



zkTube

# zkTube

# WHITEPAPER

Simple Solutions by Complex Connections

2021.07



website: [zktube.io](https://zktube.io)

# Abstract

In this document, we will describe the relevant description of the overall design framework of zkTube, a Layer 2 network protocol based on the Ethereum, including function introduction, technical operation principle, operation mechanism, usage protocol, incentive mechanism, application and development plan, etc. We expect that zkTube will play a role of strong practicability and applicability in Layer 2, supplementing and strengthening the actual application functions of the existing public chain and the entire blockchain by providing fast, safe, reliable and high-performance services.

Because of the limitation (For more information and updates on zkTube, please visit [zktube.io](https://zktube.io)). This document focuses on zkTube's innovative technical architecture and some of its unique features, which are important to achieving zkTube's goals.

zkTube will be an efficient Layer 2 blockchain operating network that can meet commercial standards and adopt cutting-edge design and advanced technology. It will focus on defining and providing the most basic, core, and services providers and make great improvements and innovations to them on the existing blockchain technology.

**Regarding the overall design of the zkTube project, we will make the following detailed descriptions in this file:**

# Catalogue

1. Project Background .....	5
2. Introduction: .....	6
2.1 zkTube Introduction: .....	6
2.2 Functional Overview: .....	8
3. Frame Design.....	9
3.1 Principle: .....	9
3.2. zkTube Operating Mechanism:.....	11
3.3 Three Types of Operations that a Trader Has on Layer 2: .....	14
3.4 Batch .....	15
3.5 Validity Proofs .....	17
3.6 Compression Mechanism .....	19
3.7. Batch Packaging and State Root Isolation .....	22
3.8. zkTube Technology Improvement (Based on PLONK Algorithm Optimization).....	22
3.9 zkTube Scaling Effect .....	24
4.0 zkTube Protocol Support .....	26
4.1 Deposit .....	26
4.2 Transfer.....	26
4.3 Withdraw.....	27
4.4 Buy .....	28
4.5 Sell .....	28
4.6 zkTube Scan .....	28
4.7 NFT .....	28
4.8. Cross Rollup.....	29
4.9 Create a Wallet.....	29
5. Economic Model.....	31
5.1 Economic Design .....	31
5.2 Consensus Design.....	33
5.3 Composition of Transaction Fees .....	34
5.4 Economic Benefits.....	35
5.5 On-Chain Governance .....	36
5.6 Community Autonomy .....	37
5.7 Mining Model.....	37
5.8 Release Rules.....	39
5.9 Miner's Admission Rules .....	39
5.10 Liquidity Calculation.....	41
5.11 ZKT Application .....	42
6. Equity Certificate .....	43
6.1 Equity Certificate Function.....	44
6.2 Value of Equity Certificate.....	44
7. ZKT Asset System.....	45
7.1 ZKT Circulation Mechanism.....	45
7.2 ZKT Repurchase and Destruction .....	47

8. PayTube Wallet.....	47
8.1. PayTube Wallet--A Cross-Platform Mobile Wallet.....	47
9. Team Members .....	48
10. Roadmap .....	49
11. Disclaimer.....	50

# 1. Project Background

Nowadays, the problems of Ethereum network congestion and high handling fees are becoming more and more serious. Solving the problem of network congestion and achieving scaling is the direction that many researchers have been working hard on.

A successful solution to the scaling problem of public blockchain is not only related to transaction throughput, it must also be achieved. The system meets the needs of millions of users without sacrificing decentralization. In the context of this era, technological breakthroughs brook no delay.

In the absence of technological breakthroughs, existing scaling solutions have to make major compromises to one or more of the above requirements. Fortunately, the latest developments in Zero-Knowledge proofs have opened up entirely new possibilities for solving this problem.

Today, zkTube Labs is honored to introduce to you our ace masterpiece: Ethereum's trustless scaling and economic applicability solution based on zkRollup, emphasizing the excellent experience of users and developers. The zkTube Protocol is designed to bring Visa-level throughput of thousands of transactions per second to Ethereum, while ensuring that funds are as safe as Layer 1 accounts and maintaining a high degree of censorship resistance.

zkTube is based on the concept of zkRollup. In short, zkRollup is a Layer 2 scaling solution, in which all funds are held by smart contracts on the main

chain, and calculations and storage are performed off-chain. Each Rollup block will generate a state transition Zero-Knowledge proof, which is verified by the main chain contract. SNARK contains proof of the validity of each transaction in the Rollup block. In addition, the public data update of each block is released as low-cost call data on the main chain network.

## **2. Introduction:**

### **2.1 zkTube Introduction:**

zkTube is a project built and developed on layer2 based on the PLONK algorithm protocol using ZK-Rollup technology. It can improve scalability by transferring batches of transactions to a single transaction. Its essence is to compress the user state on the chain and store it in a Merkle tree and transfer the user state change to the chain while ensuring the correctness of the user state change process under the chain through a Zero-Knowledge proof mechanism. The cost of directly processing user state changes on the chain is relatively high, but only using the smart contract on the chain to verify the correctness of a Zero-Knowledge proof is relatively low. In addition, the required remittance information is transmitted to the contract along with the certificate, which is convenient for users to verify accounts.

It needs to be emphasized that it is still submitted on the Ethereum chain, but part of the work on the chain is transferred to offline completion, so it will be as safe as Ethereum.

### **Lower Cost Per User Transfer:**

The transfer between off-chain tokens costs only 300–500 Gas, since each rollup operation splits the settlement cost of a single block evenly across the packing block from L2 to L1, reducing the Gas charge. The proof of updating the Merkle tree can be provided. Also, if the cost of transferring the single-chain L1 token is at least 20,000 Gas, we can reduce the Gas charge in the form of the Rollup.

### **Throughput and Scalability**

Each transaction contains less data, thereby improving the throughput and scalability of L2. You can directly use the L2 account balance to transfer directly, and you do not need to wait for the confirmation at the L1 layer to arrive. In this process, users can freely use their balance, but in the end, it is necessary to complete the final certainty at the L1 through a Zero-Knowledge proof mechanism.

### **Ethereum Mainnet Level Security Guarantee**

In the zkTube network, a large number of computations and data storage operations are placed in L2 for processing, after which a large number of transactions are aggregated and packed into the same block, and a

Zero-Knowledge proof is generated and sent to L1 for uniform verification, which can improve the transaction processing speed of the whole network and ensure security.

### **The Ecological Value to ETH**

By using the underlying protocol of zkTube, the ETH Gas charge is reduced, and user transactions at Layer 2 are more frequent than previously at Layer 1, thus solving the problem of ETH congestion. In addition, applications that require high TPS or related functions have been solved on zkTube Layer 2. when the Mainnet is launched, zkTube will provide the underlying protocol support services to users and applications vendors based on the zkTube Protocol, , which will enable better migration of applications and contracts on Layer 1. This is what zkTube desperately needs, the idea is to build an ecosystem based on ETH Layer 2.

