantum features, protecting clients' privacy and enhancing transaction security in a future quantum-computing environment. The improvement and optimisation of zk-snarks made by HCASH provides a theoretical and technical basis for secure mobile payments.

These innovative upgrades made by HCASH on the RingCT protocol and zk-snarks will provide clients who use our platform with more effective options for safeguarding their privacy in the future.

3.6 Dual Side chain comprised of Blockchain and DAG networks

As part of its core vision, the HCASH network contains a dual side chain comprised of both blockchain and DAG networks. Through this, we aim to create and implement technology that links various blockchains and enables interoperability, providing value to the ecosystem by allowing information to be freely transmitted across systems.

It's important to note that during initial stages of development, DAGs will not be part of our main focus. Before DAG becomes a primary focus, the development team will explore and research the future route of DAGs. We will continue to announce more results as we come to better understand this technology and where it will fit in our ecosystem.

With its unique technical characteristics and development potential, DAG plays an essential role in the future growth of the HCASH ecosystem. After initial research has been conducted, the HCASH development team will strive to develop distributed DAG based ledgers in a dual-chain ecosystem, providing users with a decentralised, distributed, and de-trusted tiered infrastructure network, that provides fair and open access to all users.

3.6.1 Potential Advantages of DAG Technology

A DAG (or Directed Acyclic Graph), is a directed graph that has no topological ordering. 'Directed' refers to the direction, which is exactly the same direction and in sequence. 'Acyclic' means that it is not a closed loop, and you cannot return to the starting point by travelling in a single direction. In the big data world, DAG is commonly used as the underlying structure in big data frameworks such as Hadoop, Storm, and Spark.

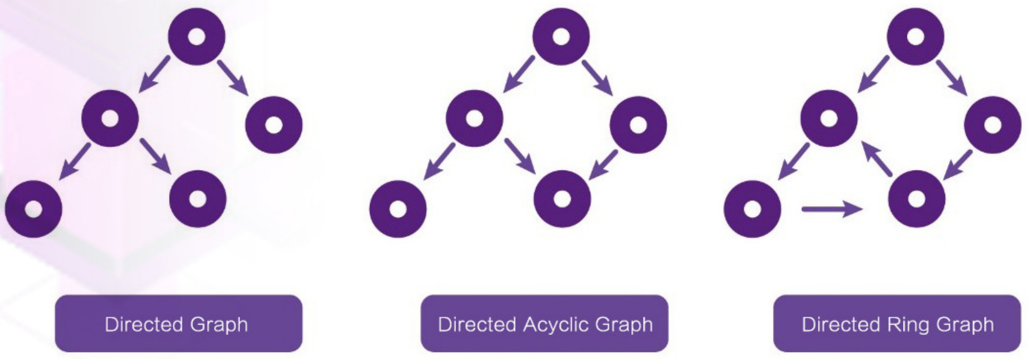


Figure 8: Directed Graph, Directed Acyclic Graph, Directed Ring Graph

A DAG does not involve any block concept. Transactions within a DAG network will form a DAG unit, and in a DAG network there is no need for miners to consolidate transactions. To participate in a transaction, you must verify previous transactions which are based on different rules. This kind of verification system means that DAGs can write many transactions asynchronously and concurrently. This ends up forming a tree structure of topology which can greatly improve network scalability.

The first thing people began to highlight when reviewing DAG networks was regarding blockchain scalability. Compared to blockchains, DAGs have considerable advantages in some aspects, including:

1) Faster transactions

Since DAGs abandon the block concept and transactions go directly into the whole network, transaction speeds are expected to be much faster than those on the blockchains that are based on PoW and PoS mechanisms.

2) Eliminates the necessity for miners

A DAG environment is such that users participate in reaching consensus by approving previous transactions. This eliminates the need for miners to agree on the transaction sequence, and then consolidate them into blocks. Since there are no mining roles in a DAG network, running the network turns into a much less energy intensive process.

3) Lower transaction fees

In a DAG network, there is no price competition as users aren't competing to verify transactions. This results primarily from the lack of mining incentive mechanisms that exist on traditional Blockchain networks such as Bitcoin and Ethereum. Because of this, transaction fees are extremely low. So low that networks such as these can facilitate small value high-frequency trading or IOT applications.

