CoinEx Chain

Whitepaper

CoinEx

Contents

Introduction

Centralized exchanges control your funds, and their security risks are the Sword of Damocles to the whole cryptocurrency industry, which has been repeatedly shown by hacker attacks. Mt.Got was hacked twice in 2011 and 2014. After that, Poloniex, Bitstamp, Bithumb and Binance were hacked in 2014, 2015, 2017 and 2019, respectively. Besides asset safety, centralized exchanges are also criticized for other risks and weakness: absconding with customers' funds, the untransparent trading rules, unexpected service outages for subjective or objective reasons, and crazily high listing fees.

Can we reconstruct the exchange market for cryptocurrencies in a more decentralized way? Without registration and approval flow, without single-point failure and censorship, a transparent decentralized exchange (DEX) can solve multiple problems of centralized exchanges. In recent years, several DEX solutions have been proposed: Bitshares [1], Etherdelta [2], 0x protocol [3], OmiseGo [4], Loopring [5], Kyber [6] and Cosmos [7]. Most of these proposals, including Etherdelta, 0x protocol, OmiseGo, Loopring, Kyber, are built on the ERC-20 standard of Ethereum. When a DEX is based on an existing public chain, its capability is limited by the underlying chain. Before Ethereum solves its scalability issue, the DEX solutions based on it can not rival centralized exchanges in processing speed and user experience. Application specific chips have shown dramatic success in PoW mining. This inspires us that application specific public DEX chains may be the solution to the problems of centralized exchanges without compromise on trading speed and user experience.

CoinEx Chain presents a public DEX chain based on Tendermint consensus protocol [7,8] and Cosmos-SDK [9,10]. With transparent trading rules, it is operated by the community and allows users to control their own funds. The native token is CoinEx Token (CET). All the existing CET tokens in ERC-20 form will be 1:1 mapped to the native token of CoinEx Chain.

The Tendermint consensus protocol has the ability to scale its throughput up to 10K TPS in a decentralized way, and confirm within seconds, which makes it an ideal solution for DEX with nearly the same smooth user experience as central exchange. Trading rules are 100% transparent as trading and matching are executed on chain. Users can gain full control of their assets through private keys and digital signatures, which avoids the single-point failure of centralized exchanges. Furthermore, through cross-chain mechanism, CoinEx Chain bridges CET to broader use cases of cryptocurrencies.

CoinEx Chain goes beyond one single public DEX chain. There is a rich ecosystem surrounding the DEX chain. To maximize its throughput, the DEX chain supports only the essential functions, instead of general smart contracts. Since smart contract is the foundation of more complex financial applications, CoinEx Chain will include a Smart Chain supporting smart contracts. The DEX Chain and the Smart Chain interoperate with each other through cross-chain mechanisms, thus we can both ensure the performance of DEX chain and gain flexibility of the Smart Chain.

Privacy and fungibility of cryptocurrencies are always concerned. The privacy provided by anonymous addresses are not enough since they can be tracked by analyzing transactions recorded on chain. Protecting users' privacy is one of the core missions of CoinEx Chain. A dedicated chain with privacy-preserving features will be added to the ecosystem and connected with DEX Chain and Smart Chain, which can promote the privacy and fungibility of all the assets on them.

Except three specific-purpose public chains, CoinEx Chain team also devotes to the following technical innovations:

1) Security: For DEX users, the security of private keys is crucial for asset safety. To protect the private key of wallets, multiparty threshold ECDSA signatures [12,13] will be supported. Compared to regular Shamir secret-sharing (SSS) scheme [11], it can directly utilize the shards of private keys to compute the final signature, without re-constructing the original private key, avoiding the risk of single-point failure of SSS.

2) Consensus protocol: Tendermint protocol requires validators to vote for each proposed block (i.e. sign it with private key), which means the signatures will grow linearly with the validator set and take up too much on-chain storage. Aggregate signatures can solve this problem. Rogue public key attacks to aggregate signature schemes can be avoided by dedicated measurements for consensus scenarios. Or safe aggregate signatures can be adopted in the plain public key model, such as MuSig [14] proposed by Maxwell et al. and the BLS aggregate signatures [15] proposed by Boneh et al.

3) Performance: According to the lessons learned from Ethereum, the authenticated data structure (ADS) used by blockchain has a remarkable impact on the speed of on-chain transaction processing. Cosmos-SDK uses IAVL+ as its ADS [16], which does not show notable improvements than Ethereum's MPT (Merkle Patricia Tree) [17]. Engineering tricks can only partly alleviate the IO bottleneck caused by inefficient ADS. CoinEx Chain will pay close attention to the advances in ADS design and try to improve public chain's performance by optimizing ADS.

Chain Components

Tendermint Core and Cosmos SDK

CoinEx Chain is built upon Tendermint Core and Cosmos SDK. Tendermint Core encapsulates P2P networking and Tendermint consensus protocol. Cosmos SDK provides the basic building blocks for the application layer, in a modularized way. These two components interact with each other through the Application BlockChain Interface (ABCI).

When Tendermint Core processes transactions, instead of considering their semantic, it takes transactions as raw byte arrays. The application layer interprets the transaction sequence and modifies its state accordingly.

Tendermint consensus protocol is semi-synchronous and Byzantine-fault-tolerant, with simplicity, efficiency and accountability. Consensus is achieved among a known validator set. And each validator is identified by its public key. The consensus process includes a multi-round two-phase (prevote and precommit) voting protocol and corresponding locking rules. At the beginning of each round, a new proposer is picked from the validators using round-robin strategy. This validator will pack and propose a new block and then the validator set will carry out a two-phase voting process to confirm this block. If it obtains more than 2/3 votes in both phases, it will be committed to the chain and executed. More than one round may be necessary when the picked validator is offline, the proposed block is invalid or less than 2/3 votes are collected at some voting phase. To simplify the handling of uncertainty, each vote in Tendermint can endorse a valid block or an empty block. The voting result can confirm a valid block or the start of a new round, thus we can avoid the complex view-changing in PBFT consensus algorithm. Tendermint achieves accountability by using public key to identify its validators.

Under the constraint of the CAP theorem [18], the Tendermint protocol prefers consistency over availability. So, it may pause temporarily until more than 2/3 validators come to a consensus. When the Byzantine validators are less than 1/3, Tendermint ensures there is no fork. Consistency and fork-free are crucial to finance applications. The initial number of validators on the CoinEx Chain is 42. According to the experiments data from Tendermint team, when these 42 validators spread out over five continents, they can process 4000 transactions per second (TPS), which is enough for DEX. Tendermint does not sacrifice latency for throughput. Transactions are confirmed in seconds with per-block finalization.

Proof-of-Stake

Tendermint protocol assumes there is a validator set, among which a block proposer is selected for each round, in a weighted round-robin way. CoinEx Chain adopts a Proof-of-Stake mechanism. Any entity can stake CET tokens to become a validator candidate. Validator set is not fixed. Participators in the ecosystem can delegate, undelegate or re-

delegate their CET to different validators to change their voting power. According to the up-to-date voting power distribution, the validator set for the next block is elected.

