Crypto.com Chain: The next generation decentralized mobile payment protocol

{chain,contribute}@crypto.com

August 3, 2020
v0.6.0

Abstract

The development of blockchain technology and cryptocurrencies represents a cryptography and security breakthrough as significant as that created by the internet in the 1990s. This technology, however, is still at a very nascent stage; in order to generate mass adoption, it will therefore be necessary to find compelling real-life use cases that can appeal to an audience larger than the small group of industry professionals and experts. We believe that enabling cryptocurrency spending in the real world will constitute an adoption catalyst. Current traditional payment institutions and existing blockchains have failed to provide a secure, scalable and decentralized solution to support cryptocurrency payment.

Accordingly, we propose Crypto.com Chain, the next generation decentralized mobile payment protocol, the most efficient and secure way to pay and be paid in crypto, anywhere, any crypto at little to no cost. Crypto.com Chain will deliver on its vision by developing innovative technology components and processes (inc. scalable encryption algorithm to protect users' privacy, utilizing trusted execution environments, sustainable price stability mechanisms, user protection via PoGSD) catered specifically to cryptocurrency payment, while leveraging proven blockchain technology structural design elements.

Table of Contents

1. Introduction to Crypto.com Chain

Current traditional payment infrastructure and existing blockchain-powered payment networks have failed to provide a wide-spread, easy-to-integrate and fast settlement of cryptocurrency in the real world.

The key limitations of existing payment network infrastructures are that they:

1. Do not integrate with cryptocurrencies systematically;
2. Do not allow a way for either customers or merchants to reconcile payment figures in a trustless way;
3. Are vulnerable due to being the central point of failure;
4. Are expensive to operate;
5. Give low limits on cryptocurrency spending by default;

Moreover, the key limitations of existing blockchain-powered payment networks are that they are:

1. Too complex to setup and use;
2. Not friendly to crypto first timers;
3. Rarely supported beyond their own blockchain;

Our vision is to accelerate the world's development, adoption of and transition to cryptocurrency. Crypto.com Chain is the best way to pay and be paid in crypto, anywhere, anytime and using any crypto — at little to no cost.

2. Design Axioms

Crypto.com Chain, the next generation decentralized mobile payment protocol, will be designed based on the following foundational Design Axioms (DA_n), listed in order of priority ($DA_i > DA_{i+1}$):

DA_1: Secure
- Protection from fraud;
- Highly compliant.

DA_2: Highly Scalable & Fast
- Peak performance on par with centralized payment providers;
- Fast confirmation; targeting < 1 second through different means (e.g. P2P payment channels).
- High transactions per second (TPS); targeting 50,000 TPS, through different means (e.g. P2P payment channels);

DA_3: Augmented Decentralization
- Self-managed settlement;
- Phased validator node set evolution;
- Automated treasury rewards and sequencing.

DA_4: Upgradable and Fast in Innovation
- Flexible process for chain upgrades;
- Low dependency on other networks.

DA_5: Data Privacy Protection
- Encrypted on-chain pseudonymous transaction data, only relevant parties involved in each transaction can decrypt it;
- Efficient transaction validation.

DA_6: Inclusive
- Seamless integration of developers or new users with low technical barriers, appropriate incentives and strict penalties.

Decentralized ledger technology, such as blockchain, provides key built-in benefits that are aligned with Crypto.com Chain Design Axioms:

- It handles double spend naturally;
- It makes reconciliation easier (it even 'removes' the need for reconciliation, as long as the blockchain is properly structured);
- It facilitates open collaboration;
- It is more inclusive; anyone can join the network;
- It lowers the likelihood of the central point of failure.

3. Architecture

Overview

Building a blockchain is not just about software/hardware development. Rather, it is a combination of technological design, incentive mechanism, game theory and governance, which together nourish a robust system that also allows for continuous innovation. Our initially proposed architecture may hence undergo future revisions in response to changes in incentives, governance or other external requirements.

Crypto.com Chain is open to the public to join, participate and scrutinise related transactions. We do not expect that, for example, mobile clients will be able to perform heavy-lifting tasks and have a reliable always-online network connection. For that reason and DA_2, in the initial prototype of Crypto.com chain, there are two different types of nodes are responsible for various duties:

1. Council Node (Validator), responsible for validating and committing new blocks to the blockchain;

2. Community Node (Full node), responsible for fetching the blockchain data and serve it upon the client's request.

Trusted Execution Environments

In order to address DA_1, DA_2, DA_3, and DA_5, the core functionality of Crypto.com Chain Nodes is designed to run in secure enclaves of Trusted Execution Environments (TEEs). TEEs, such as Intel SGX, Arm TrustZone, or Keystone[1], are extended CPU instruction sets that isolate code executed in an enclave from the host operating system in hardware-encrypted RAM. TEEs ensure that even the node administrator cannot see private data that enclave code works with. Note that enclave code must still follow secure coding practices in order to avoid leaks through memory access patterns etc.