Considering that users in a DAG network validate previous transactions and that no competition occurs, this creates an environment that can sustain extremely low fees and facilitate high frequency trades. HCASH believes that this would best suit business applications, where blockchain and other vertical industries can intersect and create more transparency, immutability and independence. By creating a DAG side chain, this will allow HCASH to greatly enhance its main chain's scalability, while providing a solid foundation to introduce more useful and innovative features in its future expansion.

3.6.2 Inadequacies in Existing DAG Technology

Although DAG sidechains have their own distinct advantages, implementation is not easy. This is because current DAGs have their shortcomings, making them difficult to implement. These include:

1) Uncontrollable transaction times

The validation rule in a DAG network is such, that current transactions are validated by past transactions, and so the final transaction in a given period is unable to be validated. This is especially true in small projects with a smaller amount nodes in their initial development phase, making transaction times difficult to estimate. Current projects like Byteball use witnesses and other means to solve this problem, but this contradicts the idea of decentralisation to some extent.

2) Data inconsistency caused by Asynchronous Communication Mechanism

A typical blockchain uses a verification mechanism for synchronous operations, which ensures high consistency. As an asynchronous communication mechanism, a DAG improves scalability, but introduces the problem of inconsistency, lacking a globally synchronised mechanism. When running a smart contract on a DAG, it is very likely that the data stored between nodes will deviate after running for a period of time.

The first thing to pay attention to in blockchains is consistency; a distributed ledger needs to be homogenous, the transaction information needing to be kept consistent after each block is generated and recorded on the blockchain. The information contained in a new block must be consistent with the historical information, and the entire system must be designed to maintain consistency. On the other hand, a DAG network pays more attention to usability, ensuring that each user can complete transactions on this network. However, the DAG asynchronous communication mechanism could lead to the disorder of data unit generation, creating transaction conflicts.

3) Security Concerns

The application of DAGs to the field of distributed ledger technology has only happened in the past two to three years. As a novel implementation of the technology, it has not been verified or tested on a large scale, unlike the blockchain, which has been able to withstand the test of time over the past decade or so.

3.6.3 HCASH DAG development direction

In light of the shortcomings of a DAG, we find that, to achieve a safer, more optimised, and better performing solution, we must solve three major problems: First, the problem of resource isolation: once DAGs begin to be used for business applications, there will inevitably be large amounts of assets and data on-chain, especially when HCASH is linked with a large number of DAG ledgers. Isolating the resources of these different domains and functional side chains, whilst maintaining the interoperability of value within the ecosystem remains a sizeable challenge; The second issue is security: DAG technology itself is not new, but its application to the field of distributed ledger technology is still recent. Its security has not been tested, unlike its blockchain counterpart, and its security has not been verified on a large scale. The third issue is consistency. a DAG's asynchronous communication mechanism increases its scalability, giving it an advantage over a blockchain, but introduces the problem of synchronicity and consistency - a major drawback. This means, that when running a smart contract, it is very likely that the data stored between nodes will deviate after running for a period of time.

Based on the above considerations, the HCASH team plans to develop a number of features in its iteration of a DAG network, such as high concurrency, resource isolation, and more efficient consensus systems (explained below). Furthermore, the HCASH team will integrate its technical achievements relating to quantum security and cross-chain bridging to create a new generation of DAG distributed ledgers.

1) Enhancing high concurrency

As written previously, DAG networks have a high concurrency advantage over blockchain distributed ledgers, and in our iteration of DAG, we are researching and developing new technologies to further enhance this.

By selecting a reasonable SQL database and updating its existing SQL data, we have found ways to effectively improve the high-speed and high-concurrency capability of the network.

The execution plan in most related database systems is an interpreted language (a statement is executed in the SQL Server, and interpreted language execution plans are generated after binding, semantic analysis and cost-based optimization, etc.), while the engine will invoke the corresponding database function when executing the execution plan to run every operator. When the database is executing complicated interpreted languages, it can be time consuming and may reduce the efficiency of the CPU, effectively lowering the TPS. In response to this problem, we made two optimisations: the first is to select the corresponding in-memory database solution and design a high-performance database engine based on memory optimisation; the other is to improve concurrency by optimising the interpreted language. For example, avoiding fields that will cause a full scan. Other means to improve efficiency include data paging (database paging), reducing the number of interaction times, optimising business logic, etc.

Based on the plans outlined above, we will work towards the goal of greatly improving the concurrency of DAG systems.

2) Layers

Since each node in a typical blockchain network contains the data of the entire network, it is unable to meet the high-capacity storage and fast processing requirements needed for a scalable, high volume and high transaction environment.