Staking and slashing can alleviate the "Nothing at Stake" problem [19] for PoS chains. Another severe challenge to PoS chains is the long-range attack [20], because PoS chains only need signatures to finalize a new block, instead of huge power consumption as in PoW chains. When an attacker obtains more than 2/3 private keys of the validators who were active in some past moment, one can fork the chain from that moment, which makes it difficult for new comers or nodes offline for a long time to decide which is the genuine chain. CoinEx Chain follows Cosmos Hub's strategy for defense:

1) Unbounding period: after delegators undelegate their tokens, they must wait for a unbounding period before they can take back the tokens. The unbounding period is three weeks currently.
2) Weak subjective [21]: when a new node joins the network for the first time or a node rejoins the network after long offline period, it must query some trusted nodes for recent blocks' hashes. The foundation of CoinEx Chain will set up trusted nodes to serve the community.
3) Nodes should get online periodically to synchronize the validator set, at least once during an unbounding period.

These strategies above alleviate the long-range attack problem, but the problem is not yet solved thoroughly. VDF (Verifiable Delay Function) [22, 23, 24, 25] has the potential to eliminate the threat of long-range attack, especially the proposition of incremental VDF. CoinEx Chain team will track the progress in VDF research, and utilize the latest research results to further enhance the security of PoS chains.

Validators play the key roles to maintain the consistency of the chain, and there are costs to run a full node, so CoinEx Chain will incent validators. The incentive is composed of two parts: block rewards and the transaction fees in the block. Block rewards are usually generated by minting new coins. PoW chains, such as Bitcoin, mint new coins through mining, and PoS chains, such as Cosmos Hub, mint new coins through inflation. Since minting new coins violates the "No Inflation" promise of CET, CoinEx Chain will use reserved tokens of CoinEx foundation for block rewards. The promised repurchase plan of CET will still be executed on time. Validators' misbehavior and unavailability threaten the stability of the chain, in which circumstances CoinEx Chain will slash the validators. For the misbehavior which violates the consensus protocol, such as signing two different blocks at the same height, a remarkable portion of the validator's stake will be slashed, and the validator will be evicted out of the validator set permanently. For temporary unavailability, a small portion of the stake will be slashed as a warning, and the validator will be jailed for a period. These slashed tokens will be retained to a reserve pool as community incentive in the future.

Account and Transaction

CoinEx Chain adopts an account model. Each account supports multiple assets natively, and assets can be sent on CoinEx Chain. To prevent numerous “zombie accounts” from consuming the on-chain storage resource, the accounts can be only operated after activation. A CET transfer to the new account can activate it and one CET will be deducted from the receivable tokens as feature fee. Inside each transaction, there can be multiple messages for different tasks, such as sending tokens, withdrawing rewards etc. Transactions with signatures authenticate account and the signing algorithm is ECDSA defined over elliptic curve secp256k1. CoinEx Chain supports multi-signature transactions. Currently, it uses a similar scheme like Bitcoin, which includes multiple signature and public keys in the transaction. This scheme is easy to implement but consumes too much computation and storage resource for the downside. CoinEx Chain charges transaction fees and only CET is accepted. Transaction fees include two parts: the usual gas fee (like Ethereum) and feature fee. Gas is calculated according to the size of transaction, the signature count, the read/write count to persistent storage and the length of read/write data. Feature fee is an extra fee charged for some particular operations, for example, issuing a new asset, listing a new trading pair, activating an account and transferring tokens with a lock time. The matched orders are charged according to the dealt amount with a configurable rate, which also falls in the category of feature fees.

CoinEx Chain is scheduled to improve the way to construct multi-signature transactions by compressing signatures and/or public keys via aggregate signature scheme. In this way, the size of multi-signature transaction and the required computation resources for verification can be reduced. A compressed n-of-n multi-signature transaction will be the same as a standard transaction since the signatures and public keys can both be aggregated. In this way, the privacy can also be improved because aggregated signature and public key hide the information of involved entities. With the help of Merkle proofs, the same privacy preserving property can be achieved for m-of-n multi-signature transaction in blockchains with scripting system [26]. However, more investigation is needed on how to accomplish this without scripting system. The biggest challenge of aggregate signature scheme is how to guarantee security in the plain public key model under threat of rogue public key attack. Rogue public key attack utilizes this fact and allows an attacker to cook up an aggregated signature without involved entities ever knowing. Conventionally, there are two methods to mitigate rogue public key attack: requesting KOSK (Knowledge of Secret Key), or always prepending the public key to every message prior to signing. Requesting KOSK is difficult to enforce in practice while prepending the public key to every message prior to signing can partially neutralize the efficiency improvements provided by aggregate signature scheme. In the plain public key model, the involved entities are not required to prove KOSK corresponding to one’s public key. Luckily, the MuSig scheme proposed by researchers from Blockstream [14] and the pairing-based aggregate BLS signature proposed by Boneh et al. [15] can meet the security criteria without introducing the drawbacks of the aforementioned methods.

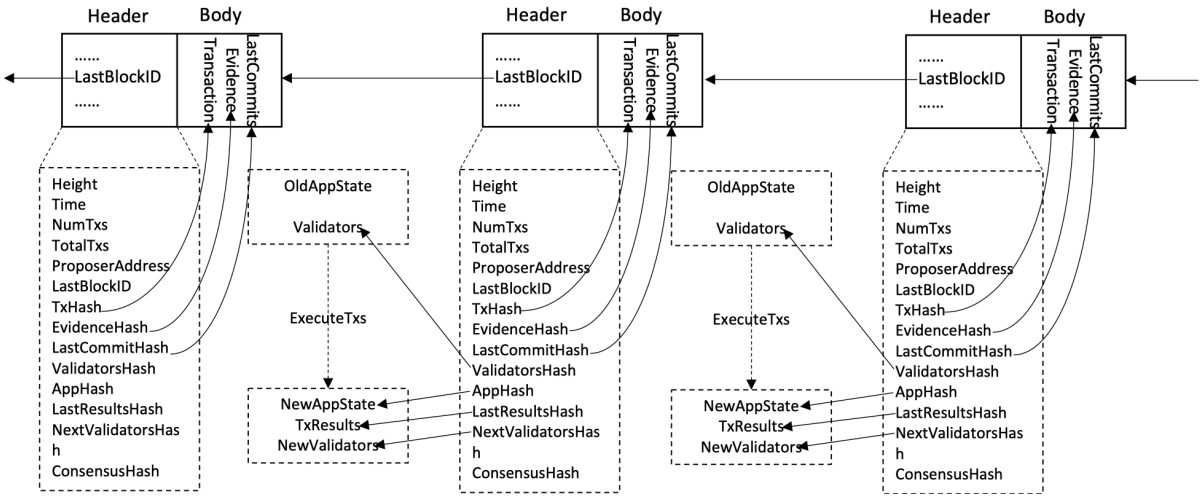
The MuSig scheme [14] built upon Schnorr signature scheme [27] can securely aggregate public keys and signatures in the plain public key model and the verification procedure is the same as verifying a Schnorr signature. The Schnorr signature has been activated on Bitcoin