[1] An ongoing open-source project for RISC-V: https://keystone-enclave.org

Specifically, Crypto.com Chain leverages TEEs for three main reasons:

1. Flexibility in terms of what computation can be done and how the data schema can evolve. As requirements for Crypto.com Chain change, it is important that the existing code and data remain forwards-compatible.

2. Auditability with potentially fine-grained access control mechanisms: in the initial implementation, it is a separation of the permission to spend and the permission to view transaction data, but it could be more flexible and fine-grained (e.g. permission to view certain parts of transaction data).

3. Performance due to a low overhead: unlike, for example, fully homomorphic encryption in software, the overhead of executing computation in TEE should be minimal and the main cryptographic primitive is symmetric encryption which can be accelerated by dedicated CPU instructions, such as AES-NI.

An important feature of TEEs is their local and remote attestation. This feature enables nodes or external parties to verify that the code they plan to interact with is indeed the certified Crypto.com Chain code. In case of remote attestation, each node completes this step before establishing secure communication channels with other nodes.

In Crypto.com Chain design, TEEs can find several compelling use cases:

1. Sealing ledger data: While all transaction data can be distributed to any node for processing, humans (even node administrators) cannot view these data in raw form.

2. "Virtual" hardware wallets: Nodes can utilize Ledger Trustlet[2]-like software to protect their private keys.

3. Payment protocol enhancements: TEEs have gained popularity in blockchain systems research, as they can offer high transaction throughputs with low latency.

[2] https://github.com/LedgerHQ/bolos-tee

4. Witnessing external data: For data from oracles or other blockchain networks, TEEs can be used to attest data authenticity.

In the light of Foreshadow[3] and LVI[4] attack, Crypto.com Chain will not rely solely on TEEs for achieving DA_1 and DA_5, we will also consider other measures, including the following:

1. CRO collaterals;
2. Additional cryptographic measures for maintaining privacy;
3. Writing core parts in Rust[5], a programming language that ensures memory safety and freedom of data races.

Consensus

Council Nodes run a Byzantine Fault Tolerant (BFT) consensus protocol among themselves which resolves the final order of transaction sequences. The initial prototype will utilize the Tendermint Core[6] consensus engine. Tendermint works well for PoS / PoA networks, allows high transaction throughputs, and provides instant transaction finality on block commitment, which aligns well with DA_2. It was chosen as the consensus engine for the Chain prototype due to the following additional reasons:

- Backed by formal research[7];
- Robustly tested implementation[8];
- Track record of adoption[9];
- Modular architecture.

Moreover, to facilitate a high availability, the total number of Council Nodes in different locations will be required to be greater than a minimum set that is based on real performance tests.

[3] https://foreshadowattack.eu
[4] https://lviattack.eu/
[5] https://edp.fortanix.com
[6] https://tendermint.com
[7] https://eprint.iacr.org/2018/574.pdf
[8] http://jepsen.io/analyses/tendermint-0-10-2
[9] https://forum.cosmos.network/t/list-of-projects-in-cosmos-tendermint-ecosystem/243

Governance

Council Nodes are responsible for the governance of the network. The Crypto.com Chain Entity will propose software upgrades for approval by Council Nodes. Following a software upgrade approval and release, any nodes on the network that fail to upgrade after a specified grace period will be considered as having dropped out of the network voluntarily.

Security

As it is a public network, security (DA_1) and robustness are critical requirements. Threat modeling is a systematic approach to find potential threats by decomposing and enumerating system components. There are many different methodologies and/or frameworks when conducting threat modeling, such as STRIDE, DREAD, Attack Tree, etc. In our case, our Threat Model is based on STRIDE and Attack Tree.

STRIDE provides a set of security threats in six categories:

1. Spoofing: Impersonating the identity of another
2. Tampering: Data is changed by an attacker
3. Repudiation: An attacker refuses to confirm an action was conducted
4. Information Disclosure: Exposing sensitive information
5. Denial of Service: Degrade the availability or performance of the system
6. Elevation of Privilege

For each category, we enumerate all the potential threats by breaking down a high-level goal into more specific sub-goals, in a way similar to attack tree enumeration. And in each sub-goal, we set the risk level by combining the Severity and Exploitability of the item.