The HCASH development team has hence decided to separate different resources on the underlying data structure of the DAG network by implementing a layer-structure, in order to improve the performance of the main chain. In our future DAG distributed ledger, a value exchange network will consist of layers with different features. These layers will be comprised of a basic consensus layer, a contract computing layer, a data storage layer, a cross-chain circulation layer and an application support layer.

The Basic Consensus Layer

Terminals can serve as nodes for the HCASH network to enable communication, application development and information circulation. HCASH provides its users with a decentralised and distributed network, with open and fair access to share and view information.

Any device connected to the HCASH network, even as small as a mobile terminal or an IoT device can join the ecosystem through a consensus protocol.

The Contract Computing Layer

An independent Node Network used in the HCASH system is comprised of decentralised nodes. Each Node is composed of HDVM Secure Virtual Machines and a HCASH DAG sidechain, extended to incorporate a smart contract interface.

The HDVM virtual machine is used for resource management isolation, encapsulating virtual machines API communication, etc.; the HCASH DAG side chain with its smart contract interface runs on the top layer and the HCASH DAG side chain with a Turing complete smart contract module is utilised for creating DAPPs. The core protocol was developed using Node.js, and the front end’s flexibility allows for the use of any language, with the front and back ends communicating through a JSON RPC protocol.

The Data Storage Layer

The HCASH side DAG stores data under the chain on a separate multi-centered/distributed storage layer:

a. Distributed unstructured storage - this method employs distributed IPFS projects for distributed storage. This solves data storage issues as each node is not required to store all the information. Only a few storage nodes, which are scattered between other nodes store all the information.

b. Distributed structured storage - this part of the HCASH DAG side-chain employs a global multi-centered/distributed database to provide rewards for active and storage nodes.

Anchoring of the data under the chain is performed by a Hash and Merkel Tree Interface API, which guarantees that checksum data cannot be modified.

The Cross-Chain Exchange Layer

The HCASH DAG side-chain facilitates information exchanges between a blockchain-based distributed ledger and DAG-based distributed ledger through an upgraded HyperExchange BMT protocol, which will be implemented soon.

The Application Support Layer

The HCASH side DAG will have an independent DAPP interface to provide convenient and fast DAPP development support for enterprise users.

3) Consensus and consistency issues

Existing DAG networks, such as IOTA and Byteball do not support smart contracts. This is because certain risks arise when using a DAG network——the originality and synchronicity of transactions cannot be guaranteed.

For instance, issues may arise when transaction times confirmed by a specific node (such as a remote node) cannot be estimated. A node in the network may not reflect the updated transaction information at a certain point in time (meaning that transaction information was not broadcasted to all the nodes). This poses a significant risk.

Trading units in DAG structures are called ‘normal trading units’, which contain signatures, transactions, and parent hash values. Due to uncertainties in implementing a DAG network, specifically regarding transaction confirmations, node synchronization, and transaction order inconsistencies (arising from network delays), this can delay a transaction confirmation. Each node is confirmed by subsequent nodes. However, the fact that subsequent nodes join the network in an inconsistent order make implementing smart contracts a difficult process.

Part of the HCASH design vision features the introduction of a multi-chain DAG structure, which makes basic participant units in the network independent. A verification chain equipped with a voting algorithm is introduced to solve the uncertainties mentioned above. Through this, smart contracts can be implemented. Moreover, smart contracts and transaction information remain consistent.

4) Cross-chain bridging

Early cross-chain technology implemented in blockchains such as Ripple and BTC Relay mainly focused on asset transactions. The second-generation of cross-chain technology, implemented in blockchains such as Polkadot and Cosmos mainly focused on cross-chain infrastructures. HyperExchange, our third-generation cross-chain technology has implemented a multi-asset cross-chain trading function, and Turing complete smart contract system which supports multi-asset transactions. Our development team will continue to conduct research and development in the field of cross-chain technology as per our roadmap, and any additional information discovered will be reflected in newer iterations of this yellow paper. We will continue in our pursuit to enable value to circulate between blockchain and non-blockchain (i.e. DAG) based distributed ledgers. (Refer to the HyperExchange cross-chain principle and process above)

5) Post-quantum

The HCASH development team have proposed a new communications protocol based on a segregated witness which relieves pressure on the network caused by long signatures. At the same time they've implemented a post-quantum signature scheme. In addition, HCASH have created a RingCT protocol based on a lattice signature scheme (which we'll call lattice RingCT). This too has quantum-resistant features. After further research and testing, these protocols may be applied to our DAG side-chain in the future to enable a high level of user security in a post-quantum environment.