Cash network to pave the way for adopting MuSig scheme in the future. Also, there are a series of major upgrade plans related to Schnorr signature scheme and MuSig scheme to be deployed on Bitcoin network [26, 27, 28, 29]. The nice thing about MuSig scheme is that it can be initialized upon elliptic curve secp256k1, a cryptographic primitive that is already supported by Cosmos SDK. The pairing-based aggregate BLS signature scheme proposed by Boneh et al. [15] can also be utilized to improve multi-signature transactions in the plain public key model. But, supporting pairing-based aggregate signature scheme would require significant modification to the current wallet implementation, especially hierarchical deterministic wallets[30], in order to complete the transition from secp256k1 to pairing-friendly elliptic curve (e.g. BLS12-381 curve constructed by Zcash project [31]). Thus, MuSig seems like a more compatible way to improve the multi-signature transaction with the current setup.

Ideally, all transactions' signatures in a block can be aggregated, leaving only one signature per block to be verified. This requires an aggregate signature scheme that can aggregate signatures produced by different keys for different messages. In this case, the miner can aggregate all the transaction signatures in the block. MuSig scheme does not meet this criterion yet. It is mentioned in [14] that fixed Interactive Aggregate Signature scheme can aggregate signatures for distinct messages, but rigorous security proof is still missing. The aggregate multi-signature scheme (AMSP) based on a stronger security assumption proposed in [15] can be utilized to aggregate signatures across many transactions, yielding more on-chain space saving. Besides the stronger security assumption, more investigation is needed on how to securely deploy this construction in the blockchain context.

Blockchain

Instant finality and the guarantee of no forks (in the presence of asynchrony if less than 1/3 of processes are fault) provided by Tendermint consensus protocol greatly simplifies the design and implementation of the underlying blockchain data structure. A single linked list with hash pointer is enough. There is no need to tackle the existence of uncle blocks as in Ethereum in the aspect of data structure design, or handle the blockchain re-organization events as in Bitcoin in the aspect of implementation. Block in Tendermint Core is composed of the block header and block body. In block header, there are block height, time, number of transactions in the current block (NumTxs), number of accumulated transactions till the current block (TotalTxs), hash pointer to last block (LastBlockID), proposer of the current block (ProposerAddress), Merkle roots of transactions (TxHash), votes for last block (LastCommitHash), evidences (EvidenceHash), last block's validator set (ValidatorsHash), the new validator set updated by last block (NextValidatorsHash), upper-level application state (AppHash), executed results of the transactions in the last block (LastResultsHash), etc., as the following figure shows. The transactions, evidences and votes are included in the block body.



Note that the votes for a block are actually stored in the next block. The vote as defined in the Tendermint consensus protocol is essentially a signature over the block, produced with validator's private key. Currently, the adopted signature scheme is EdDSA scheme defined over elliptic curve Ed25519 [32]. With the growing of validators' set, the storage and verification of the votes will consume non-negligible resources. Ed25519's batch verification mechanism can be utilized to speed up the verification process. But an aggregate signature scheme can improve both the storage consumption and the verification process. MuSig scheme requires multiple rounds of communication when producing the final aggregated signature. Considering the fact that validators would spread all over the world, we'd like to avoid more interactive operations except for what are required by the consensus protocol. Hence MuSig is not an appropriate improvement method in this context. Note that unlike the foregoing transaction signature part which is closely related to the hierarchy deterministic wallet, the voting procedure is a separate module on its own. This means new signature scheme can be introduced without significant modification to existing codebase. In this case, the aggregate BLS signature scheme in [15] is a more appropriate method to accomplish the expected improvements to Tendermint Core's voting procedure.

For better support of cross-chain mechanism and light clients, the block headers include merkle roots for the state of the upper-level application, which provide existence proofs and non-existence proofs to light clients for fast verification. The per-block updating to state requires incremental appending and modification of ADS. Currently, CoinEx Chain uses IAVL+ as ADS, which is developed in the Tendermint Core project.

IAVL+ does not show notable improvements, compared to Ethereum's MPT (Merkle Patricia Tree). According to the lessons learned from Ethereum, the operation of upper-level application is amplified by the MPT data structure (accessing the nodes from root to leaves need multiple operations of the underlying KV database), which is a potential bottleneck for performance. Engineering tricks can only partly alleviate the IO bottleneck caused by inefficient ADS. Solving this problem needs new ADS design. A possible ADS design is like this: its special tree structure allows the internal nodes of Merkle tree to be stored outside of the KV database and can be recalculated from the data in KV database after unexpected

crashes, which keeps the consistency of storage. Because the frequently used internal nodes are cached in DRAM and infrequent nodes are on the disk, most of the access to internal nodes can be performed very fast without accessing the disk, greatly reducing the pressure to the underlying KV database.

Private Key Security

Ownership of assets in cryptocurrency domain is authenticated with digital signature produced by a private key. To transfer assets with a valid transaction requires access to the private key. The protection and management of private key has always been one of the confronting problems that the whole blockchain industry faces. In general, on-disk protection of the private key in digital wallets is achieved via encryption and the encrypted private key is stored in a keystore file. The encryption key is derived with key derivation function using user password among other things as input. The on-disk guard solution might be sound, but the harder part is the in-use protection of private key. To sign a transaction, the on-disk encrypted private key needs to be decrypted, which will leave the private key in a plaintext form in the system, thus leakage risk is introduced. One can choose to separate the system and use human aided interaction whenever necessary, as the strategy adopted by cold wallet. The private key leakage risk can be reduced with cold wallet and this can be regarded as security enhancement via people management.

Hardware Security Model (HSM) is normally the go-to solution for scenarios requesting stronger security protection. The private key is generated and managed by HSM. HSM can produce valid signature when needed and guarantees that the internal private key will never leave the HSM in a plaintext form. Same as cold wallet solution, HSM can provide better protection but is not convenient or flexible enough, especially when the private key access control needs to be distributed among multiple entities. It is unpractical and a huge burden to require each involved entity possessing HSM. Besides, although HSM is normally regarded as an unbreakable security vault, researchers from Ledger will present at Black Hat 2019 showing that HSM itself can be hacked [33, 34].

By requiring m distinct signatures to move the assets, m-of-n multi-signature transaction can distribute the ownership of the assets among multiple entities. To steal funds in this scenario, the attacker has to crack m entities, which is presumably more difficult. By adjusting the parameters of m and n, multi-signature mechanism can tolerate the undesired but unavoidable events of losing private keys. As long as the number of lost keys is less than n-m, the assets is still under control rather than lost forever. The problem with the multi-signature scheme is that it usually costs more gas fee. Besides that, alternation of the access control strategy is cumbersome and with all the public keys and signatures published on the blockchain, the m-of-n strategy and the involved entities' public key are also leaked.