## **2.2 Functional Overview:**

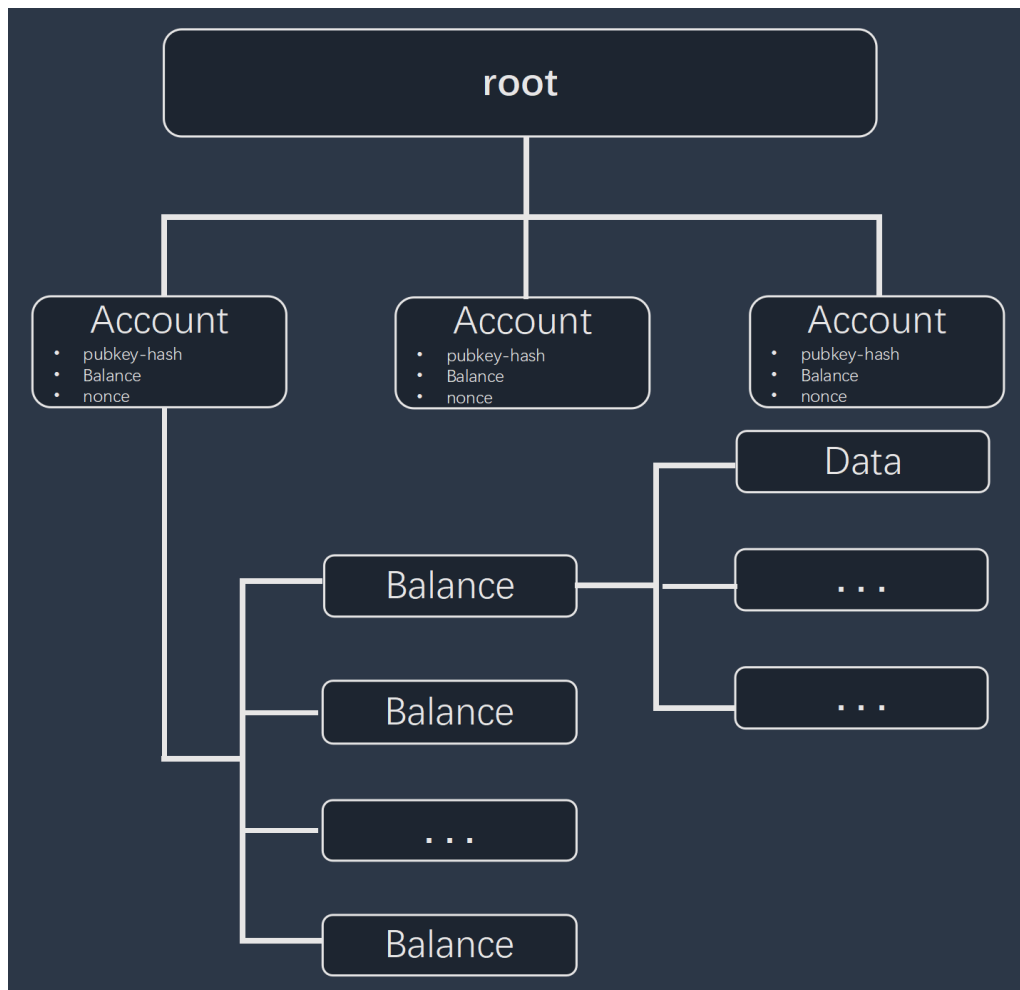
- zkTube supports mutual transfer from Ethereum L1 and L2, allowing users to receive funds without creating a wallet.
- Supports transfers to Ethereum address or Ethereum contract address (contract address supports L2 zkTube Protocol).



- For the transfer of L2 or withdraw to the account of L1, the processing fee is temporarily supported by ETH. After the mainnet is launched, only the native token ZKT of the zkTube Protocol is supported for payment.
- The zkTube Protocol supports all swap transactions in the ETH ecosystem on both L1 and L2.
- The zkTube Protocol supports purchases in fiat currency directly to the L1 account or the L2 account.
- The zkTube Protocol supports direct sales of fiat currency to L1 or L2 accounts.
- Both the L1 account and the L2 account support linking to NFT applications.
- Support for the Polkadot cross-chain protocol.
- Support Solidity language for developing smart contracts.
- Support ERC721

## **3. Frame Design**

### **3.1 Principle:**



The first account with account ID being 0 is used for depositing the storage cost until it is withdrawn to L1.

Currently  $2^{32}$  L2 accounts and  $2^{11}$  tokens are supported. Each L2 account has a unique number, starting from zero. The default value 0 is the verifier account. The account token contains the following information (PubKeyHash is the public key information of L2 account):

The random number of each account (other than the random number of each token owned by the user) can be used to order the request under the chain at the account level desired by the user.

Transfer means changing the token balance of two accounts in the Merkle Tree.

Because the balance is stored in its subtree, only the small subtree needs to be updated twice. The account is only updated once.

The L2 state is constituted by two parts: Account Root and Token Leaf Node under the account.

### Account

Name	Type	Size(bytes)	Comments
PubKeyHash	-	20	L2 Address
Address	-	20	L1 Address
Nonce	Uint32	4	Nonce of L2 Transaction
Balance Tree Root	-	Fr	The Root of the Token Subtree

### Information on Each Token Node

Name	Type	Size(bytes)	Comments
Balance	Uint32	4	Token Balance

## 3.2. zkTube Operating Mechanism:

The zkTube system covers two kinds of roles: Ordinary User and System Role.

### Ordinary User

Ordinary User refers to the account corresponding to Layer 2 of zkTube. The user constructs a transfer transaction and signs it with a private key, then collects the transaction in the Pool, and submits it to the first floor by the ETH sender.

## **System Role**

ZK Rollup=Rollup technology + Zero-Knowledge proof. The core logic design of zkTube is to realize the interaction between L2 data and contract through the rollup and zero-knowledge proof PLONK algorithm. The main design

breakthroughs are manifested in the collaboration between different programs:

The Watcher in the zkTube design is responsible for monitoring L1 and L2 transactions and adding the monitored to the Pool, then Block Proposer selects three dimensions of time, number and data size, to package the transactions in the Pool.

The packaged transaction is updated by the State Keeper and pending to the BlockCommitter. At this time, the Block Committer stores the block information in the storage to obtain the proof.

The zkTube Node extracts the Block from Storage and generates Witness (information required for proof), and stores it in Storage again.

Block Committer obtains the updated state and pubdata, submits the pubdata to the chain, and the transaction enters the Committed state. At the same time, Block Committer needs to prove the latest state. The proof process is calculated by zkTube Prover.

zkTube Prover checks the information that needs to be certified from the storage, generates a Zero-Knowledge proof, and then stores the proof in the Storage. Block Committer gets the proof and sends it to L1 through the sender, and finally proves that the transaction with no issues entering the Verified state.