3.7 The Decentralised Autonomous Organisation (DAO) Governance

The Decentralised Autonomous Organisation (DAO) is an unexpected, yet ideal product of the cryptography technological revolution. The root of the Decentralised Autonomous Corporation (DAC) can be traced back to the decentralised organisation described by Ori Brafman in "Starfish and Spider" (2007) [11], and "peer production" described by Yochai Benkler in “Web Fortune” (2006) [12]. However, these two concepts are later linked together by cryptocurrency-related technology, and gradually entered into cryptocurrency lexicon. In October 2013, Dan Larimer first put forward the idea of Decentralised Autonomous Corporation (DAC), a classification which he considers Bitcoin falls under.

To provide a clear definition of DAC, we summarise the seven features that are necessary to a DAC:

- *Openness:* The design of DAC system is made with a priority for transparency. The principle of openness and transparency is the cornerstone of the entire DAC system. An organisation that operates behind closed doors cannot be considered as a DAC. Nowadays, the spirit of open source software has become a typical example of openness;

- ***Decentralisation:*** No centralised individuals or organisations can control the entire DAC. This feature determines self-similarity. The decentralised characteristic of the system ensures the vitality of the DAC system and protects people from corruption and abuse of authority;
- ***Autonomy:*** Everyone can participate in the DAC system. All participants are either subsidiary or sub-unit of DAC system, which will promote the development of DAC from their own point of view. The spontaneous behavior of the participants guarantees that DAC is operational.
- ***Value:*** A DAC system must have use value and can be put into practical application. For example, several features of the Bitcoin system, such as international payment network, anonymous transaction, tax avoidance and value storage, have determined the profitability of the Bitcoin DAC system and contributed to improve the value and utility of the coin for coin holders;
- ***Profitability:*** DAC participants will receive rewards for contribution to DAC system development, and the profitability is determined by the value of the DAC itself.
- ***Self-similarity:*** Even there are only a few DAC nodes exist, the DAC system can still function and develop normally. The destruction of some unit nodes will not affect the development of DAC, which is guaranteed by the decentralization property.
- ***Democracy:*** Changes in the core protocol of the DAC system require voting from vast majority of units, and the decentralization and autonomy feature have determined that the DAC must be a system capable of democratic voting.

Vitalik later extended this idea of DAC and proposed a more general concept: DAO (Decentralised autonomous organisation). Unregulated crowd funding and service segregations are components of a DAO, as well as cryptography technological management and trust-base automation. However, this kind of autonomous organisation in ideal state can also lead to serious consequences if there is no strict control during the system design stage. In June 2016, the DAO,a large crowdfunding project in the history of Ethereum blockchain, raised more than $150 million USD worth of ETH. Nevertheless, due to the loopholes in code, the organisation was attacked and lost more than 3.6 million ETH, worth more $60 million USD at that time. Consequently, the ETH community split with the announcement of new security protocols, resulting in the co-existence of two blockchains: ETC and ETH.

In the HyperCash system, 5% of the coins will be sent to a DAO, and all HC holders can determine the use of funds through a real-time dynamic voting system, for example, development of wallets and other infrastructures, or carrying out marketing campaign and other public relation activities. The DAO is the driving force behind future advancement of HCASH community and ensuring the longevity of the community. At the same time, the code for HCASH DAO will go through rigorous audits and involve human intervention, if necessary, at the initial stages (the Foundation would invite a third party to conduct security audits on the code). This is to protect the DAO from making significant errors in the utilization of funds at early stage.

4. Technical Development Roadmap

After the release of the HCASH white paper, we have formed a development team including the world-renowned university blockchain joint lab, and actively pushed forward several development works according to the seven visions of the white paper. As our aim is to build new technical standards and redefining value in the blockchain domain, the challenges are also unprecedented.

In the past year, the joint efforts of the development team has resulted in varying degrees of progress in all aspects, especially the break through in the field of anti-quantum signature and interoperability between blockchains.

In a nutshell:

- Initial implementation in the value interoperability between blockchains;
- Initial implementation in quantum resistant signature;
- Innovatively combined quantum resistant technology with ring signature, which provided a feasible development plan to development quantum resistant ring signature;
- Improved the zero-knowledge proof, this would lay the foundation in building zero-knowledge proof based mobile payment.