Shamir's Secret Sharing (SSS) scheme can solve the problems of multi-signature scheme. With SSS scheme, assets' ownership is controlled by only one private key, and transferring the asset requires only one signature which is basically the same as standard transaction. However, this private key is controlled by multiple entities, by splitting the key into multiple slices and distributing the slices to different entities. The original key can be reconstructed

with enough key slices. Similar to multi-signature transactions, one can choose to use m-of-n SSS scheme where at least m slices are required to successfully recover the original key while m-1 or less slices reveal zero information about the original key. By using m-of-n SSS scheme, the solution also provides enhanced key security and tolerance of key slice erasion and loss. The reconstruction of the original key needs to be carried out by an entity, meaning that the chosen entity would have full knowledge of the original key. The scheme has to trust that this entity would securely erase and forget the key after use. Once again, on the way of developing distributed key access control, the trusted entity or single point of failure is introduced.

The development in the field of secure multiparty computation of cryptography provides more tools to secure the private key. The recently proposed multiparty threshold ECDSA ({m,n} threshold ECDSA) scheme [12,13] can be utilized to solve the problems of multi-signature scheme and SSS scheme while still preserving the advantages:

1) Similar with SSS scheme, multiparty threshold ECDSA scheme divides the private key into multiple parts to be possessed by different entities. Unlike SSS scheme, with multiparty threshold ECDSA scheme, there is no need to construct the original key information and the single point failure of SSS scheme is solved. This is the magic of multiparty threshold ECDSA: by combining intermediate results produced by each entity with one's own key part, a valid signature corresponding to the original private key can be produced while the private key never appears during the computation.
2) By choosing proper m and n values, enhanced security and tolerance of key loss can be achieved with threshold ECDSA scheme. As there is only one signature produced, the transaction produced this way is exactly a standard single signature transaction. This reduces the gas fee and avoids the leakage of access control strategy of multi-signature.

CoinEx Chain team plans to implement a proper multiparty threshold ECDSA scheme as an option among other things to further enhance the protection of private key.

DEX

CoinEx DEX chain is a public chain specially developed for decentralized exchange based on the Tendermint consensus protocol. On DEX chain, users can receive and send CET tokens, issue new tokens and conduct minting, burning, locking, unlocking operations to the new issued tokens. Besides, one can also create trading pairs, place orders, query transaction records, and compete to serve as a validator operator.

The risk of single-point failure inherent to centralized exchanges is eliminated by returning the control right of users' assets back to users, who are in charge of their own private keys. The scheduled support of multiparty threshold ECDSA signature scheme of CoinEx Chain will help users to further enhance the private key security. The adoption of Tendermint consensus protocol and the principle of lean on-chain functions allow for block time within seconds and instant transaction confirmation. Moreover, a fair and transparent trading experience is realized through the strategy of asset-mapping, on-chain trading and on-chain matching. Different from traditional issuance via smart contracts, issuance function will be natively embedded in CoinEx Chain, making it more efficient and secure to issue tokens. Without any permission, users can issue tokens (including but not limited to stable coins) and create its trading pairs simultaneously, which will free users from the lengthy process and high listing fees of centralized exchanges. Every step of the operation on the DEX public chain is standardized and consumes predictable resources, so the DEX public chain can process up to thousands of transactions per second.

CET will be 1:1 mapped to the CoinEx Chain, used as fees for on-chain transactions, and staked too. CET holders can participate in staking. In addition, CET holders can also initiate proposals and vote on them, participating in community governance.

CET Mainnet Mapping

The mainnet mapping of CET will be completed jointly by CoinEx Foundation and CoinEx's business ecosystem. CET holders need to deposit ERC20 CET to CoinEx exchange and the corresponding CET on the mainnet will be distributed to CoinEx exchange, after which users can withdraw CET from CoinEx exchange. The specific process is as follows:

1. Before the launch of the mainnet, the exchange is only available for deposit of ERC20 CET and the withdrawal service will be suspended.
2. All the unlocked tokens are deposited in the CoinEx exchange.
3. When the mainnet is launched, both CET in CoinEx exchange and the locked ERC20 CET on ethereum will be mapped to the mainnet.
4. Open withdrawal and deposit for mainnet CET.
5. Users can create a CoinEx Chain account address and withdraw the mainnet CET from CoinEx exchange.
6. Users can also withdraw coins to a third-party Wallet or an independent Wallet and use CET on the mainnet to participate in staking.

CET Allocation

After mainnet mapping, the CET distribution will be as follows:

Holder	Account type	Amount	Unlocking time (UTC)
Common Users	Normal Account	2.888 billion	Circulating
CoinEx Foundation	Normal Account	885 million	Unlocking in Need
CoinEx Foundation	Normal Account	315 million	Reserved Inventives for reward
CoinEx Team	Locking Account	360 million	2020/1/1
CoinEx Team	Locking Account	360 million	2021/1/1
CoinEx Team	Locking Account	360 million	2022/1/1
CoinEx Team	Locking Account	360 million	2023/1/1
CoinEx Team	Locking Account	360 million	2024/1/1

CET Incentive

As mentioned above, CoinEx will honor its previous promise not to issue additional CET and will not create new tokens by inflation. However, block incentive is crucial to community participation. Therefore, after the mainnet is launched, CoinEx Foundation will allocate about 315 million CET as incentives to the initial validators and staking participants. The time span for distributing 315 million CET incentives is related to the interval between blocks. The incentive plan is estimated with a 3-second-per-block assumption. The specific reward for each block is shown as follows:

	Starting height	Ending height	CET amount	Incentives per block
Year 1	0	10512000	105,120,000	10
Year 2	10512000	21024000	84,096,000	8
Year 3	21024000	31536000	63,072,000	6
Year 4	31536000	42048000	42,048,000	4
Year 5	42048000	52560000	21,024,000	2

In addition to the block incentives, reward for each block also includes the transaction fee, which consists of two parts: Gas fee and feature fee: gas fee is charged to prevent the malicious abuse of system resources and feature fee is mainly used to improve the quality of the ecosystem on-chain, which prevents the malicious use of related functions and ensures

user experience. When the mainnet is launched, the feature fee for special operations will be set up, and in the later stage, it can be adjusted by means of community proposal according to the evolution of the main chain. Special operations include issuing new tokens, adding new trading pairs, locking transfers, activating new accounts and order matching.

Token Issuance and Trading

Without any permission, any user can issue new tokens and create its trading pairs simultaneously free from approval process. To ensure the liquidity, the first trading pair created for the newly-issued token must be the new token against CET. To avoid the abuse of system resources and ensure the quality of the on-chain ecosystem, a certain amount of CET will be charged as feature fee for issuing token and creating new trading pairs. The Symbol of new tokens is composed of 2-8 characters/numbers and cannot begin with a number. The accuracy of token is 8 decimal digits, and the maximum amount of a token is 90 billion. The token issuer is also the token owner, and the ownership can be transferred to others.