Severity

- 5: Severe impact on the whole system
- 4: High impact on the whole system
- 3: Moderate impact on the whole system or Severe impact on individual user/node
- 2: High impact on individual user/node
- 1: Moderate impact on individual user/node

Exploitability

- 5: Existing exploit code available
- 4: Relatively easy to exploit
- 3: Attack is practical but not easy, a successful attack may require some special conditions
- 2: Theoretically possible, but difficult in practice
- 1: Very difficult to exploit
- 0.1: Almost impossible

Assets

- The integrity of the account balance: the most important piece of information in the blockchain.
- Validator secret keys (block-signing keys of Council Nodes): one of the most powerful keys, losing 1/3 of these keys will render the whole system to an unstable state.
- User secret keys: key owner implies fund owner
- Transaction encryption keys: transaction privacy of the system relies on the secrecy of this key

Scope

The whole Crypto.com Chain is a complex system and involves many different components. And therefore, the scope of this threat model is limited only to the major components of the system. To be more specific, the threat modeling of Tendermint and Intel SGX is not in the scope of this threat modeling.

We also assume standard security measures such as OS level hardening, software patching, anti-virus, network firewalls, physical security etc. are properly implemented, executed and monitored. These mitigation strategies are not mentioned here.

Threat Model

The initial threat model can be found here.

Privacy

It is not only Crypto.com's belief, but many data privacy-related regulations in their essence mandate that it is your basic human right to control your money, data and identity. Data confidentiality is one of the aspects needed for this right to be fulfilled. As TEEs are used, code and data are isolated from the operating system, and provides a way for the code to "attest" its authenticity and integrity to

remote parties; the data inside secure enclaves is protected; even the node administrators cannot directly view raw transaction data on their nodes.

To further enhance privacy capabilities (addressing DA_1 and DA_5), Crypto.com Chain will include other software-based measures in case of secure enclave breaches. The initial prototype will utilize tree signatures[10] for threshold multi-signatures which provide a good trade-off between privacy and accountability. Furthermore, we will potentially explore employing other techniques, such as additively homomorphic commitments (as used, for example, in Confidential Transactions[11]), where data remains private even in the case of secure enclave breaches, and its processed parts can be securely and verifiably exposed for third-party auditing.

[10] https://blockstream.com/2015/08/24/treesignatures/

[11] https://elementsproject.org/features/confidential-transactions

4. Block Structure & Incentives

The details described in this section, as with other technical aspects of the Crypto.com Chain, are subject to revision. The described data structures only highlight a subset of end-user metadata that will be exchanged in protocol messages among relevant nodes. The exact details, handling etc. will depend on interactions with the underlying consensus layer, privacy and security mechanisms.

Hybrid accounting model

The native token used in Crypto.com Chain serves two main purposes:

1. High volume regular payments/value transfers transactions which data confidentiality is desired, and

2. Network operation, such as staking related transactions and events that are designed to be publicly visible.

To facilitate this, Crypto.com Chain uses a hybrid accounting system which combines the advantages from both Unspent Transaction Outputs (UTXOs) and account-based model for its accounting model, inspired by the work on chimeric ledgers. Specifically, there are two different address types, namely the "*Transfer Address*" and "*Staking Address*" that handle different kinds of accounting models and transactions. These differences are highlighted in the table below:

Transaction Type	Transaction Volume	Visibility	Accounting model	Address Type
Payments	High	Minimal: Confidentiality is desired	UTXOs (Bitcoin-like)	*Transfer address*
Network operation	Low	Maximal: Transparency is desired	Account Based (Ethereum-like)	*Staking address*

Textual Address Format

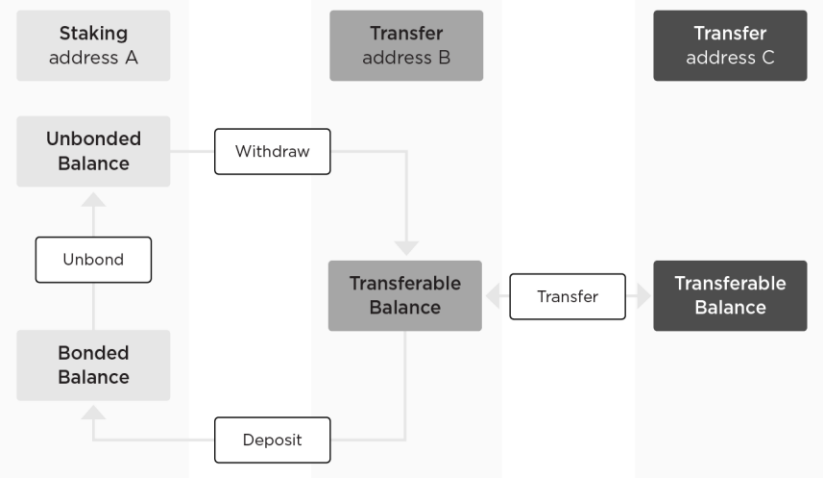
For backwards-compatibility with the existing contract on Ethereum[12], the "*Staking Addresses*" would preserve the hexadecimal textual representation. In which, it follows the format of the 20 bytes Ethereum account address.