As a new technology that has the potential to change the world, the blockchain has unlimited possibilities. Thus, HCASH's research on blockchain technology is also never ending. Based on the results we have achieved so far and our overall vision, we will accelerate on the development process in order to achieve our technical vision. Meanwhile, we will keep an eye on the developments of blockchain technology and actively be looking to implement them.

On the basis of the first phase of development, we have adjusted our plan for the next phase of development work. The next phase will be an important stage to fully realize the vision of HCASH technology, the upcoming technology development roadmap is as follows:

HyperCash

Accomplished features

- Quantum resistant signature
- PoW + PoS hybrid consensus
- Community voting module function ready

2018

July

- Improving HyperCash mainchain performance
- RC2 open beta

August

- Complete HyperCash mainchain performance improvement

September

- Start HSR/HC swap

2019

19 Q1

- Kick-off HAIL Protocol development

19 Q2

- Complete HAIL Protocol development

19 Q3

- Complete development for automatic deployment of community autonomous governing protocol function

19 Q4

- Complete layer and sharding technologies Achieve high TPS DAG side chain development

HyperExchange

2016

16 Q2

- Base layer platform construction
- Initiated smart contract development

2017

17 Q1

- Completed Virtual Machine architecture

2018

18 Q1

- Initiated cross-chain technology development
- Completed core functionality of cross-chain technology (BTC/LTC)
- Completed testing on cross-chain technology

18 Q3

- Complete cross-chain development testing main net
- Consolidate with HyperCash main net ecosystem
- Accomplish HyperExchange smart contract transation fee distribution function
- Complete optimisation of smart contract executing environment, accomplish lite node, lite storage

18 Q4

- Complete Integrated development environment / IDE
- Complete SDK (Software Development Kit)
- Complete Dapp development environment

2019

19 Q1

- Accomplish DAG cross-chain transaction

19 Q2

- Complete the development of high performance paralled computing functionality

19 Q3

- RPPOM consensus upgrade

5. Project Risk and Risk Management

NOTE: As noted elsewhere in these Terms, HCASH Token are not being designed to be sold as currency, securities or any other form of investment product. Accordingly, none of the information presented in this Exhibit A is intended to form the basis for any investment decision, and no specific recommendations are intended. The HCASH foundation expressly disclaims any and all responsibility for any direct or consequential loss or damage of any kind whatsoever arising directly or indirectly from: (i) reliance on any information contained in this Exhibit A, (ii) any error, omission or inaccuracy in any such information or (iii) any action resulting from such information.

By purchasing, holding and using HCASH Token, you expressly acknowledge and assume the following risks:

Risk of Losing Access to HCASH Token Due to Loss of Private Key(s), Custodial Error or Purchase Error

A private key, or a combination of private keys, is necessary to control and dispose of HCASH Token stored in your purchaser wallet or other digital wallet or vault. Accordingly, loss of requisite private key(s) associated with your Purchaser Wallet or other digital wallet or vault storing HCASH Token will result in loss of such HCASH Token. Moreover, any third party that gains access to such private key(s), including by gaining access to login credentials of your Purchaser Wallet or other digital wallet or vault service you use, may be able to misappropriate your HCASH Token. Any errors or malfunctions caused by or otherwise related to your Purchaser Wallet or other digital wallet or vault you choose to receive and store HCASH Token, including your own failure to properly maintain or use such Purchaser Wallet or other digital wallet or vault, may also result in the loss of your HCASH Token. Additionally, your failure to follow precisely the procedures set forth in Exhibit B for buying and receiving HCASH Token may result in the loss of your HCASH Token.

Risks Associated with the HCASH blockchain

Because HCASH Token and the Platform are based on the HCASH blockchain, any malfunction, breakdown or abandonment of the HCASH blockchain may have a material adverse effect on the platform or HCASH Token. Moreover, advances in cryptography, or technical advances such as the development of quantum computing, could present risks to HCASH Token and the platform, including the use of HCASH Token for token utility, by rendering ineffective the cryptographic consensus mechanism that underpins the HCASH blockchain.

Risk of Mining Attacks

As with other decentralised cryptographic tokens based on the Ethereum blockchain, HCASH Token are susceptible to attacks by miners in the course of validating HCASH Token transactions on the HCASH blockchain, including, but not limited, to double-spend attacks, majority mining power attacks, and selfish-mining attacks. Any successful attacks present a risk to the platform and HCASH Token, including, but not limited to, accurate execution and recording of transactions involving HCASH Token.