Options for issuing new tokens include burning, minting, locking address and locking tokens. These options can only be specified when issuing the token and cannot be changed after issuance. If locking address and locking token options are not enabled during the issuance, the transfer and ownership of the token is free and unrestricted.

When “lock address” option is activated, the token owner can lock any addresses as needed. The token in any locked address cannot be transferred or used for exchange, but other tokens in the address will not be affected. When “lock token” option is enabled, the token owner can lock the tokens on demand, during which transfer and exchange of the tokens are fully disabled. During the locking period, the token owner can create a whitelist of addresses that can initiate transfer transactions but no exchange transactions. And during that period, the token owner’s operations on the tokens are unaffected.

	Currency is forbidden		Addresses are forbidden
	Normal address	Whitelist addresses & Token Owner	Forbidden Addresses
Creating order	✗	✗	✗
Matching order	✗	✗	✗
Transfer	✗	✓	✗
Receiving	✓	✓	✓

Governance

The initial number of validators on the CoinEx Chain is 42. In the scenario where the upper limit of validators is not reached, anyone can become validator by sending the CreateValidator transaction. After the number of validators reaches the upper limit, validators will be sorted by the amount of staked CET, and 42 validators with the highest staking quantity will be selected.

Community governance is achieved through proposal and voting. A validator can vote on behalf of its delegators, while the delegators have the right to vote on their own and overwrite the validator's vote.

There are four voting options: Yes, No, No With Veto, Abstain:

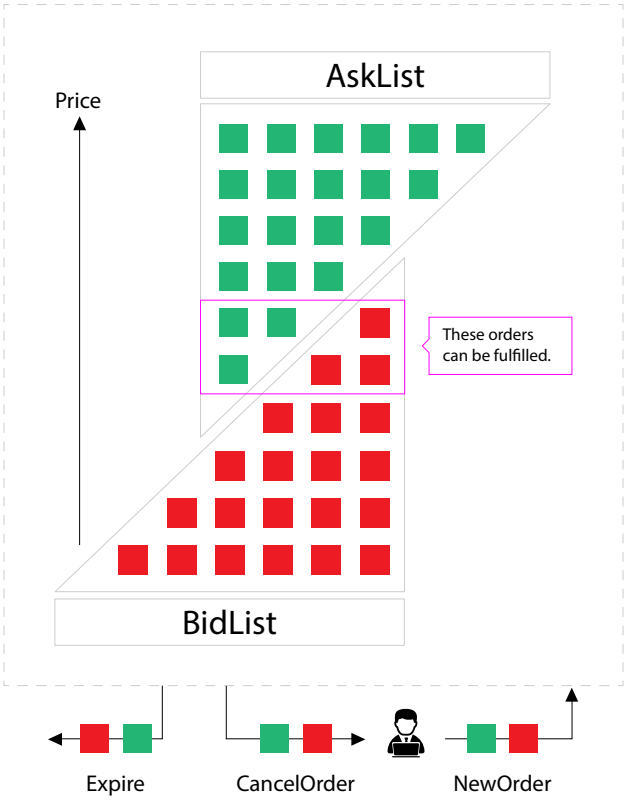
- If more than one third of the votes are "No With Veto", the proposal is rejected;
- If participated voting power does not reach 40% of the staked voting power, the proposal is rejected;
- If more than half of non-abstain votes are Yes, the proposal is considered passed.

When the proposal is made, the community is required to stake 10000 CET to the relevant proposal as a pledge to prevent the abuse of proposal. After the proposal is passed, the staked CET will be returned to the corresponding account address. If the proposal is rejected due to the following reasons, the pledged stake will not be returned and instead, be retained by the system for community incentives in the future.

- The stake fails to reach 10000 CET, meaning the community is not interested in or does not support the proposal;
- Participating voting power does not reach 40% of the staked voting power;
- More than one third of the votes are No With Veto.

Order Matching

Similar to mainstream centralized exchanges, Order Book is used for matching. Please refer to the figure below:



The order book contains an Ask List marked green and a Bid List marked red. An ask order always wants to drive up the price while a bid order always wants to lower the price. Currently, only limit orders but not market orders are supported. If the Ask1 price and Bid1 price are not crossed, the order cannot be fulfilled.

The way Ask List and Bid List are organized internally is that the orders are sorted according to their prices and ages: orders with better prices are prioritized to deal, and among the orders with same price, the ones with older age are prioritized to deal. In the above figure, ask orders with lower price are placed in the queue head (at the bottom), and bid orders with higher price are placed in the queue head (at the top).

When there is a crossover between the Bid1 price and the Ask1 price, the bid orders which have higher price than Ask1 price and the ask orders which have lower price than the Bid1 price will be matched in the sequence decided by their prices and ages. The ask orders and bid orders as circled with the purple box in the figure are eligible to participate in matching and possible to fulfill. Whether or not they can all be filled depends on the amount to be bought and sold.

Users can submit new bid orders or ask orders to the order book through the New Order transaction, or cancel self-placed orders from the order book at any time. Good Till Expire

(GTE) orders and Immediate Or Cancel (IOC) orders will be automatically deleted from the order book after expiration. The former expires at 00:00 UTC after the predefined life time, which can be lengthened by paying more feature fee; While the latter expires at the next block after entering the order book (i.e., there is only one chance to be matched).

For off-chain order matching, orders are always accepted by the server one by one and can be prioritized by the submission time. The most important difference for order matching on-chain is that orders are packaged in blocks and the orders within the same block cannot be prioritized by time. In order to ensure that the orders within the same block are treated equally, we adopt the method of "batched auction" for order matching. For all bid and ask orders that can be filled, a single price for execution will be calculated. The principles are as follows:

1. Maximum matched volume.
2. Minimum remaining volume. If more than one price has the same executable volume, the price should be the one with the lowest remaining volume. The remaining volume is the unexecuted amount for all orders with executable price.
3. Market Pressure. If multiple prices satisfy 1 and 2, then identify where market pressure of the potential price exists. A positive remaining amount indicates bid side pressure and higher prices are preferred, while a negative remaining amount indicates ask side pressure and lower prices are preferred.
4. When both positive and negative remaining amounts exist, the latest execution price is reference price, and the price closest to the reference price should be chosen.

Front running, which is inevitable for both centralized and decentralized exchanges, refers to obtaining transaction-related information in advance based on technological or market superiority. This will bring benefits to the traders and losses to other participants by affecting transaction price. Centralized exchanges can take a more holistic view of transaction information, making it possible to formulate the optimal transaction strategy given current market dynamics and reap benefit by priority transaction before the orders are matched. Similar problems also exist in decentralized exchanges; therefore, various measures are taken in decentralized exchanges to hinder profit-making by front running.

CoinEx DEX Chain natively supports certain anti-front-running features. Firstly, Tendermint supports block time within seconds, which makes the time window for front-running very narrow. Secondly, in the P2P network, it is very difficult to know all the broadcasting orders and which orders will be included in the next block. Finally, prices are calculated in batched auction, which limits the advantage of a front-running transaction and the normal transactions in the same block. So ordinary users can hardly make a profit from front-running.