On the other hand, the *"Transfer address"* for payments / value transfers utilize a new Bech32-based[13] textual address representation where the human-readable prefix would denote the network (i.e. mainnet, testnet, regnet) transactions are meant for.

Transaction Types and Flows

For transfer addresses, the balance of the address simply refers to the available and transferable balance (UTXOs) of the address. For account-based staking addresses, each address has its own "state", it contains: Bonded amount, Unbonded amount and some other metadata, such as account nonce and slashing related information.

To move funds around, there are four basic transaction types: Deposit, Unbond, Withdraw and Transfer. The following diagram summarizes how different types of addresses and transactions interact with each other:



Additionally, there are some advanced transaction types such as node-join and unjail, of which more details can be found here.

[12] https://etherscan.io/address/0xa0b73e1ff0b80914ab6fe0444e65848c4c34450b
[13] https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki

Payment Transaction Data Structure

As mentioned in the last section, for common value transfer, the accounting model in the initial prototype of Crypto.com Chain prototype will follow the UTXO model similar to Bitcoin, except that Chain's transaction output locking will be more restrictive (addressing DA_1 and DA_2). Furthermore, in a broader sense, a hybrid accounting model will be adopted to utilize staking-related functionality. (see technical documentation for details). The committed transaction included in a block will include at least these parts:

A. Raw transaction data encrypted against a verifiable shared secret of Council Nodes;
B. Hash of the raw transaction data.

Moreover, depending on the implementation, it may additionally include references to past data that needed to be fetched for transaction validation purposes.
Part A) Raw transaction data will never be revealed directly, as discussed in Privacy. The following encrypted data may contain additional obfuscation to prevent data leaks in case of TEE access policy breaches:

1. Transaction data:
    a. Transaction inputs;
    b. Transaction outputs.
2. An access policy of what can be exposed to whom, and under what circumstances, from the raw transaction data (enforced by TEEs). This access policy will refer to one-time keys related to:
    a. The customer's wallet,
    b. The merchant's wallet,
    c. The optional escrow service provider (PoGSD),
3. Transaction metadata: this includes versioning information, network identifier, and metadata related to the use of external currencies.
4. Collective witness, including signatures on the transaction's ID, against each transaction input.

To communicate transaction data across the network to related parties, interim state transactions will contain some of the above-mentioned data.

In order to address DA_6, all transaction data will be serialized in a backwards and forwards-compatible way, using a well-established binary format (the initial prototype will utilize the Simple Concatenated Aggregate Little-Endian binary codec defined in Section B.1 of Polkadot RE Protocol Specification); this will help to ensure

the ease and consistency of implementations across different programming languages used in third party integrations.

Genesis file structure

The genesis file defines the initial state of the Crypto.com chain. On top of the standard tendermint genesis format, we customize our own genesis file and facilitate the special features of the Crypto.com chain, for example:

- `council_nodes` defines the initial council node set with the key packages that other nodes can use for key agreements.
- `network_params` includes the transaction fee policy as well as the staking requirements, reward/slashing configuration for council nodes.

Sample genesis file of Crypto.com chain can be found here.

Block Data Structure

Crypto.com Chain will utilize Tendermint Core as its consensus engine, the block structure will follow the descriptions provided as described in Tendermint's documentation.

Furthermore, Crypto.com Chain will employ these additional conventions:

1. AppHash consists of a root of an authenticated data structure, such as a Merkle tree[14], constructed after committing a set of valid transactions in a given block and other parts, denoting the application state. Given an AppHash portion, a transaction ID and a Merkle proof, one can verify whether a transaction corresponding to a given ID was included in a block;
2. Each block will be tagged with a fixed sized probabilistic data structure, such as a Bloom filter[15], that will encode participants from all transactions in a given block.
3. The last two characters of ChainID will be assumed to be hexadecimal digits. These encode a single byte that should be included in every transaction's metadata. This value will vary for different network deployments, such as tests and main networks.

[14] Merkle, R. C. (1988). "A Digital Signature Based on a Conventional Encryption Function". *Advances in Cryptology — CRYPTO '87*. Lecture Notes in Computer Science. 293. pp. 369–378.
[15] Bloom, Burton H. (1970), "Space/Time Trade-offs in Hash Coding with Allowable Errors", *Communications of the ACM*, 13 (7): 422–426,

Transaction Fees

In the initial prototype, the purpose of transaction fees is an anti-spam measure, which is aimed to prevent valid transactions being broadcasted indefinitely. Generally speaking: If the transaction type allows indefinite valid transactions in an immediate time span, a fee must be paid; Otherwise, there will be no fee if the transaction type allows a limited number of valid transactions.