## **Incentive Mechanisms**

To ensure the timeliness, stability, and security of the network, also increase the enthusiasm of zkTube Prover, zkTube has designed a complex algorithm mechanism to ensure that Prover does not do evil. Frequency of Prover submission of tasks, submission duration, and mortgage token are weighted to judge, and the qualified Prover can get ZKT as a reward through the zkTube reward mechanism.

## **Prover Adopts Decentralized Design**

The main function of Prover is to generate Zero-Knowledge proof data, and the user's asset data signature is managed by the user's wallet (such as MetaMask). The contract and service of zkTube are responsible for the transfer and storage of data, and ultimately will not affect the user's assets. Originally, zkTube or some organization could do the prover, but zkTube uses decentralization for the following purposes:

1. zkTube allows as many users as possible to provide professional mining machines to participate in the provision of the prover, avoiding the risks caused by a monopoly;
2. Prevent various authority institutions and avoid single-point risks;
3. Prevent the behavior of an organization or a combination of organizations manipulating zkTube;
4. When there is a task to receive, there will be CPU consumption.

### 3.3 Three Types of Operations that a Trader Has on Layer 2:

#### Sign Up

To register, users must provide a Merkle tree branch showing some index  $i$ , where  $i=0$  and  $A[i]=0$  or  $i > 0$  and  $A[i] = 0$  and  $A[i-1] \neq 0$ . The Merkle tree is updated so that  $A[i]$  is now equal to the address of msg. sender and the Merkle tree branch is recorded so that the client can read the log to get all the data needed to create its Merkle tree branch.

#### Deposit/Withdraw/Transfer

To deposit or withdraw, users need to provide a Merkle tree branch, which shows some index  $h$  (where  $A[h]$  is equal to the address of msg. sender) and the corresponding branch  $B[h]$ , and they want to deposit or withdraw/ the transfer amount  $m$  (negative for withdrawal). The contract checks this  $B[h][0] + m \geq 0$ . If  $m > 0$ , it verifies (if the system is used for ETH)  $\text{msg. value} == m * 10^{12}$  (that is, the basic unit of the system is  $10^{-6}$  ETH), otherwise it will call the appropriate ERC20 contract. If so, it sends the ETH or token to the contract address. Then, let its smart contract generate and update the Merkle tree root. Please note that to improve efficiency, the registration and deposit steps of traders who have not yet registered can be combined.  $\text{transferFrom}(\text{msg.sender}, \text{self}, m * 10^{12})$   
 $< 0$   
 $\text{msg.senderB}[i][0] += m$

#### Send

To send, the user constructs the data: From Address to To Address (By index 3 bytes), number (Represented by a power of 10 in scientific notation, the general number of bytes is  $\leq 4$ ), gas cost (0-0.5 bytes), random number (2 bytes). User broadcasting (From, To, Amount , Fee , Nonce) and add a Signature.

Prover can aggregate many transactions in the pool and create ZK to use the Plonk protocol to prove that when all operations are processed in order, at the beginning of each operation  $B[\text{from}][0] \geq \text{amount} + \text{fee}$ ,  $B[\text{from}][1] == \text{nonce}$  and from the known valid signature  $A[\text{from}]$ , then update the Merkle root to have  $B[\text{from}][0] -= \text{amount} + \text{fee}$ ,  $B[\text{to}][0] += \text{amount}$ ,  $B[\text{relayer}][0] += \text{fee}$ ,  $B[\text{from}][1] += 1$ . A log is issued to remind users that the transaction is an unverified payment transaction, and they will need to recalculate their Merkle tree witnesses.

### 3.4 Batch

When the merchant receives the transaction, he must "execute" it. The so-called execution, in essence, is to change the state of the relevant account, and STF is a function to change the state of the account. STF is an abbreviation for the state transition function.

The state refers to the state machine, each state machine has a state at a time.

We can assume that the initial state is a state machine  $S[0]$ . When an action  $T[1]$  acts on the state machine, the state machine of the occurrence of migration.

We can use the following equation to represent the migration process.

The three states of the transaction: initiated, pending, completed (Verified means that the block has been validated on Layer 1).

$$S[1] = STF(S[0], T[1])$$

Here  $S[0]$  is the initial state,  $S[1]$  is an execution state of the state machine after-action  $T[1]$ . Then several new actions  $T[2], T[3], \dots, T[n]$  continue to act on the state machine, the state machine migration is sequential.

$$S[2] = STF(S[1], T[2])$$

$$S[3] = STF(S[2], T[3])$$

...

$$S[n] = STF[S[n-1], T[n]]$$

Briefly, we can also combine  $T[1], T[2] \dots, T[n]$  as a whole, the state transfer process can be simplified as:

$$S[n] = STF(S[0], T[1], T[2], \dots, T[n])$$

More generally, suppose the current state of the state machine is  $PRE\_STATE$ , then there are  $n$  Actions  $T[1], T[2], \dots, T[n]$  that are sequentially applied to the state machine, then the state machine is  $POST\_STATE$ , this can be expressed as:

$$POST\_STATE = STF(PRE\_STATE, T[1], T[2], \dots, T[n])$$



If the above Action is replaced by a transfer transaction, the set of accounts in the system is treated as a state machine, then the entire process is the on-chain transaction execution. The execution of the transaction changes the global state of the whole chain. The global state on the chain is also the state set of each account, which is formed into a Merkle tree. The leaf node of the tree is the account state, and the root of the tree can be directly used to represent the state set. Therefore, the above PRE\_STATE and POST\_STATE are the roots of the global account status tree.

After each batch, it needs to be submitted to L1. In order to ensure security and match the contract status root of L1, Zero-Knowledge proof verification of the batch at the time of submission is required.

### **3.5 Validity Proofs**

The account information of all users is maintained in a Merkle tree. The root of the Merkle tree is recorded in a smart contract on the chain, the root of this value also represents the current state of all accounts across the system. When a user initiates a remittance transaction, this state changes. However, the changes must be made by the rules.

- First, we must ensure the legitimacy of the transaction.
- Whether there is enough money in the remittance account to pay the remittance amount and the processing fee.