A validator can select transactions to pack into next block. It can perform "censorship attack" to include self-generated profitable transactions, and exclude others' transactions conflicting with its own, which might harm other users' interest. But, as the transaction data and execution logic are all open, if a validator frequently performs censorship attack and packs self-generated transactions without pre-broadcasting, then it can be observed easily. This will cause injury to the validator's credit, and the validator will lose the support from delegators.

A possible improvement is that users utilize the “commit-reveal” scheme to hide the detail of their orders for a short period (for example, 2~3 blocks). Thus the validator cannot see the full content of the order book when it propose on the next block, making it even harder to construct profitable front-running transactions. CoinEx Chain team will continue to study this issue and provide advanced countermeasures against front-running.

Automatic Market Making

Currently, most exchanges use Order Book for matching. To successfully make a match, there should be demands from both buyers and sellers. Also, to make a deal, there should be crossover between “ask” and “bid”, meaning the Bid1 price should be greater than or equal to the Ask1 price. Currencies with plenty liquidity, BTC, for example, could be traded in exchanges to a relatively large volume due to large number of orders between buyers and sellers. But those unpopular currencies may not be able to support large orders. To solve this problem, the market makers are introduced to increase the liquidity. The philosophy is, the market makers balance the market liquidity through certain rules and earn a profit from the price differences in the transactions, compensating the costs of services provided.

Market makers face a big problem of high cost. This cost is constituted of two parts: commission charged by the exchange and service fees charged by market makers, for increasing the liquidity. On the other hand, although the market makers conduct activities via automated programs, they have very limited capability when there is large number of single-side orders. Ultimately, market making depends on real transaction volumes. CoinEx Chain will meet the demands of token liquidity through algorithms in DEX public chain based on two automatic market making protocols, Bancor [35] and UniSwap [36,37]. CoinEx Chain will extend the two protocols for better adapting to DEX and providing more liquidity at a reasonable price. The market maker will only need to provide capital instead of setting the price when providing such liquidity.

Bancor protocol supports the operation of Token transaction network with decentralized liquidity, without relying on the demand match between the buyer and seller but setting the price through connectors. Token price = connectorBalance / (smartTokenSupply * CW), where Connector Weight (CW) is the parameter that defines how sensitive the token price is to be influenced by the token supply. Users can buy Token from connector at a price calculated automatically. Vice versa, they can sell Token to connector at a price calculated automatically.

UniSwap is a decentralized token exchange protocol. It sheds the concept of a limit order book entirely and uses a Constant Product Market Maker Model to pool everyone’s liquidity together. What’s more, UniSwap will incrementally increase the token price while as the number of bid orders increases, and will incrementally decrease the token price as the number of ask orders increases. Through algorithms, a decentralized exchange system that does not require pending orders nor market depth can be achieved. The core concept of Constant Product is x*y=k, where x is the quantity of Base Currency, y the quantity of Quote Currency,

and k the product of the two quantities. When k is kept constant, the larger the value of x is, the smaller the value of y will be, and vice versa.

Multi-Chain and Cross-Chain

Smart Chain

To achieve maximum transaction processing speed, the DEX chain supports only the necessary functions required by a decentralized exchange. The experience of Ethereum has shown that smart contract is indispensable component when it comes to building decentralized finance technology. To complete CoinEx Chain ecosystem and to build programmable cash, based on the idea of application specific chain, the team of CoinEx Chain is tasked with the mission of building a smart chain with integrated smart contract functions.

Privacy Chain

Privacy protection is another challenging mission in the field of blockchain as no one wants to be tracked in economic activities. Bitcoin and Ethereum's pseudorandom address is not enough to protect one's activity from being tracked. Numerous cryptographic schemes have been proposed to protect privacy and some of them are implemented in blockchain projects. Among which, the prominent ones are Dash [38], Monero [40], Zcash [44] and Grin/Beam [46,47] etc.

Dash utilizes CoinJoin [39] method to combine multiple transactions from multiple senders into one unified transaction to improve privacy. Monero uses Pedersen commitment and range proof [41,42] to hide the amount carried in a transaction, linkable ring signature [43] to hide the transaction sender while still preserving the ability to identify double spend, and stealth address to hide the receiver of a transaction. Zcash utilizes highly sophisticated zero-knowledge succinct non-interactive argument of knowledge (zkSNARK) [45] to hide amount, sender and receiver of a transaction. Grin and Beam are two projects built upon MimbleWimble protocol [48] which uses Pedersen commitment, range proof, as well as two-party interactive aggregate Schnorr signature scheme to protect onchain information. All these projects are built under unspent transaction output (UTXO) model, and the cryptographic constructions are evolving gradually to provide better privacy protection and to reduce computation as well as storage consumption.

However, there are very few mature privacy protection solutions working with the account model. During the Byzantium hard fork of Ethereum, by introducing the zkSNARK technology from the Zcash project and with the help of smart contract, Baby ZoE managed to bring some Zcash's privacy features into Ethereum [49].

Zerochain project [50,51] is another attempt to build a privacy-preserving chain under account model. Zerochain project tries to hide the amount of a transaction as well as the balance of an account. A proof-of-concept implementation is built with the Substrate blockchain development framework [54] and the adopted cryptographic constructions are lifted-ElGamal public key encryption scheme [55] featuring the homomorphic addition

property and zkSNARK scheme from Groth16 [52]. The homomorphic addition property of Lifted-ElGamal allows to hide the amount in a transaction and update the balance of an account in an encrypted way while Groth16's [52] zkSNARK provides an efficient way to prove something that is true in a zero-knowledge way. The PoC shows that the Zerochain's method is a feasible way to protect online information. Yet, there is no discussion of how to deal with problems such as replay protection and front-running problems that are unique to privacy solution working with account model.

Zether [56] is a solution of privacy-preserving for account model recently proposed by researchers from Stanford and Visa research department with detailed discussion of how to deal with replay protection and front-running problems. Zether also adopts the lifted-ElGamal public key encryption scheme [55] to hide transaction amount and account balance. Instead of using zkSNARK technology, Zether utilizes an improved version of Bulletproof called Σ-Bullets to improve the zero-knowledge proof procedure. Zether is implemented as a smart contract in the proposition paper [56], but can also be used to build an anonymous blockchain in account model. Besides, by further extending the Zether protocol, the sender and receiver of a transaction can be hidden in a group of accounts, making Zether a complete privacy preserving solution for account model with the ability to hide not only the amount, sender and receiver of a transaction but also the account balance. Notably, Zether is independent of the underlying blockchain's consensus protocol. Implementing Zether protocol upon Tendermint Core and Cosmos SDK is a feasible way to build an anonymous blockchain in account model. This is also the mission of CoinEx Chain team. Note that JPMorgan has done the first attempt towards the aforementioned direction, by integrating Zether protocol to its Ethereum-based Quorum blockchain [57]. In the meantime, the CoinEx Chain team will pay close attention to the evolvement of privacy preserving solutions for blockchains built upon account model to build privacy chain with cutting-edge privacy preserving solution.