The network will initially have the following minimal linear transaction fee scheme:

$$A + B * (\textit{transaction size in bytes}),$$

where A and B are constants (fractions of the native currency). These fees are paid to the rewards pool.

In the long term, we will investigate the possibility of dynamic or zero fee schemes as long as these schemes preserve the payment data confidentiality and do not open the network to spam.

Council Nodes: Reward & Punishment

Crypto.com Chain is based on Tendermint Core's consensus engine, it relies on a set of Council Node (validators) to participate in the proof of stake (PoS) consensus protocol, and they are responsible for committing new blocks to the blockchain. Specifically, to participate in this consensus, validators will have to stake a minimum amount in their staking address. Afterwards, they can sign and submit a "Node-join" transaction and join the network once they are ready.

Reward
To incentivise the Council nodes to run the network, rewards are accumulated and distributed to the council nodes. There are three sources for the rewards:

1. Fixed total supply from a portion of the Secondary Distribution & Launch Incentives Pool and subsequently the Network Long Term Incentive Pool;
2. Transaction fees in the last epoch;
3. Slashing amounts from the byzantine or non-live nodes (if any).

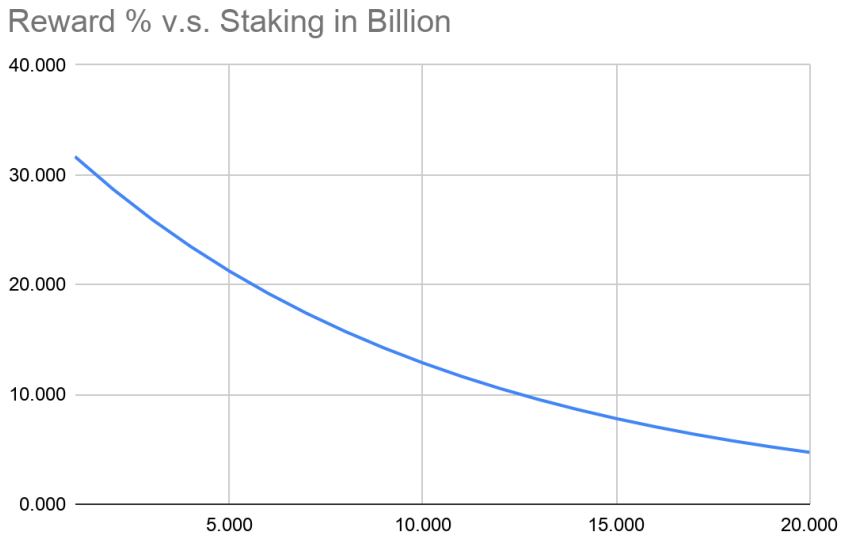
The reward scheme should be designed in a way that reacts dynamically to the actual network conditions, in particular, it should provide a greater incentive for the council node to stake the when the total staking is relatively low, and vice versa.

In response to this, the reward is depending on two major factors: i) the total staking, and ii) the length of time since the genesis block. The reward rate per annum, R, can be expressed by the following function:

$$R(s,t) = R_0\, e^{-\frac{s}{\tau(t)}},$$

where s is the total amount of tokens staked by validators to participate in the consensus process; R_0 is the upper bound for the reward rate, and Tau is a monotonically decreasing constant that controls the exponential rate.

For example, if we fix Tau=10 Billion, R_0=35%, the following graph shows the relation between the reward rate (per annum) and the total staking (in Billion).



Rewards will be periodically distributed to the Council Nodes based on their proven operation, i.e. based on the blocks they co-signed, as long as their corresponding accounts were not deactivated nor frozen due to misbehavior. In the consensus round, the chance of being a proposer is directly proportional to their *voting power* at that time, which, in general, is equal to the bonded amount (rounded to the whole unit) in the associated staking address of the council node. Some concrete examples on the reward and its distribution can be found in the technical documentation.

Furthermore, detailed data structures in the reward module of the reward mechanism can be found here.

Punishment

It is important that the council node maintains excellent availability and network connectivity to perform their tasks. Validators should be penalised if they fail to achieve these goals.

Punishments for a validator are triggered when they either make a byzantine fault or become non-live:

- *Liveness Faults* (Low availability)
  A validator is said to be non-live when they fail to sign a certain number of blocks within a given threshold[16];

- *Byzantine Faults* (Double signing)
  A validator is said to make a byzantine fault when they sign conflicting messages/blocks at the same height and round.