- Is the nonce of the remittance account correct?
- Is the signature of the remittance transaction correct?
- Then, the corresponding role in the system executes the transfer transaction, modifies the leaf nodes of the transfer-out account and transfer-in account in the Merkle tree, and then recalculates the root of the new Merkle tree.
- Repeating the second step, Prover will process multiple transactions at one time in the sequence, and then submit the root of the finally calculated Merkle tree as a new state to the on-chain contract. However, to prevent fraud in the batch submitted by Prover, it is required to verify and generate a signature before submission. The proof is as follows:

After the Pool has collected a series of transactions, it needs to use the pre-defined ZK PLONK protocol to generate a PROOF:

- Make sure that the nonce, value, charge in each transaction  $T[1], T[2], \dots, T[n]$  are all correct and the signature is correct.
- Make sure there is no problem with the state transition, i.e.  $STF(PRE\_STATE, T[1], T[2], \dots, T[n]) = POST\_STATE$
- Then submit this PROOF along with  $POST\_STATE, t[1], t[2], \dots, t[n]$  to the chain contract. Among them,  $t[1], t[2], \dots, t[n]$  are simplified information of

the transaction, without nonce and signature. Therefore,  $t[i]$  is smaller than  $T[i]$ .

And then the smart contract just verifies that the PROOF is correct. If the PROOF is correct and the state stored in the contract is replaced by `PRE_STATE`, then the new state `POST_STATE` is added to the contract and replaces the state. Since Prover must generate the PROOF of the ZK PLONK protocol before submitting to the contract, if the Prover modifies the user's transaction maliciously, the PROOF will not be verified.

In addition, since the transactions  $t[1]$ ,  $t[2]$ , ...,  $t[n]$  submitted to the chain do not contain nonce and signature, the data on the chain will become smaller (In the above examples, only 11 bytes will be chained per transaction).

At this time, Prover has been unable to modify the user's transaction due to certification restrictions. However, a malicious Prover can still refuse to serve a transactor. To prevent this behavior, the contract supports on-chain withdrawal, that is, any transactor can withdraw its token from the chain.

### **3.6 Compression Mechanism**

In terms of compression, zkTube adopts the compression principle ZK Rollup, making the file smaller.

For example, a simple Ethereum transaction (sending ETH) is about 110 bytes in size. The ETH transfer on zkTube is only about 12 bytes in size:

Parametric	Ethereum	zkTube
Nonce	~3	0
Gas Price	~8	0~0.5
Gas	3	0~0.5
To	21	4
Value	~9	~3
Signature	~68 (2+33+33)	~0.5
From	0 (Recover from signature)	4
Total	~112	~12

**Nonce:** The main purpose of this parameter is to prevent replay attacks. If the current account of the random number is 5, then the next transaction for the account must contain 5 random numbers, but the transaction has been processed, the random number in the account is increased to 6, the transaction cannot be reprocessed. In zkTube, we can eliminate the random number, because we can directly restore the previous state of the random number, if someone tries to use a random number to replay a previous transaction, the signature can not be verified, because the signature is checked against data that contains a high random number.

**Gas Price:** Users pay a fixed gas price range, it is billed according to 14 times a power of 2. It will be adjusted according to the price of Ethereum. Of course,

users can customize the adjustment according to the range between the minimum and maximum.

**Gas:** Gas form is set as a power of 2, which is set by zkTube.

**To:** You can replace a 20-byte address with an index (For example, if an address is added to the tree Merkel addresses of 4527, we simply use the index 4527 to represent it and then add a "subtree" to store the mapping between the index and the address itself).

**Value:** Use scientific notation to store the value. The number of bits supported by each currency is different, and the number of bytes ranges from 0 to 0.5.

**Signature:** Use the BLS aggregation signature to aggregate a large number of signatures into about 32-96 bytes and complete the ZK PLONK signature. The aggregate signature can be checked at one time based on the message set and batch sender set. The "~0.5" in the table indicates that there is a limit to the number of signatures that can be included in an aggregate signature.

**From:** Replacing a 20-byte address with index works the same way as To.

**Total:** We can use a scientific approach to store multiple values, the same with the Value above.

### **3.7. Batch Packaging and State Root Isolation**

zkTube adopts independent batch packaging, separating the batch of committed L2 transactions from the process of submitting state root:

Unlike in the past, zkTube separates batches, sorts them according to time, and then certifies, validates them and updates their status when they are submitted to the Ethereum L1, so that operators can commit multiple batches at once, and multiple operators can commit different batches simultaneously. The advantages are :

- Enables multiple batches to be released simultaneously to improve audit resistance, while avoiding the problem of some batches being packaged first and others becoming ineffective;
- If a state root is invalid, instead of rolling back the whole batch, we can just roll back the state root and wait for someone else to provide a new state root for the batch. This ensures that the transaction from the sender will not be rolled back

### **3.8. zkTube Technology Improvement (Based on PLONK Algorithm Optimization)**

zkTube uses the Zero-Knowledge proof PLONK algorithm in Layer 2. Theoretically, the STARKS algorithm is the most secure. It does not rely on the assumption of pairing and exponential knowledge but is completely based on hash value and information theory to calculate, which is an anti-quantum computer attack. Correspondingly, it has increased the number of proof bytes, from the original 288 bytes (b) to several kilobytes (kb), which is not suitable for zkTube to build a general Layer 2 protocol. Secondly, one of the biggest problems with STARKs is that it is not universal. They require different arithmetic solutions for different problems or scenarios. It is not practical for zkTube at present. Maybe it will be a good choice for Layer 2 sharding technology when Ethereum 2.0 launches in the future.

SNARKs is the algorithm that uses the least number of bytes in the algorithm, and Groth16 is the one that uses the most. First of all, Groth16 is non-general, and it relies on one-time non-upgradable settings. If the system changes or encounters any small bug, new rituals are needed to deploy and fix it. Secondly, different CRS (the Common Reference String) is needed for different problems, which is equivalent to designing different circuits for different scenes. Therefore, this algorithm is used for specific scenes, such as DEX, payment and other simple scene circuit designs.

1. The advantage of PLONK is that it supports universal and upgradeable reference strings, and as long as the size of the circuit design does not exceed the upper limit of the SRS threshold, some scenes and functions can share the same SRS, which is very helpful to zkTube. zkTube utilizes this feature to maximize this feature in specific scenarios such as deposit, withdraw, transfer, buy, and sell. Originally, its proof time was shortened by about 5 times compared with SNARKs, but after the optimization of zkTube, the proof time was shortened About 15-20 times.

2. To maximize the CPU, we optimize from the two points of reducing the occupied memory and reasonably allocating memory. In the circuit design, we try our best to meet a general SRS, so in the Merkel tree, we divided SRS into different groups and proves. Therefore, in the Merkel tree, we group different SRSs and prove it to introduce repeated data calculations to reduce memory usage as much as possible. At the same time, we used a monitoring mechanism to adjust the allocation of memory. For example, if a certain circuit was too high frequency during this period. A special thread is used to process the pre-stored scheme to achieve the effect of allocating memory.

### **3.9 zkTube Scaling Effect**



On the existing Ethereum chain, the upper limit of gas is 12.5 million. In a transaction, each byte of data costs 16 gas. This means that if a block contains only one batch (we say that it is equivalent to packaging a ZK Rollup and spending 500,000 gas on proof verification), that batch can contain (12 million / 16) 750,000 bytes of data. As shown above, for an Ethereum transfer Rollup, each user operation only needs 12 bytes, which means that the batch of transactions can contain up to 62,500 transactions. Now the average block time is 13 seconds, which is equivalent to about 4807 transactions per second (compared to the current direct transfer on Ethereum is 12.5 million / 21000 / 13 ≈ 45 transactions per second).