Cross-Chain

It's impossible to build one chain that satisfies all the needs of people. The coexistence of multiple chains with different feature and usage is a growing trend and will be the direction of future development. The ability to transfer information as well as values between heterogeneous blockchains would promote the development and technology of blockchain significantly.

There are basically three primary strategies for blockchain inter-operation as Vitalik Buterin summarized in [58]: hash-locking, centralized or multi-signature notary scheme and relays. Hash-locking has been extensively used to build payment channels such as lightning network [59]. It's impossible to achieve fully decentralized trust in notary scheme. But when inter-operated with existing public blockchain without finality feature, such as Bitcoin and Ethereum, notary scheme is the most practical way and has been deployed in practice.

Relay scheme is the most appropriate scheme to realize decentralization and is also the inter-operate scheme adopted by Cosmos and Polkadot, the two most famous projects focusing on blockchain inter-operation. The inter-blockchain communication (IBC) protocol proposed by

Cosmos supports both value and data inter-operation between different blockchains, and is more compatible with the underlying platform of CoinEx Chain: Tendermint Core and Cosmos SDK. CoinEx Chain will continue to follow the IBC protocol for cross-chain inter-operation.

Atomic swap, based on hash-locking, is a mature and easy-to-implement method for cross-chain asset transfer, and CoinEx chain will adopt it for multiple applications, for example:

1. It helps in issuing assets on CoinEx chain anchoring other cryptocurrencies (such as BTC, ETH). With the issuer, user can atomically exchange some cryptocurrency on its native chain, and the corresponded token on CoinEx Chain.
2. Token issuers can issue homogeneous tokens on multiple chains, like USDT, which is issued on Bitcoin, Ethereum, EOS and Tron. The token issuer can issue her token simultaneously on Ethereum, CoinEx Chain and even other public DEX chains and their users can transfer tokens among these chains by atomically exchanging with the issuer.

Conclusion

CoinEx Chain is committed to building the next generation of blockchain as financial infrastructure which includes a series of public chains to realize programmable cash. CoinEx Chain has planned three public chains for specific applications:

1) DEX Chain that supports decentralized exchange functions;
2) Smart Chain that supports smart contract functions;
3) Privacy Chain that supports on-chain privacy protection.

The three public chains are interconnected through IBC (Inter-chain Communication Protocol). Each of them plays its own role while cooperating with each other to provide full functions.

With functions such as asset-mapping, on-chain trading and on-chain matching, DEX public chain can address the problems of centralized exchanges, such as poor safety and intransparency that have been widely criticized. Returning asset control to users, Order-Book based fair on-chain matching algorithm, permissionless on-chain listing and creating trading pairs, all these efforts are aimed to build a transparent, secure, and permissionless financial platform for free trading. At the same time, features such as high TPS performance and second-level transaction confirmation provided by the underlying Tendermint Consensus can maximumly restore the user experience as that in centralized exchanges.

To maximize the transaction processing speed, DEX chain will limit its functions to the essential ones needed for a decentralized exchange. Yet, smart contract is a must-have component to build decentralized finance technology. Besides that, the on-chain privacy has always been a sharp focus of the blockchain industry. Take these into consideration and comply with the idea of application specific blockchain, CoinEx Chain is committed to building two more public blockchains, Smart Chain and Privacy Chain: with the integration of smart contract function, Smart Chain will act as the platform to build more complex and rich finance applications; with the help of developments in the cryptographic constructions in the domain of on-chain privacy preserving technology, such as the Zether protocol, Privacy Chain aims to build a public blockchain in account model where the sender, receiver, single transaction amount and the account balance can be hidden with strong cryptographic primitives.

DEX chain, Smart chain and Privacy chain are not islands isolated from each other. Instead, through a relay-based cross-chain communication scheme, they are interconnected and complementary with each other. CET tokens that need to involve in complex financial contracts can be transferred to the Smart chain from the DEX chain, and then back to the DEX chain when finished. CET tokens that need to involve in token mixing can also be exchanged through the private transaction of the Privacy chain, and can eventually return to the DEX chain. In this way, the three public chains perform their duties. Apart from the guaranteed transaction processing speed and function attributes, they can also jointly provide more functions in a safer way. In the future, based on the community requirements, CoinEx

team will continue to build application-specific public chains to further enrich the CET ecosystem.

CoinEx team will also optimize the existing technology infrastructure. For example, attempts will be made to use the incremental VDF mechanism to enhance public chain security which is based on the PoS scheme, and MuSig solution will be deployed to improve the multi-signature transactions so as to reduce space occupied on chain and enhance the privacy in multi-signature transactions. CoinEx team will utilize the BLS signature aggregation scheme to improve the voting process of the Tendermint Consensus protocol to reduce the on-chain storage occupied by votes. In addition, CoinEx team will provide an advanced solution for users' private key protection through the multiparty threshold ECDSA signature scheme.

References

1. Bitshares Blockchain. Open-source business development and financial management platform. https://bitshares.org/
2. EtherDelta. https://etherdelta.com/
3. 0x Protocol. Powering Decentralized Exchange. https://0x.org/
4. Joseph Poon, and OmiseGO Team. OmiseGO: Decentralized Exchange and Payments Platform. 201706. https://cdn.omise.co/omg/whitepaper.pdf
5. Daniel Wang, Jay Zhou, Alex Wang, and Matthew Finestone. Loopring: A Decentralized Token Exchange Protocol. 201809. https://loopring.org/resources/en_whitepaper.pdf
6. Kyber: An On-Chain Liquidity Protocol v0.1. 201904. https://files.kyber.network/Kyber_Protocol_22_April_v0.1.pdf
7. Ethan Buchman. Tendermint: Byzantine Fault Tolerance in the Age of Blockchains. 201606. https://allquantor.at/blockchainbib/pdf/buchman2016tendermint.pdf
8. Ethan Buchman, Jae Kwon, and Zarko Milosevic. "The latest gossip on BFT consensus." arXiv preprint arXiv:1807.04938 (2018). https://arxiv.org/pdf/1807.04938.pdf
9. Cosmos SDK. Blockchain Application Framework. https://cosmos.network/docs/
10. Jae Kwon, and Ethan Buchman. Cosmos: A Network of Distributed Ledgers. https://cosmos.network/cosmos-whitepaper.pdf
11. Adi Shamir. How to share a secret. Communications of the ACM 22, no. 11 (1979): 612-613.
12. Yehuda Lindell. Fast secure two-party ECDSA signing. In Annual International Cryptology Conference, pp. 613-644. Springer, Cham, 2017. https://eprint.iacr.org/2017/552.pdf
13. Rosario Gennaro, and Steven Goldfeder. Fast multiparty threshold ecdsa with fast trustless setup. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1179-1194. ACM, 2018. https://eprint.iacr.org/2019/114.pdf
14. Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multi-signatures with applications to bitcoin. Designs, Codes and Cryptography (2018): 1-26. https://eprint.iacr.org/2018/068.pdf
15. Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multi-signatures for smaller blockchains. In International Conference on the Theory and Application of Cryptology and Information Security, pp. 435-464. Springer, Cham, 2018. https://eprint.iacr.org/2018/483.pdf
16. IAVL+ Tree: Merkleized IAVL+ Tree implementation in Go. https://github.com/tendermint/iavl
17. Gavin Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger Byzantium. Version d6ff64f - 2019-06-13. https://ethereum.github.io/yellowpaper/paper.pdf.
18. Seth Gilbert, and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. Acm Sigact News 33, no. 2 (2002): 51-59. https://users.ece.cmu.edu/~adrian/731-sp04/readings/GL-cap.pdf
19. Vitalik Buterin. On Stake. 201407. https://blog.ethereum.org/2014/07/05/stake/