The penalties include losing a portion of their stake, losing their ability to vote, collect rewards etc. The detailed data structures in the punishment module of the slashing mechanism can be found here.

Signature Schemes

Besides the standard Ethereum signature for backward compatibility, Crypto.com Chain implemented the threshold Multisig[17] for multi-signature related features. Specifically, a threshold multi-signature address is formed by the public keys from different parties. In order to spend the funds in the multi-signature address, it requires a minimum number of keys to authorize the transaction. These are often referred to as M-of-N transactions and address, where

- M is the minimum signatures required to spend the funds from the multi-signature address;
- N is the total number of keys involved;
- N has to be greater or equal to M.

[16] See punishment-related parameters in the appendix

[17] https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki#multisignatures-and-threshold-signatures

Some of the actual use cases of multi-signature are covered in these application examples.

Example: Proof of Goods & Services Delivered (PoGSD)

Here we demonstrate a particular use case - “Proof of Goods & Services Delivered” (PoGSD) that provides third-party escrow service in a typical online checkout by utilizing the multi-signature wallets in the Crypto.com Chain protocol:

1. Buyer selects goods, ticks “proof-of-goods-and-services”;
2. Buyer retrieves invoice info from merchant's web;
3. The total amount and public keys retrieved from the merchant and escrow through APIs.
4. The buyer sends the total amount to the 2-of-3 multi-signature address generated based on the public keys from the parties involved.

Note that, under the multi-signature scheme, at least two of the three signatures are required to "unlock" and spend the funds in the address. Buyer and merchant's own benefits in any of the following scenarios:

Scenario A: The item is shipped:

A1) Normal
If the buyer confirms and accepts the delivered item, he/she can complete the purchase by co-signing the transaction to the merchant with his/her signature.

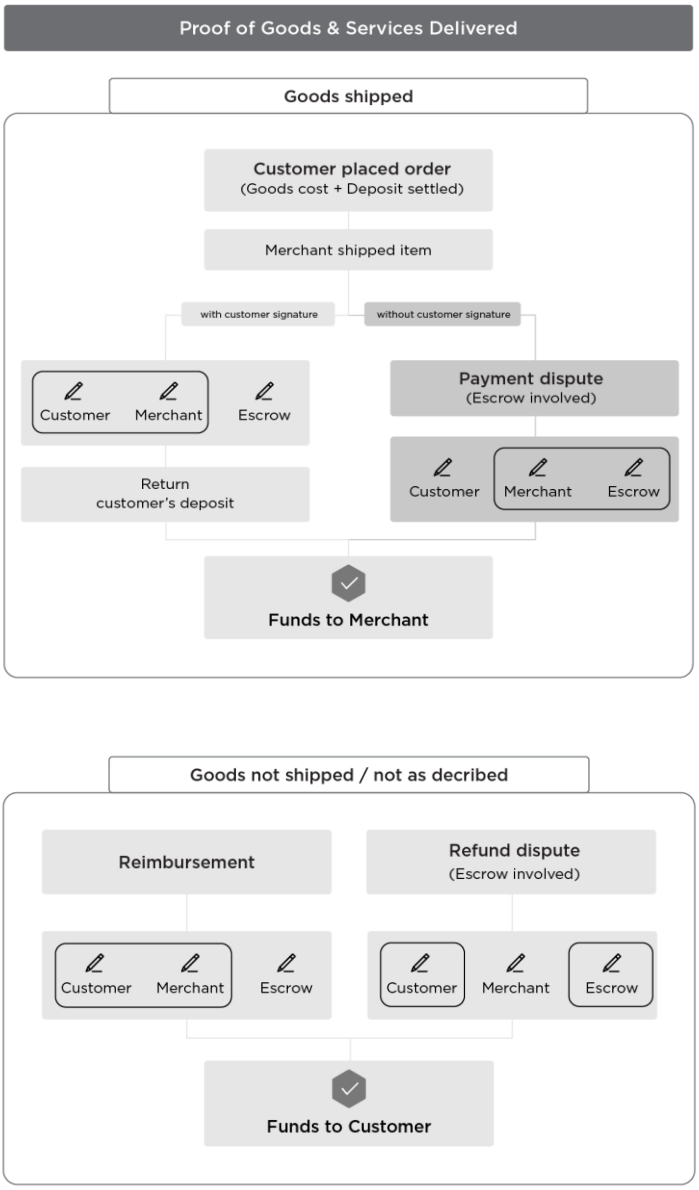
A2) Payment dispute (escrow involved)
If the merchant has not received the payment after a certain period of time, he/she can contact and provide evidence of delivery to the escrow. Once it has been confirmed, the transaction will be co-signed by the escrow and the funds will be released to the merchant.