The following table is an example for another application example:

Application	Bytes occupied by Rollup	L1gas	The highest level of capacity expansion
Eth transfer	12	21.000	105X
ERC20 transfer	16	~50.000	187X
Uniswap	~14	~100.000	428X
Optimistic	296	~380.000	77X
zkRollup	40	~380.000	570X
zkTube	42	~380.000	617X

## 4.0 zkTube Protocol Support

### 4.1 Deposit

If the ETH is stored in a zkTube Layer 2 account, the object created when the wallet is created must have access to the Ethereum signer.

If the ERC-20 token is stored, the transfer of the token must be approved through the wallet first, so that the contract can transfer it to the Layer 2 account. In this process, users can unlock ERC20 tokens to obtain permanent approval for ERC20 deposits.

After submitting the operation to the Ethereum network, we must wait for a certain number of confirmations before we can accept it in the zkTube Layer 2 network. After submitting the transaction to the zkTube network, the recipient can already use the funds. If Prover does not process the deposit after a few seconds, the user can directly withdraw the deposit amount + ETH fee from the contract.

### 4.2 Transfer

When the L2 account is transferring or withdrawing, the user needs to sign and associate the account with the private key of zkTube, which is the so-called unlocked account in the UI form.

"To the address" can be a zkTube account or an uncreated Ethereum address. If it is an Ethereum address, the system will create an account on zkTube based

on the "To address". At this time, the user only needs to use this address to connect to zkTube Layer 2 to check the balance.

After confirming the above operations, before sending the transaction, the user will be required to use their Ethereum account to sign a specific message with transaction details. During the transfer process, Prover needs to be paid a certain certification fee, any currency supported by zkTube can be used as a handling fee. After the mainnet is launched, only ZKT will be supported for payment.

### **4.3 Withdraw**

When the L2 account is transferring or withdrawing, the user needs to sign and associate the account with the private key of zkTube, which is the so-called unlocked account in the UI form.

Since zkTube L1 account and L2 account correspond one-to-one and share a private key, the account address is consistent, which supports transferring to this address as well as other ETH addresses. It is important to note that whether transferring between L2 accounts or directly withdrawing the balance from L2 to L1, the "To address" supports the Ethereum address and once transferred to some contract address, it will not be retrieved unless the contract address supports the relevant protocol.

After confirming the above operations, before sending the transaction, the user will be required to use their Ethereum account to sign a specific message with

transaction details. During the transfer process, Prover needs to be paid a certain certification fee, any currency supported by zkTube can be used as a handling fee. After the mainnet is launched, only ZKT will be supported for payment.

## **4.4 Buy**

The underlying protocol of zkTube supports the purchase of ETH and ERC20 tokens directly through fiat currency through third-party payment. At the protocol layer, we will support that the purchased tokens reach the Layer 1 account, and also support the Layer 2 account. The plan to reach the Layer 1 account is the same as the transfer between ETH, and the plan to reach the Layer 2 account is the same as the Deposit method.

## **4.5 Sell**

zkTube is working with some third-party payment companies to sell ETH and ERC20 directly through fiat currency. At the protocol layer, we will support the token sold to be deducted from the Layer 1 account and also support the Layer 2 account, depending on the seller. The method of deduction from the Layer 1 account is the same as the transfer method between ETH, and the method of deduction from the Layer2 account is the same as the Withdraw method.

## **4.6 zkTube Scan**

zkTube has also developed its web3.0 browser, and currently supports the main network and test network (temporarily supporting Ropsten and Rinkeby) Deposit, Withdraw, Transfer, Buy, Sell related records queries.

## **4.7 NFT**

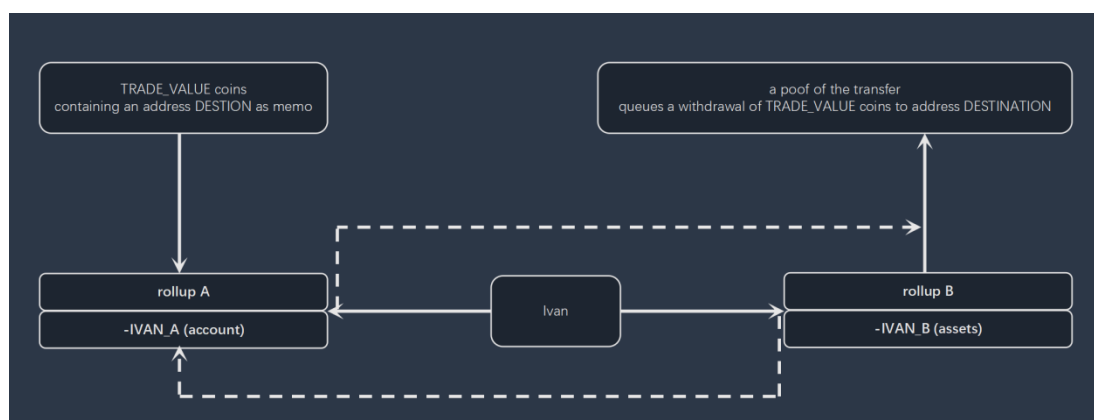
The underlying protocol of zkTube has also been extended in the NFT field to support NFT transactions on L2, and at the same time, the main bytes are submitted to L1 as proof through the zkPLONK protocol. In addition, zkTube will support a large

number of Dapps, including games that will be displayed in the PayTube wallet, providing the underlying protocol for these Dapps. Regardless of the scene or the protocol layer, it will bring great value to the game project, and there are many scene applications.

## 4.8. Cross Rollup

zkTube implements the data interaction between Rollup A and Rollup B through the smart contract on Ivan and Rollup A. When IVAN\_A receives a transaction that needs to be processed, it will use TRADE\_VALUE as a note to include the address destination. After a certain period, this task will be sent to IBAN\_B, IBAN\_B will queue up the withdrawal of TRADE\_VALUE tokens to the address destination.

When Ivan discovers that he has received the money in Ivan\_A, he can personally send TRADE\_VALUE\* (1-fee) tokens to DESTINATION. He can use the IVAN\_B method to send a transaction to complete the above operation. This method saves a record to prevent the automatic sending clause in the contract from triggering the transaction.



## 4.9 Create a Wallet

Ways to create a Layer 2 account:

Successfully create a wallet in zkTube, and a corresponding L2 account will be generated.

Successfully use a private key to import a wallet in zkTube, and a corresponding L2 account will be generated.