Risk of Hacking and Security Weaknesses

Hackers or other malicious groups or organisations may attempt to interfere with the Platform or HCASH Token in a variety of ways, including, but not limited to, malware attacks, denial of service attacks, consensus-based attacks, Sybil attacks, smurfing and spoofing. Furthermore, because the Platform is based on an open-source protocol, there is a risk that a third party or a member of the Foundation team may intentionally or unintentionally introduce weaknesses into the core infrastructure of the platform, which could negatively affect the platform and HCASH Token, including HCASH Token's use for token utility.

Risks Associated with Markets for HCASH Token

HCASH Token are intended to be used solely in connection with the platform, and foundation does not support or otherwise facilitate any secondary trading or external valuation of HCASH Token. This restricts the contemplated avenues for using HCASH Token, and could therefore create illiquidity risk with respect to HCASH Token you hold. Even if secondary trading of HCASH Token is facilitated by third party exchanges, such exchanges may be relatively new and subject to little or no regulatory oversight, making them more susceptible to market-related risks. Furthermore, to the extent that third parties do ascribe an external exchange value to HCASH Token (e.g., as denominated in a digital or fiat currency), such value may be extremely volatile and diminish to zero.

Risk of Uninsured Losses

Unlike bank accounts or accounts at some other financial institutions, HCASH token are uninsured unless you specifically obtain private insurance to insure them. Thus, in the event of loss or loss of utility value, there is no public insurer, such as the Federal Deposit Insurance Corporation, or private insurance arranged by company, to offer recourse to you.

Risks Associated with Uncertain Regulations and Enforcement Actions

The regulatory status of HCASH Token and distributed ledger technology is unclear or unsettled in many jurisdictions. It is difficult to predict how or whether regulatory agencies may apply existing regulation with respect to such technology and its applications. It is likewise difficult to predict how or whether legislatures or regulatory agencies may implement changes to law and regulation affecting distributed ledger technology and its applications, including the platform and HCASH Token. Regulatory actions could negatively impact the platform and HCASH Token in various ways, including, for purposes of illustration only, through a determination that the purchase, sale, delivery or use of HCASH Token constitutes unlawful activity, or that registration or licensing is required for HCASH Token or for some or all of the parties involved in the purchase, sale, delivery or use of HCASH Token. Foundation may cease operations in a jurisdiction in the event that regulatory actions, or changes to law or regulation, make it illegal to operate in such jurisdiction, or commercially undesirable to obtain the necessary regulatory approvals to operate in such jurisdiction.

Risks Arising from Taxation

The tax characterization of HCASH Token is uncertain. You must seek your own tax advice in connection with purchasing HCASH Token, which may result in adverse tax consequences to you, including withholding taxes, income taxes and tax reporting requirements.

Risks of Competing Protocols

It is possible that alternative platforms could be established that utilize the same open source code and protocol underlying the platform. The platform may compete with these alternative platforms, which could negatively impact the adoption of the Platform and HCASH Token, including HCASH Token's use for Token Utility.

Risk of Insufficient Interest in the Platform or Distributed Applications

It is possible that the Platform will not be used by a large number of individuals, companies and other entities or that there will be limited public interest in the creation and development of distributed protocols and decentralised applications. Such a lack of use or interest could negatively impact the development of the Platform and the potential utility of HCASH Token, including its use for Token Utility.

Risks Associated with Development of the Platform

Although the Platform will be deployed and operational at the time of the Token Sale, it is still subject to ongoing development and may undergo significant changes over time. How other participants will use the Platform is also outside of foundation's control. This could create the risk that HCASH Token or the Platform, as further developed and used, may not meet your expectations at the time of purchasing HCASH Token. It is also possible that the Platform will experience malfunctions or otherwise fail to be adequately developed over time, which may negatively impact the Platform and the potential utility of HCASH Token, including its use for Token Utility.

Risk of Dissolution of the Foundation

It is possible that, due to any number of reasons, including, but not limited to, an unfavorable fluctuation in the value of Ether (or other cryptographic and fiat currencies), decrease in HCASH Token's utility (including its use for Token Utility), the failure of commercial relationships, or intellectual property ownership challenges, the foundation may dissolve. The dissolution of foundation may still adversely impact the Platform and the utility of HCASH Token, given foundation's role in developing the Platform and its anticipated role in contributing to the ongoing development of the Platform.