Scenario B: The item is not shipped/not as described:

B1) Reimbursement (without escrow involvement)
The buyer can request a refund when the merchant fails to fulfil the order. If the merchant accepts the request, he/she can co-sign the transaction and refunds the buyer.

B2) Refund dispute (escrow involved)

In case the merchant neither agrees with the refund claim nor responds to the buyer's refund request, the buyer can reach the escrow to resolve the issue. If the resolution outcome is in favour of the buyer, escrow will issue a refund to the buyer by providing the co-signature.



5. Resilience & Agility

Network Redundancies

To ensure the robustness of the network, a minimum number of Council Nodes spread across the globe will need to be up and running. The minimum number will be decided based on real performance tests to balance the following factors:

1. robustness against compromising a supermajority of Council Nodes;
2. efficiency/high-performance.

Network Scalability & Performance

Crypto.com Chain aims to be a distributed network that is able to handle high transaction throughputs and low latency. Scalability and performance are hot research topics in the blockchain space. While adopting TEEs in the infrastructure may achieve a performant network, we will also explore other advances in the field, including sharding, consensus protocol improvements, transport network enhancements etc.

Transaction signatures will employ both ECDSA (for backwards compatibility) and a variant of the Schnorr signature scheme[18]. The Schnorr signature scheme has been recently proposed for the Bitcoin network[19]. One of the most compelling applications of this scheme thus far is compact multi-signatures, as n-of-n signatures are no different from ordinary signatures from the verifier's perspective (the same scheme is used).

In future phases, Crypto.com Chain may incorporate more recent developments from the blockchain research space in order to meet its network scalability and performance demands.
One such development direction relates to the blockchain compression approaches. For instance, Coda[20] is a proposed cryptocurrency protocol which introduces a "succinct blockchain". Instead of storing the entire transaction history, as in the current blockchain systems, it constructs a constant-sized cryptographic proof of the validity of blockchain state; this is accomplished through the recursive composition

[18] C.P. Schnorr (1990), "Efficient identification and signatures for smart cards", in G. Brassard, ed. *Advances in Cryptology—Crypto '89*, 239-252, Springer-Verlag. Lecture Notes in Computer Science, nr 435
[19] https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki
[20] https://cdn.codaprotocol.com/v2/static/coda-whitepaper-05-10-2018-0.pdf

of zk-SNARKs. Coda promises to reduce the enormous blockchain sizes from hundreds of GBs or TBs to merely a few KBs.

Furthermore, when bootstrapping procedures are developed for Tendermint Core, Crypto.com Chain may allow for the safe snapshotting and pruning of historical data that is unneeded for transaction validation.

Crypto.com Chain will initially use the standard network protocol stack (such as TCP+TLS) for different node-to-node communications. Depending on the performance needs, future Crypto.com Chain phases may explore other options. For example, QUIC[21] is a recent protocol standard proposed by Google on top of UDP that improves over TCP and TLS. QUIC can achieve better network latencies than TCP and TLS thanks to various features, such as faster connection opening and negotiation, out-of-order packet delivery or forward error correction.

Lightweight client support

We also provide support to lightweight clients: By connecting to a full note that has access to the complete blockchain, only a small part of the blockchain has to be downloaded and verified by the lightweight client. This allows lightweight users to access and interact with the blockchain without having to synchronise the entire blockchain. As a result, clients will be able to perform/verify payments in cryptocurrency using a mobile wallet on devices operating under resource constraints, such as smartphones or laptops.

To ensure the light client is served by a full node with the correct blockchain data, besides checking their genesis hashes, the lightweight client can query multiple full nodes[22] to obtain a semi-trusted hash and height then check it against the full node that they are connecting to.

Upgrading the Network

Software development is an iterative process. Until the Crypto.com Chain stabilizes, Crypto.com will continue to play a role in upgrading the network directly, taking community contributions into account by rigorously reviewing pull requests.

[21] https://datatracker.ietf.org/doc/draft-ietf-quic-transport/

[22] https://docs.tendermint.com/master/tendermint-core/light-client-protocol.html#where-to-obtain-trusted-height-hash

Every transaction and every request will include fields related to software versioning.

When a non-backwards-compatible upgrade happens, each honest node will be aware of both the version number and the time the version upgrade should begin, and will thus drop any request or transaction that is broadcasted using an older version.

When two nodes connect, a handshake procedure must be established with remote attestation and some standard checks. Connected nodes would also periodically check the security version number[23] with each other; if a connected node is deemed outdated, it will then be disconnected.

