A Design of Anti-Tampering Technology Based on Block Chain

SongBo Bu

Information Engineering Branch, Yangling Vocational and Technical College, Xianyang,712000, China

3387527933@qq.com

Keywords: Block chain, Integrity protection, Tamper-proof

Abstract: The development of block chain has become the focus of the global spotlight in recent years and is another technological revolution after the Internet. Throughout the whole development history of block chain, there are still many breakthroughs in block chain technology at present, and there is still a gap between block chain technology and large-scale application. This paper analyzes the tamper-proof technology based on block chain integrity, and analyzes the application scenario of this block chain technology.

1. Introduction

A public key infrastructure (Public-Key Infrastructure) is a general term for a series of specifications and specifications developed to enable the effective use of public keys. PKI is generally referred to as PKI [1]. by its acronym Public key infrastructure can be understood as using public key technology to provide cryptographic services such as encryption and digital signature for network applications, as well as the necessary key and certificate management system. As an infrastructure to provide security services, PKI technology is the core of information security technology and the key and basic technology of electronic commerce [2]. PKI main purpose is to issue a “certificate of identity “, when communicating on the network, if you can confirm the certificate of identity with each other, you can be sure to communicate with the right person. The certificate of identity can not be customized privately and requires a credible certification authority to issue it, just like the identity card issued by a police station to an individual [3]. Although this can solve you have to save a key for everyone you want to contact. But it still involves managing the key, even if you have to have a private key alone [4]. Moreover, the cost accounting needs to consider the hardware + audit + the time source + manpower, the key life cycle needs to be updated regularly, the security that depends on the private key and key management, the key management extensibility and the trust foundation based on the third party, etc [5].

TPAB block chain focuses on data integrity, immutability, and original provenance in our architecture; compared with traditional blockchain accounting techniques, TPAB focus on consistency, sharing, and smart contracts for transaction data over the next year.

Overall architecture:

TPAB provide a security infrastructure consisting of core nodes, aggregators and gateways to create TPAB signatures to achieve data integrity.

1.1 Gateway Layer Facilities

The first layer of aggregation server is the gateway responsible for collecting and processing requests from clients, and then sending aggregate requests to upstream servers. Gateway is a customer-oriented component of infrastructure and provides TPAB services to clients.

The network aggregates hash values and distributes signatures. Each aggregate server processes requests from its servers, adds them to the hash tree, and sends the local root hash to the next higher level server.

  DOI: 10.25236/assah.2021.031

1.2 Laminated Facilities

The hierarchy of aggregate servers creates a global hash tree for each round. the validation network (part of the aggregation network) provides extensive witness access to calendar states, as well as access to the root hash history used in the validation solution.

1.3 Core Nodes

The core cluster is located at the top of the convergent network, running distributed state machines and managing calendars. at an interval of 1 second (approximately), it computes the top root hash value and votes through a distributed consensus algorithm and recommends the top root hash value to the CHC.. Figure 2 The core is responsible for agreeing to the top root hash of each aggregation cycle, then storing it in the calendar database and returning the results to the aggregation network. the regular interval loops (round) used in aggregation and core processes produce an accurate time metric and are embedded in TPAB signatures.

2. Tpab Algorithm Mainly Contains the Following Parts

2.1 Hash Function

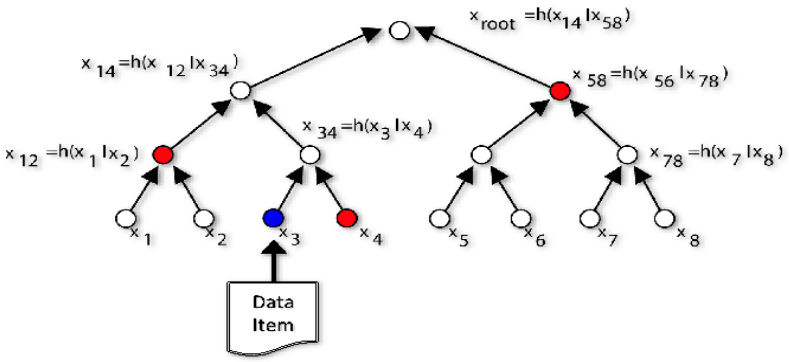
TPAB signature technique is based on the password calculated by the hash function. the hash function is a calculation process that accepts any segment of data and returns a summary of a fixed length, called a hash value or a summary of the original data block. In the past few years, a large number of hash functions have been created.

The hash function has several important characteristics. First of all, the hash operation is one-way irreversible, which indicates that the original data can not be calculated from the output of the operation, which ensures the privacy of the input data [6]. Secondly, any modification of input data will lead to different output results, which ensures the uniqueness of output hash value. Finally, it is impossible to get the same hash value by operation from two different data. This collision attribute again shows that a data can only produce a unique hash value.

TPAB use standard encryption algorithms for SHA -256. internal hash operations The hash value calculated on the client TPAB provides a recommended list of hash algorithms, including the most widely known algorithms, which are the basis of TPAB SDK.

2.2 Ha Tree and Hash Chain

a ha-line tree is required to hash values as inputs, and through a series of repeated hash functions a unique' root value'.