Every transaction involves Ethereum signature and permission, so we have done special processing for the signature. As long as the user creates a transaction in the zkTube network wallet, he must have a zkTube key pair associated with it. The zkTube keys are processed by the Signer object. These keys can be created by deriving them from the Ethereum signature of a specific message. If the user does not provide Signer and the key is created using another method, this method is used by default.

In order to make the zkTube key valid, the user should use the set signature key to sign or authorize once in the zkTube network.

- Signature

In the zkTube network, Signer is used to signing authorization during the process of creating a wallet, authorizing ERC20 transfer, Transfer, Withdraw, etc.

Among them, transactions such as Transfer and Withdraw are signed. The purpose of this signature is to provide higher security when the zkTube key of the wallet is stolen. The user is required to sign the transaction description and perform a signature check when submitting the transaction to zkTube.

- Supported Signature Types

Ethereum Signature

Support ERC1271 smart contract protocol

## **5. Economic Model**

### **5.1 Economic Design**

zkTube's economic design is to keep the interests of all participants in the same direction as the value growth of the zkTube agreement. On the one hand, it must protect the interests of all participants, and on the other hand, it must also maintain the stability of the zkTube system. That is to say, all participating parties contribute to the zkTube system while pursuing their interests.

To achieve our economic design goals, we must think from several aspects:

- How to ensure the security of the zkTube protocol;
- How to maintain the sustainable development of the zkTube system;
- How to protect the interests of participants;
- How to maintain the interests of the participants and the value of the zkTube system are in the same direction.

**Before designing the zkTube economic model, let's analyze the existing distributed system model first:**

As the earliest blockchain protocol, Bitcoin uses native tokens to incentivize nodes to verify transactions and uses PoW consensus to coordinate competition between nodes. In the Bitcoin economic model, early block rewards are the major way to maintain the interests of nodes. After the block rewards are reduced in the later stages, fees become the main way to maintain the benefit of nodes.

Bitcoin is widely accepted as having two functions: Value storage and circulation payment. Value store users expect to hold tokens to maintain or increase in value. They are concerned about the security of the Bitcoin network protocol and the policy of currency deflation. Current payment users use the network's peer-to-peer transfer of value, similar to fiat currency payments, to focus on bitcoin's transaction costs and volatility in value.

Without changing the existing Bitcoin economic model, the interests of value store users can be guaranteed. In such a user-dominated network, there will not be many transactions, so in the long run, it is difficult for fees to maintain nodes and guarantee network security. This will affect the sustainable development of the whole system.



Ethereum is the largest smart contract platform. The native tokens are used to pay for computing services. Similar to Bitcoin, the service fee may become the main way to maintain the interests of the nodes after the block reward is reduced. The difference is that Ethereum has more transactional users, and its monetary policy is not fixed. It is now an inflation policy.

The planned ETH2.0 system changes the consensus of Ethereum to PoS, which is designed to protect the interests of nodes through perpetual inflation. Inflation will depreciate the value of the token. Its economic model will balance this relationship as much as possible.

After learning the economic models of Bitcoin and Ethereum, zkTube proposed the zkTube economic model and asset system based on its characteristics.

## **5.2 Consensus Design**

For a decentralized system, a consensus is particularly important for the system. Different mining mechanisms have slightly different consensus mechanism designs, but their functions are the same. They all have corresponding governance tokens to coordinate the entire system to operate according to certain rules. The layer 2 network also draws on the Bitcoin mining mechanism

to design consensus. First of all, ZKT participated in the incentive and punishment measures of the consensus protocol to ensure the enthusiasm of miners to submit data and anti-cheating methods. Secondly, ZKT uses its equity certificate attributes as the hub of connection protocols and protocol-based applications, such as Defi, NFT and other scenarios. Finally, ZKT is just like BTC, follows the principles of value growth and quantity deflation in the financial market economy, and can adapt and adjust the market itself.

### 5.3 Composition of Transaction Fees

In the zkTube network, the following fees are mainly incurred:

**Deposit** -- the ETH transaction fee generated by transferring money from an Ethereum Layer 1 account to a zkTube Layer 2 account, which is charged by Ethereum miners;

**Transfer** -- The transaction fee for zkTube L2 transfer is almost negligible. This fee can be paid in any token supported by the ZKTube platform. The fee is used for L2 miner packaging and zero-knowledge proof verification;

**Withdraw** -- there will be a transaction fee from the zkTube Layer 2 account to the Ethereum Layer 1 account, which can be paid with any tokens supported by

the zkTube platform. The fee is used for the packing of the Layer 2 miners and the verification of the zero-knowledge proof;

Change pubkey will consume a small amount of ETH, which will be collected by Ethereum miners;

The zkTube protocol will use ZKT to offer low or free service on DeFi, NFT and various SWAP services.

## **5.4 Economic Benefits**

The benefits generated by the zkTube agreement are divided into internal and external:

### **Internal Benefits:**

First of all, compared with the Ethereum chain, transactions on zkTube have increased TPS and reduced transaction fees. TPS has increased from 14 to 3000+, and the fees have been reduced by about 100 times, improving overall transaction efficiency. This makes the transaction efficiency close to or even surpassing some centralized services; Secondly, zkTube supports the purchase and sale of fiat currencies from all over the world at the layer2, which further improves the circulation efficiency between fiat currencies and various tokens ;

Finally, there may be a small number of fees in circulation, which can be paid with any token.

### **External Benefits:**

The underlying protocol of zkTube supports all kinds of applications including the applications of offline entity business attributes, which have been opened up with the Layer 2 accounts. The application experience and centralization can be consistent. zkTube has confirmed the cooperation plan with the decentralized Dex Uniswap in the DeFi field. zkTube will provide agreement services to Uniswap to set up Layer 2 swap transactions, 3000+ TPS and very low transaction fees, which can ensure that the performance of various swap transactions will not be affected, coupled with negligible transaction fees, will make swap replace centralized exchanges; however, in other areas of DeFi, such as decentralized insurance, decentralized mortgage lending and other fields, it also brings at least a 30 times increase in efficiency; such as NFT collections, vouchers and games will all use the zkTube protocol to build applications quickly.

## **5.5 On-Chain Governance**

zkTube on-chain governance updates and upgrades the protocol through smart contracts. On-chain governance is governed by node voting, punishment, and elimination mechanisms, allowing all ZKT holders to participate in the network construction, and the on-chain governance mechanism is still under development.

## **5.6 Community Autonomy**

To reward early investors, ZKT will give some investors the right to govern the zkTube agreement, and they can participate in the decision-making of zkTube as important external members. The investors who have invested certain funds in the early stage can obtain the same proportion of the allocation amount following the proportion originally held by the investors during the allotment process.