Risks Arising from Lack of Governance Rights in Foundation

Because HCASH Token confer no governance rights of any kind with respect to foundation, all decisions involving the foundation will be made by foundation at its sole discretion, including, but not limited to, decisions to discontinue contributions to the platform's ongoing development or to sell or liquidate the foundation. As noted above, the consequences of those decisions could adversely impact the platform and the utility of HCASH Token that you hold, including HCASH Token's use for Token Utility.

Risks Associated with New and Evolving Laws Impacting Decentralised Application Technology

The distributed ledger and decentralissed application ecosystem, and by extension the Platform, may be subject to a variety of federal, state and international laws and regulations, including those with respect to financial services, consumer privacy, data protection, consumer protection, content regulation, network neutrality, cyber security, intellectual property (including copyright, patent, trademark and trade secret laws), and others. These laws and regulations, and the interpretation or application of these laws and regulations, could change. In addition, new laws or regulations affecting the Platform could be enacted, which could adversely impact the foundation, the Platform and HCASH Token, including HCASH Token's use for Token Utility.

Additionally, the users and developers of the Platform may be subject to industry-specific laws and regulations or licensing requirements. If any of these parties fails to comply with any of these licensing requirements or other applicable laws or regulations, or if such laws and regulations or licensing requirements become more stringent or are otherwise expanded, it could adversely impact the platform and HCASH Token, including HCASH token's use for token utility.

Specific Risks Relating of Value and Function of HCASH Token

The launch of new features on the platform utilizing HCASH Token may be delayed for reasons beyond foundation's control and may ultimately prove unsuccessful. The value of HCASH Token will depend on the token utility. The value may be affected by market conditions and other factors.

The ability to convert HCASH Token into other cryptocurrencies or fiat currencies will depend on the development of a trading market for the token. Foundation has no obligation to promote or support trading of HCASH Token.

No promises of future performance or value are or will be made with respect to HCASH Token, including no promise of inherent value, no promise of continuing payments, and no guarantee that HCASH Token will hold any particular value.

Unanticipated Risks

Cryptographic tokens such as HCASH Token are a new and untested technology. In addition to the risks included in this Section, there are other risks associated with your purchase, holding and use of HCASH Token, including those that the foundation cannot anticipate. Such risks may further materialize as unanticipated variations or combinations of the risks discussed in this Section.

6. References

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/ bitcoin.pdf. Oct 2008

[2] Vitalik Buterin. Ethereum White Paper : A Next-Generation Smart Contract and Decentralised Application Platform. https://github.com/ethereum/wiki/ wiki/White-Paper.

[ABL+17] Divesh Aggarwal, Gavin K Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. Quantum attacks on bitcoin, and how to protect against them. arXiv preprint *arXiv:1710.10377, 2017.*

[BBD09] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Postquantum cryptography.* Springer Science & Business Media, 2009.

[BDH11] Johannes Buchmann, Erik Dahmen, and Andreas Hu¨lsing. Xmss: a practical forward secure signature scheme based on minimal security assumptions. *Post-Quantum Cryptography*, pages 117–129, 2011.

[BHH+15] Daniel J Bernstein, Daira Hopwood, Andreas Hu¨lsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 368–397. Springer, 2015.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In Asiacrypt, volume 2248, pages 157–174. Springer, 2001.

[DDLL13] L´eo Ducas, Alain Durmus, Tancr`ede Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Advances in Cryptology– CRYPTO 2013, pages 40–56. Springer, 2013.

[DLL+17] L´eo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehl´e. Crystals-dilithium: Digital signatures from module lattices. Technical report, Cryptology ePrint Archive, Report 2017/633, 2017.

[dOL15] Ana Karina DS de Oliveira and Julio L´opez. An efficient software implementation of the hash-based signature scheme mss and its variants. In International Conference on Cryptology and Information Security in Latin America, pages 366–383. Springer, 2015.

[DOTV08] Erik Dahmen, Katsuyuki Okeya, Tsuyoshi Takagi, and Camille Vuillaume. Digital signatures out of second-preimage resistant hash functions. PQCrypto, 5299:109–123, 2008.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In ACNS, volume 5, pages 164–175. Springer, 2005.

[Duc14] L´eo Ducas. Accelerating bliss: the geometry of ternary polynomials. IACR Cryptology ePrint Archive, 2014:874, 2014.