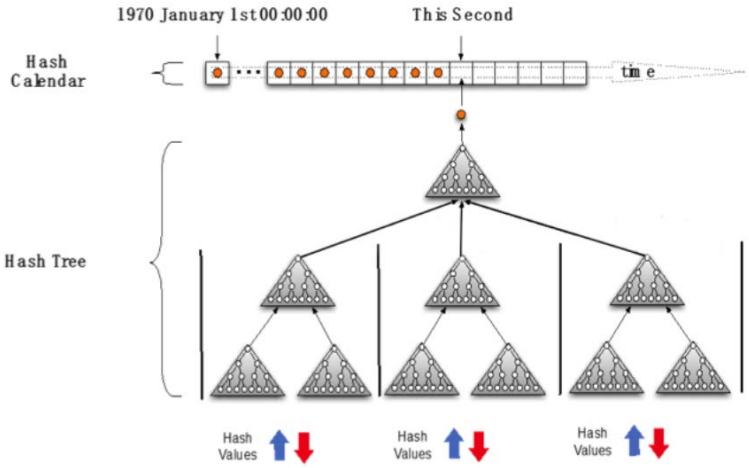


3. Hash Tree

X 1 in the figure above X 8 represent hash values, which are used as inputs to the hash tree (or leaf node), H() represent hash functions, and vertical lines | indicate series (e.g. ABC |XYZ=ABCXYZ). Note h (x 1|x 2)(x 2|x 1).

3.1 Distributed Hash Tree

TPAB services include a global distributed hash tree, as shown in the simplified version below.



3.2 Distributed Hash Tree

Hash trees are created and destroyed every second. The tree is composed of geographically independent operation node levels. We call it the aggregation layer. Each aggregation layer operates as an asynchronous aggregation, receives hash values from its subtree, then generates a hash tree, and then transmits hash root values to multiple parent trees [7]. Aggregation is unbounded and runs on virtual machines or dedicated hardware. In TPAB systems, there are four layers of aggregates. the system acceptable theoretical limit is 264 signatures per second.

4. Aggregation Hash Chain

The purpose of aggregation hash chains is to prove that a particular document hash (signed document) is part of the global aggregation tree. The component of the aggregate hash chain is the brother hash, which will eventually produce the root of the aggregate hash tree when connected from left or right in a given order. Includes:

4.1 Ahc Multiple Instances

the global aggregation tree is constructed by aggregators of many hierarchical tissues. Each aggregator provides a chain for the subtree of the aggregator. When linked together, those chain “fragments” from each aggregator form the complete chain from the document hash to the root of the global hash tree. The calculated output of the aggregate chain (fragment) must match the input of the next hash chain (fragment).

4.2 Aggregation Time

Aggregation time is document signature time, usually expressed from 1970-01-01 The number of seconds UTC (i.e. POSIX time) begins at 00:00:00 and should correspond to the time indicated by the root hash of the global aggregate tree registered in the calendar block. As a result, it is the same for all AHC that together lead to the root cause. This particular field is not encrypted (provided by the calendar block chain) for technical purposes. For example, find the correct calendar block chain for signature extensions.

4.3 Input Hash

The input hash is where the hash chain calculation begins. In the case of the “lowest” aggregate chain in the signature, this is the hash value of the signed document. The input hash will be connected to the left front corner of the right brother hash, and the result will be hash. then the latter will be connected with the next sibling hash level until the final output of the hash chain is calculated.

4.4 Metadata

Left or right links can be metadata, not necessarily brother hash values. This is used to embed specific information into the signature hash chain and to password protect the information. For example, each aggregator (starting from Gateway) embeds the identity of the client into the hash tree.

Metadata has a must attribute - client ID - This is just a UTF-8 string. From an encryption point of view, metadata is the same as any other hash in the tree. that is, the source of the computed hash is also stored in the signature and can be retrieved later.

The output is a series of identities. For example, the following outputs:

user 1::gateway X ::aggregator Y ::aggregator Z

This means that when a SDK application is used to request a signature from a TPAB gateway, its user ID is user1. As the gateway computes the local tree and sends the request to the next level aggregator, it uses user ID gateway X.

5. Calendar Hash Chain

The main purpose of calendar hash chain (CHC) in TPAB signature is to prove the signature time of a particular signature and to ensure long-term integrity. calendar hash chains will lead to widely witnessed events.

CHC AHC. similar since the calendar block is saved and maintained by the Core, only one instance of the CHC component in the TPAB signature hashes from the input to the published value. The output hash of the (highest) AHC in the signature must match the CHC input hash. when TPAB signature is first obtained, the CHC in it is incomplete (or completely lost) because there is no publication available for a particular signature.

release time corresponds to the time of CHC output hash. If the hash chain causes a release, the time is equal to the release time. If the publication does not exist or the signature has not been extended to the publication, the release time only shows the time of the output hash of the (incomplete) CHC.

both left and right links are exactly the same purpose as AHC - they are used to compute the output hash.