## **5.7 Mining Model**

$$R = \beta * D_n + \mu$$

R : Miner's income per task

$\beta$  : Bonus base permission

$D_n$  : Degree of difficulty, Difficulty levels are  $D_1$   $D_2$   $D_3$   $D_4$   $D_5$   $D_6$

$\mu$ : Miner's increasing coefficient

### ● Rules Description

1. When the number of miners is 0-5000 ,

$\beta = 5$  ,  $\mu = 0$  ;

2. When the number of miners is 5001-20000 ,

$\beta = 5$  ,  $\mu = (\text{Current number of miners})/5000$  ;

3. When the number of miners is 20001-50000 ,

$\beta = 8$  ,  $\mu = (\text{Current number of miners})/20000$  ;

4. When the number of miners is 50001-100000 ,

$\beta = 15$  ,  $\mu = (\text{Current number of miners})/50000$  ;

5. When the number of miners is 100000 ,

$\beta = 20$  (Reaching the extreme),  $\mu = (\text{Current number of miners})/100000$ .

### ● Data Model

### Calculate the relationship between the number of mining bonuses earned per mission and the number of miners

Number of Miners	Calculate the number of mining bonuses earned per mission.[The difficult level is 6]	Number of tasks generated in 24 hours	The daily output of the single miner	Number of mining in 24 hours	Number of releases in 24 hours	Number of mining in one month	Number of releases in one month	Number of mining in one year	Number of releases in one year
1000	5	1440	7.2	7200	2543.4	216000	82128.6	2592000	1545946.04
5000	5	1440	1.44	7200	2543.4	216000	82128.6	2592000	1545946.04
6000	6.2	1440	1.488	8928	3153.816	267840	107236.44	3214080	2681330.616
20000	9	1440	0.648	12960	4578.12	388800	155665.8	4665600	3892254.12
40000	10	1440	0.36	14400	5086.8	432000	172962	5184000	4324726.8
50000	10.5	1440	0.3024	15120	5341.14	453600	181610.1	5443200	4540963.14
90000	16.8	1440	0.2688	24192	8545.824	725760	290576.16	8709120	7265541.024
100000	17	1440	0.2448	24480	8647.56	734400	294035.4	8812800	7352035.56
200000	22	1440	0.1584	31680	11190.96	950400	380516.4	11404800	9514398.96
400000	24	1440	0.0864	34560	12208.32	1036800	415108.8	12441600	10379344.32
500000	25	1440	0.072	36000	12717	1080000	432405	12960000	10811817

## 5.8 Release Rules

35% of the rewards generated by each task will be released immediately,

65% will be released linearly, and the release cycle is 200 days:

$$SF_1 = SY_1 * 0.35 + SY_1 * \frac{0.65}{200}$$

$$SF_2 = SY_2 * 0.35 + (SY_1 + SY_2) * \frac{0.65}{200}$$

$$SF_{200} = SY_{200} * 0.35 + (SY_1 + SY_2 + \dots + SY_{200}) * \frac{0.65}{200}$$

$$SF_{201} = SY_{201} * 0.35 + (SY_2 + SY_3 + \dots + SY_{201}) * \frac{0.65}{200}$$

...

$$SF_N = SY_N * 0.35 + (SY_{N-199} + SY_{N-198} + \dots + SY_N) * \frac{0.65}{200}$$

## 5.9 Miner's Admission Rules

$$P=25-\omega$$

$P$  : Number of tickets for each miner

$\omega$  : Miner's increasing coefficient

- **Rules Description**

1. When the number of miners is 0-5000 ,  $\omega=0$  ;

2. When the number of miners is 5001-1000,  $\omega=(\text{Current number of miners})/5000$  ;

...

3. When the number of miners is greater or equal to 75000 ,  $P=10$  (Reaching the extreme) ;

4. When a miner withdraws from mining, he will be released at the end of one year based on the withdrawal time.

- **Data Model**



The relationship between the number of miners and the number of admissions tickets for each miner		
The number of miner	Admission tickets for each miner	Cumulative number of admission tickets
1000	25	25000
5000	25	125000
6000	23.8	142800
20000	21	420000
40000	17	680000
50000	15	750000
75000	10	750000
100000	10	1000000
200000	10	2000000
400000	10	4000000
500000	10	5000000

## 5.10 Liquidity Calculation

The data model of the number of miners and the annual release is as follows:

Number of miners	Number of releases in one year	Number of whitelist releases in the first year	Cumulative number of admission tickets	Actual annual circulation
1000	1545946.04	554400	25000	2075346.04
5000	1545946.04	554400	125000	1975346.04
6000	2681330.616	554400	142800	3092930.616
20000	3892254.12	554400	420000	4026654.12
40000	4324726.8	554400	680000	4199126.8
50000	4540963.14	554400	750000	4345363.14
90000	7265541.024	554400	750000	7069941.024
100000	7352035.56	554400	1000000	6906435.56
200000	9514398.96	554400	2000000	8068798.96
400000	10379344.32	554400	4000000	6933744.32
500000	10811817	554400	5000000	6366217

### Whitelist Release Rules:

Whitelist Release Rules: 1% for each of the first eight weeks, 2% per month starting from the third month (first year), and 3% per month for the second and third years.	
The Amount of ZKT for Whitelist	1980000
The first week	19800
The second week	19800
The third week	19800
The fourth week	19800
The fifth week	19800
The sixth week	19800
The seventh week	19800
The eighth week	19800
The third month	39600
The fourth month	39600
The fifth month	39600
The sixth month	39600
The seventh month	39600
The eighth month	39600
The ninth month	39600
The tenth month	39600
The eleventh month	39600
The twelfth month	39600
Release in the first year	554400
Release in the second year	712800
Release in the third year	712800

## 5.11 ZKT Application

As the native token of the zkTube protocol, ZKT represents the rights of the holder and also has practical use-value. ZKT can be used in the following scenarios :

### Governance Token

zkTube is a decentralized project led by the community. ZKT is the certificate of community participation in governance:

Users who hold a certain number of ZKT can initiate upgrade proposals, such as modifying the ZKT long-term incentive plan, etc.;

All ZKT holders can vote on the proposal, and the proposal that receives a majority vote will be approved and implemented by the development team.

### **Transaction Fees**

ZKT can be used as a transaction fee and fuel in the zkTube network.

### **Mining**

As the value carrier of zkTube, ZKT supports Staking, CPU mining and import wallet mining, and the income is ZKT.

### **Cryptocurrency Assets**

As the zkTube network grows, ZKT, as an Ethereum asset, its investment value will continue to rise with market demand.

### **Circulation in Defi and NFT**

ZKT will conduct extensive circulation in DeFi, participate in the swap and provide liquidity, also launch ZKT insurance and lending services on the insurance and lending platform.

## **6. Equity Certificate**

ZKT is a functional token that realizes the value of the zkTube network. And it has cross DeFi and NFT circulation performance, similar to ETH in the Ethereum network or DOT in the Polkadot network.