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoit Gerard, and Mehdi Tibouchi. Side-channel attacks on bliss lattice-based signatures – exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. Cryptology ePrint Archive, Report 2017/505, 2017. https://eprint.iacr.org/2017/505.

[Flu17] Scott Fluhrer. Further analysis of a proposed hash-based signature standard. Technical report, Cryptology ePrint Archive, Report 2017/553, 2017.

[GLP12] Tim Gu¨neysu, Vadim Lyubashevsky, and Thomas P¨oppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In CHES, volume 7428, pages 530–547. Springer, 2012.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[Kat] Jonathan Katz. Analysis of a proposed hash-based signature standard, rev. 4.

[Kat16] Jonathan Katz. Analysis of a proposed hash-based signature standard. In Security Standardisation Research, pages 261–273. Springer, 2016.

[LM95] Frank T Leighton and Silvio Micali. Large provably fast and secure digital signature schemes based on secure hash functions, July 11 1995. US Patent 5,432,852.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In An- nual International Conference on the Theory and Applications of Cryptographic Techniques, pages 738–755. Springer, 2012.

[MBDG14] Carlos Aguilar Melchor, Xavier Boyen, Jean-Christophe Deneuville, and Philippe Gaborit. Sealing the leak on classical ntru signatures. In International Workshop on Post-Quantum Cryptography, pages 1–21. Springer, 2014.

[MCF17] Dr. David A. McGrew, Michael Curcio, and Scott Fluhrer. Hash-Based Signatures. Internet-Draft draft-mcgrew-hash-sigs-08, Internet Engineering Task Force, October 2017. Work in Progress.

[Mer89] Ralph C Merkle. A certified digital signature. In *Conference on the Theory and Application of Cryptology*, pages 218–238. Springer, 1989.

[MW17] Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 455–485, 2017.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[NSW05] Dalit Naor, Amir Shenhav, and Avishai Wool. One-time signatures revisited: *Have they become practical? IACR Cryptology ePrint Archive, 2005:442, 2005.*

[PBY17] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. *To bliss-b or not to be - attacking strongswan's implementation of post-quantum signatures. Cryptology ePrint Archive*, Report 2017/490, 2017. https://eprint.iacr.org/2017/490.

[PCG01] Jacques Patarin, Nicolas Courtois, and Louis Goubin. Quartz, 128-bit long digital signatures. In Cryptographers' Track at the RSA Conference, pages 282–297. Springer, 2001.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Annual Cryptology Conference, pages 80–97. Springer, 2010.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303– 332, 1999.

[Szy04] Michael Szydlo. Merkle tree traversal in log space and time. In *Eurocrypt*, volume 3027, pages 541–554. Springer, 2004.

[3] Bonneau, J., Clark, J., Felten, E., Kroll, J., Miller, A. and Narayanan, A. 2014. Mixcoin: Anonymity for Bitcoin with Accountable Mixes. In Financial Cryptography.

[4] Maxwell, G. 2013. Coinjoin: Bitcoin privacy for the real world. Bitcoin forum thread. https://bitcointalk.org/index.php?topic=279249.0.

[5] Chiesa, A., Garman, C., Miers, I., Virza, M., Ben-Sasson, E., Green, M., and Tromer, E.2014. Zerocash: Practical decentralised anonymous e-cash from bitcoin. In IEEE Symposium on Security and Privacy (S&P).

[6] Miers I., Garman C., Green M., and Rubin A. 2013. Zerocoin: Anonymous distributed e-cash from bitcoin. In IEEE Security and Privacy (S&P), pages 397{411, 2013.

[7] Torres, W. A. A., et al. Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1. 0). *Cryptography ePrint Archive Version.*

[8] Noether, S. (2015). Ring SIgnature Confidential Transactions for Monero. I*ACR Cryptology ePrint Archive, 2015, 1098*.

[9] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2017) Bulletproofs: Short Proofs for Confidential Transactions and More. *IACR Cryptology ePrint Archive, 2017, 1066*.

[10] Kluczniak, K. & Au, MH. (2018) Fine-Tuning Decentralised Anonymous Payment Systems based on Arguments for Arithmetic Circuit Satisfiability. *IACR Cryptology ePrint Archive, 2018, 176*.

[11]HOFFSTEIN J, PIPHER J, SILVER MAN J H. NTRU: A ring - based public key cryptosystem.

[12]LYUBASHEVSHY V, PEIKERT C, REGEV O. On ideal lattice and learning with errors over rings.