20. Vitalik Buterin. Long-Range Attacks: The Serious Problem With Adaptive Proof of Work. 201405. https://blog.ethereum.org/2014/05/15/long-range-attacks-the-serious-problem-with-adaptive-proof-of-work/
21. Vitalik Buterin. Proof of Stake: How I Learned to Love Weak Subjectivity. 201411. https://blog.ethereum.org/2014/11/25/proof-stake-learned-love-weak-subjectivity/
22. Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In Annual International Cryptology Conference, pp. 757-788. Springer, Cham, 2018. https://eprint.iacr.org/2018/601.pdf
23. Benjamin Wesolowski. Efficient verifiable delay functions. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 379-407. Springer, Cham, 2019. https://eprint.iacr.org/2018/623.pdf
24. Krzysztof Pietrzak. Simple verifiable delay functions. In 10th Innovations in Theoretical Computer Science Conference (ITCS 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. https://eprint.iacr.org/2018/627.pdf
25. Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. Continuous Verifiable Delay Functions. 201706. https://eprint.iacr.org/2019/619.pdf
26. Johnson Lau. BIP114: Merelized Abstract Syntax Tree. 201604. https://github.com/bitcoin/bips/blob/master/bip-0114.mediawiki
27. Pieter Wuille. BIP draft: Schnorr Signatures for secp256k1. https://github.com/sipa/bips/blob/bip-schnorr/bip-schnorr.mediawiki
28. Gregory Maxwell. [bitcoin-dev] Taproot: Privacy preserving switchable scripting. 201801. https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2018-January/015614.html
29. Gregory Maxwell. [bitcoin-dev] Graftroot: Private and efficient surrogate scripts under the taproot assumption. 201801. https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2018-February/015700.html
30. Pieter Wuille. BIP32: Hierarchical Deterministic Wallets. 201202. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
31. Sean Bowe. BLS12-381: New zk-SNARK Elliptic Curve Construction. 201703. https://electriccoin.co/blog/new-snark-curve/
32. Simon Josefsson, and Ilari Liusvaara. RFC8032. Edwards-Curve Digital Signature Algorithm (EdDSA). 201701. https://tools.ietf.org/html/rfc8032
33. Gabriel Campana, and Jean-Baptiste Bédrune. Everybody be Cool, This is a Robbery! https://www.blackhat.com/us-19/briefings/schedule/?hootPostID=db681a52c6a321681e1f9281b5124457#everybody-be-cool-this-is-a-robbery-16233
34. Graham Steel. How Ledger Hacked an HSM. 201906. https://cryptosense.com/blog/how-ledger-hacked-an-hsm/
35. Eyal Hertzog, Guy Benartzi, and Galia Benartzi. Bancor Protocol: Continuous Liquidity for Cryptographic Tokens through their Smart Contracts. 201803. https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf
36. Uniswap Whitepaper. https://hackmd.io/C-DvwDSfSxuh-Gd4WKE_ig
37. Yi Zhang, Xiaohong Chen, and Daejun Park. Formal Specification of Constant Product (x × y = k) Market Maker Model and Implementation. 201810. https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf
38. Dash - Dash is Digital Cash You Can Spend Anywhere. https://www.dash.org/

39. Gregory Maxwell. CoinJoin: Bitcoin privacy for the real world. 201308. https://bitcointalk.org/index.php?topic=279249.0
40. Monero - secure, private, untraceable. https://www.getmonero.org/
41. Gregory Maxwell, and Andrew Poelstra. Borromean ring signatures. 201506. https://pdfs.semanticscholar.org/4160/470c7f6cf05ffc81a98e8fd67fb0c84836ea.pdf
42. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In 2018 IEEE Symposium on Security and Privacy (SP), pp. 315-334. IEEE, 2018. https://eprint.iacr.org/2017/1066.pdf
43. Shen Noether. Ring Confidential Transactions for Monero. IACR Cryptology ePrint Archive, 2015, p.1098. https://eprint.iacr.org/2015/1098.pdf
44. Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash Protocol Specification. Version 2019.0.2 [Overwinter+Sapling]. 201906. https://raw.githubusercontent.com/zcash/zips/master/protocol/protocol.pdf
45. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von Neumann architecture. In 23rd USENIX Security Symposium (USENIX Security 14), pp. 781-796. 2014. https://eprint.iacr.org/2013/879.pdf
46. Grin. https://grin-tech.org/
47. BEAM: Mimblewimble-based Privacy Coin. https://www.beam.mw/
48. Tom Elvis Jedusor. Mimblewimble. (2016). https://github.com/mimblewimble/docs/wiki/MimbleWimble-Origin
49. Christian Reitwiessner. An Update on Integrating Zcash on Ethereum (ZoE). 201701. https://blog.ethereum.org/2017/01/19/update-integrating-zcash-ethereum/
50. Osuke. Announcing Zerochain: Applying zk-SNARKs to Substrate. 201903. https://medium.com/layerx/announcing-zerochain-5b08e158355d
51. Zerochain: A privacy-protecting blockchain on Substrate. https://github.com/LayerXcom/zero-chain
52. Jens Groth. On the size of pairing-based non-interactive arguments. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 305-326. Springer, Berlin, Heidelberg, 2016. https://eprint.iacr.org/2016/260.pdf
53. Gavin Wood. Polkadot: Vision for a Heterogeneous Multi-Chain Framework. Draft 1.
54. Substrate: The foundation for blockchain innovators. https://www.parity.io/substrate/
55. ElGamal, Taher. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory 31, no. 4 (1985): 469-472. https://link.springer.com/content/pdf/10.1007/3-540-39568-7_2.pdf
56. Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards Privacy in a Smart Contract World. 2019. https://eprint.iacr.org/2019/191.pdf
57. Benjamin E. Diamond. Anonymous Zether: Technical Report. 201905. https://github.com/jpmorganchase/anonymous-zether/blob/master/docs/AnonZether.pdf
58. Vitalik Buterin. Chain Interoperability. 201609. https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/5886800ecd0f68de303349b1/1485209617040/Chain+Interoperability.pdf
59. Joseph Poon, and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 201601. https://lightning.network/lightning-network-paper.pdf

Decentralized public chain ecosystem

Born for financial liberalization