## **6.1 Equity Certificate Function**

In the zkTube network, ZKT mainly has the following functions:

- Maintain zkTube network consensus;
- As a transaction fee for using the network;
- As a transfer fee;
- Cross Defi circulation;
- Cross NFT circulation;
- Apply for nodes and maintain node security;
- It can be used for the election and voting of the governance mechanism on the chain, and to vote on the proposal.

## **6.2 Value of Equity Certificate**

ZKT is the only equity certificate of the zkTube protocol, and its value depends on the zkTube network. The value of ZKT is positively related to the scale of the

zkTube network. When the zkTube protocol is widely used, the demand and value of ZKT will rise accordingly.

ZKT can be used as fuel to pay network fees, which has use value and can be used as a dividend voucher, which can produce certain income and value by itself.

## 7. ZKT Asset System

### 7.1 ZKT Circulation Mechanism

The zkTube Protocol is a scaling solution for the Ethereum community, so most of the ZKT will be generated by mining and distributed to community participants who maintain the operation of the system.



**Total ZKT: 330,000,000 ZKT**

**Community Mining 89.4%**

295,020,000 ZKT, mined by the community.

**zkTube Foundation 5%**

16,500,000 ZKT, no release, only for emergency treatment, market value management and voting use, the proceeds will be destroyed according to the destruction mechanism.

**Technical Team 0.4%**

A total of 1,320,000ZKT will be locked for one year from the mainnet launch and distributed four times a year from the second year. The distribution will be completed in a total of five years.

**Community Operation 0.75%**

A total of 2,475,000ZKT will be locked for one year from the mainnet launch and distributed four times a year from the second year. The distribution will be completed in a total of five years.

**zkTube Node 3.85%**

A total of 12,705,000ZKT is used for node construction and maintenance.

**Investors 0.6%**

Total 1,980,000 ZKT.

## **7.2 ZKT Repurchase and Destruction**

The zkTube Official will use a certain percentage of transaction fees to repurchase ZKT aperiodically to stabilize the market value of ZKT. The ZKT obtained will be directly destroyed, and no individual or organization may use it for other purposes.

## **8. PayTube Wallet**

### **8.1. PayTube Wallet--A Cross-Platform Mobile Wallet**

PayTube Wallet is a Web3.0 application on the zkTube network. Users can access any Ethereum Dapp through a one-click link to the wallet. After the connection is established, users can buy/sell cryptocurrency, explore Dapp, staking, insurance, games and a series of operations.

PayTube Wallet is a non-custodial decentralized wallet. In the first version, users need to save their own private keys. (About the private key: The PayTube team is studying and exploring the safest way for users to solve the solution once the private key is lost, without affecting the nature of decentralization.)

PayTube Version 1.0 will provide the basic PC and mobile terminal functions for users, such as deposit, withdrawal, transfer, buy, sell, etc. At the same time, PayTube Wallet will provide iOS and Androids for users to download and use.

**PayTube Wallet will Support:**

1. Support deposit from L1 to L2;
2. Support the transfer between L2 and L2;
3. Support withdraw from L2 to L1;
4. Realize the data interaction between L1 and L2 and Dapps in the DeFi, such as DEX and NFT;
5. Seamless connection with zkTube Dapp;
6. Support buy and sell between L2 and more than 40 stable currencies including USDT, USDC, TUSD, GUSD, EURS, etc.

## **9. Team Members**

**[Team Members] :**

[https://drive.google.com/file/d/18pdl55HXRc4NW4MhTE5SjOQATuV\\_oZPV/view?usp=sharing](https://drive.google.com/file/d/18pdl55HXRc4NW4MhTE5SjOQATuV_oZPV/view?usp=sharing)



## 10. Roadmap

### Ecological Planning of zkTube

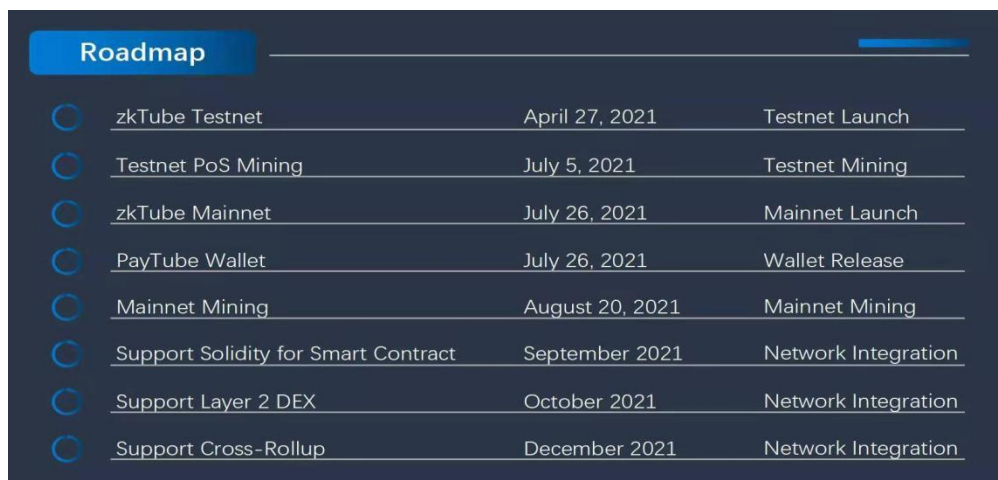
#### An Open-Source Payment Network

PayTube Wallet is based on the development of the zkTube protocol. Payment Wallet for all users of open source, through our cross-platform mobile wallet PayTube Wallet, had enough energy to provide users with convenient and cost-effective use of the Layer 2 payment environment.

#### N + DApp Solution

Through the Web 3.0 Dapp PayTube Wallet based on the zkTube protocol, to achieve interaction with DeFi, NFT and other areas, PayTube Wallet will become the "entry" for various Dapp applications to access zkTube.

### Roadmap



Roadmap			
○	zkTube Testnet	April 27, 2021	Testnet Launch
○	Testnet PoS Mining	July 5, 2021	Testnet Mining
○	zkTube Mainnet	July 26, 2021	Mainnet Launch
○	PayTube Wallet	July 26, 2021	Wallet Release
○	Mainnet Mining	August 20, 2021	Mainnet Mining
○	Support Solidity for Smart Contract	September 2021	Network Integration
○	Support Layer 2 DEX	October 2021	Network Integration
○	Support Cross-Rollup	December 2021	Network Integration

## **11. Disclaimer**

1. This document is organized into a book based on its project ideas and technical principles, for reference by technical personnel, or communication and academic research among enthusiasts, and does not constitute any investment advice.

2. This document is not a binding contractual agreement between zkTube and its investors, as this will change with the further development of zkTube.

3. The zkTube Labs does not make any promises and guarantees for the completeness and trend judgment of the content of this document. The current analysis does not designate to represent future development opinions, and any investment behavior may cause asset losses. Anyone making investment decisions based on this will be at their own risk.