Augmented Decentralization

In line with DA_3, the capabilities of Crypto.com Chain Council Nodes will be split and the entities operating them will be extended to third parties. Adding new Council Nodes (or removing them) requires the approval of at least 67% of all the Council Nodes.

This mechanism will enable phased decentralization, meaning that third parties can participate as Council Nodes and ensure that the CRO Network can continue to operate regardless of any unforeseen circumstances in the operation of Council Nodes. Regardless of Council Node-operating entities being removed or added, the transfer of value will still continue to function, and customers and merchants alike can still use the network to spend and receive their cryptocurrencies.

As the large-scale distributed consensus algorithms and incentive mechanisms mature, the validation capability may be extended to all nodes that post CRO collaterals.

[23] https://github.com/crypto-com/chain_docs/blob/master/docs/modules/tdbe.md#security-upgrades

6. Contribution & Integration

Contribution

Crypto.com Chain code is open source and is available at this repository. We encourage research and peer reviews; we also support external open source projects here through bounties. The community can report bugs or request features by opening relevant issues. Any contributor can also suggest bug-fixes or additional features by submitting pull requests. The core development team will review and merge these pull requests according to the contribution guidelines.

Integration: SDK for Crypto.com Chain

To support the seamless integration of application subsystems, we provide some handy libraries and software development kit for the developers, for example:

- cro-clib[24]: C library (SDK and basic light client support);
- sample-chain-ios-example[25]: Sample code to demonstrate how Crypto.com Chain C bindings can be used in an iOS project;
- sample-chain-java-example[26]: Sample code to demonstrate how Crypto.com Chain C bindings can be used in a Java project;
- Chain-nodelib[27]: Node.js SDK library for interacting with Crypto.com Chain.

Application: Crypto.com Pay

Crypto.com Pay is a mobile QR code payment solution powered by the Crypto.com Chain. It enables customers to complete checkout and pay for goods and services with cryptocurrencies using the Crypto.com App, while merchants can get paid in cryptocurrency or in their preferred fiat currency. Specifically, Crypto.com Chain will be the privacy-preserving blockchain network underneath that processes the transactions on Crypto.com Pay. To allow for ease of use and drive integration adoption, we will provide acquirers off-the-shelf SDKs and leverage container technologies during integration, paired with documentation[28] and APIs[29] that are easy to comprehend.

[24] https://github.com/crypto-com/chain/tree/master/cro-clib
[25] https://github.com/crypto-com/sample-chain-ios-example
[26] https://github.com/crypto-com/sample-chain-java-example
[27] https://github.com/crypto-com/chain-nodelib
[28] https://crypto.com/images/pay_integration_overview.pdf
[29] https://pay-docs.crypto.com/

7. Conclusion

Crypto.com Chain is a privacy-preserving payment network that focuses on enabling crypto spending in the real world and thus powering the future of mobile money.

Everyone is free to witness and participate in the network. Actors meeting the adequate staking and compliance requirements can perform validation and settlement activities and get rewarded for it.

We will relentlessly iterate our technical design and implementation until Crypto.com Chain becomes the best way to pay and be paid in crypto— anywhere, anytime, with any crypto, at little to no cost.

Appendix

Reward-related network parameters:

Key	Description
monetary_expansion_cap	The total amount of tokens reserved for validator's reward in the basic unit
reward_period_seconds	The period of reward being distributed (unit: seconds)
monetary_expansion_r0	The upper bound for the reward rate per annum
monetary_expansion_tau	Initial value of tau in the reward function
monetary_expansion_decay	The decay rate of tau.

Slashing-related network parameters configuration:

Key	Description
block_signing_window	Window to calculate validators' liveness
missed_block_threshold	Threshold of total missed blocks
byzantine_slash_percent	Maximum percentage of stake reduction for byzantine validators
liveness_slash_percent	Maximum percentage of stake reduction for validators with low availability

Effects of the reward parameters and sample configurations:

	monetary_expansion_cap	reward_period_seconds	monetary_expansion_decay
Higher	More reserved validator reward	Less frequent reward distribution	Tau decays slower
Lower	Less reserved validator reward	More frequent reward distribution	Tau decays faster
Constraints	Less than the maximum token supply	Value has to be positive	Positive value less than 1000000
Sample configuration	2e18 (20% of the total supply)	86400 (reward distributed daily)	999860 (Tau dropped by 5% yearly)

	monetary_expansion_r0	monetary_expansion_tau
Higher	Higher ceiling for reward rate	Steeper exponential curve
Lower	Lower ceiling for reward rate	Flatter exponential curve
Constraints	Value has to be positive	Value has to be positive
Sample configuration	350 (35% reward rate p.a.)	10 billion