6. Publication Record

When the release is available and the user extends the signature to this release, the correct CHC replaces the existing CHC. in the signature In addition, information about the corresponding publication itself will be inserted into the signature and the calendar authentication record will be deleted.

the record contains the published hash (which will match the output hash of the CHC in the signature), according to the time of this hash of the calendar block chain and the reference of the published paper or electronic media.

7. Calendar Authentication Record

TPAB authentication record in the signature is used to enable key-based (e.g. PKI) authentication of the TPAB signature section.

Calendar authentication records contain the PKI signature of the output hash of the calendar hash chain [8]. The PKI signature operation is performed by the core initiating calendar hashchain. The corresponding PKI signature is used as the trust anchor in the key-based verification of TPAB signature. typically, it is used as a short-term trust anchor until the release is available.

The published hash is the output of the calendar hash chain, and the release time is the time corresponding to the calendar block chain. note that despite such field names (publish,publication), calendar chains may or may not result in actual TPAB releases.

PKI signature type displays the type and format of IANA defined key-based signature. PKI signature value is a key-based signature that specifies the type and format. The signed file is the published data structure. the certificate identifier and the certificate library (optional) point to the certificate used to verify a key-based signature.

8. Working Process:

TPAB architecture mainly includes two parts: signature and verification. The signing process is as follows:

8.1 Signature Issues

1) the hash value reaches the gateway layer, it will solidify the gateway and send the formed root hash value to the aggregation layer through the Merkle tree;

2) the aggregation layer server processes the root hash value sent by the gateway layer and adds it to the calendar chain, the aggregation layer server sends the formed root hash value to the core layer;

3) the core layer server is connected to the block chain network after authentication to complete the data chain;

4) the client will receive a layer-by-layer signature returned by each layer of server containing the hash value submitted by the user, the root hash and its calculation path;

5) verification uses signature files to verify the integrity of the data.

8.2 Validation Issues

After the gateway receives the hash value, the signature file, obtains the corresponding hash computation path from the signature file, carries on the operation again, compares the root hash obtained by the final operation with the root hash in the signature file, thus obtains the verification result;

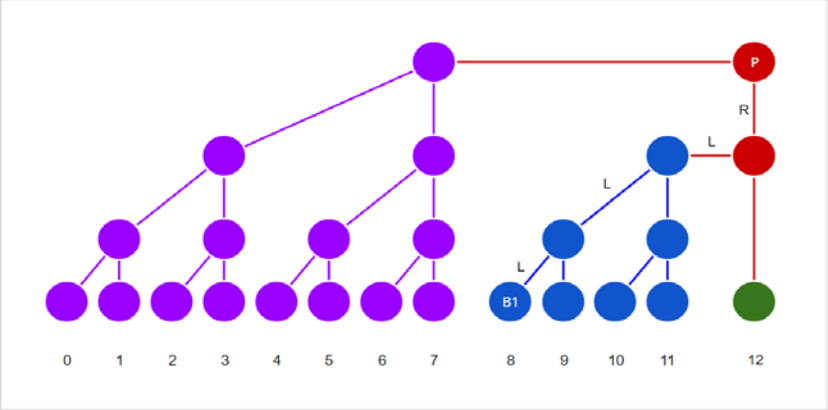
Each signature takes about 2-5 seconds, when the gateway signature, the gateway server is still calling different servers to operate, it takes time; but can support multiple data to sign at the same time, support high concurrency; The signature file contains the path of hash value operation in the chain. If the signature file is lost or modified, the path of hash value operation in the chain can not be verified;

9. Time Verification Algorithm

TPAB signature time is also encoded in the shape of the calendar hash chain. This enables the computation/validation of the time from the calendar hash chain and the resulting release time. such validation is performed by TPAB SDK as part of an internal validation policy.

9.1 Construction of the Calendar Tree

The binary tree of the calendar block chain (where each leaf corresponds to one second) is constructed in a certain deterministic way. The following example trees illustrate the build process:



The sample tree has 13 leaves:

First, take 8 leftmost leaves because 8 is the smallest number and is a power of 2 less than or equal to 13. this allows us to build the largest perfect binary subtree (purple).

Apply the same procedure repeatedly to the remaining leaves. In the case of the sample tree, the next subtree (blue) is formed by four leaves (because 4 is the maximum number of powers of 2 and less than or equal to 5) [9]. The remaining green leaves form a third subtree.

Once each subtree is formed, combine it into a tree. This process starts from the right by combining the roots of the green and blue trees, and then results in the combination with the roots of the purple trees.

This means that the following applies to any node in the tree:

• Its left subtree is always a perfect binary tree (the number of leaves is exactly 2 N).

Its right subtree follows the same structure as the whole tree, with only a small number of leaves (for example, its left subtree is perfect again, and its right tree follows the same structure as the whole tree).

9.2 Finding the Position of a Leaf

Building a tree in this way can locate any leaf in the following ways:

• Total number of leaves in the tree; and

• The path from the root node to the desired leaf.

The total number in the above example is 13. If we want to find or verify the location of the B1, the path from the root node to it will be RLLL (meaning from the root B1, we will first move once to the right and then move 3 times to the left).

the algorithm used to find the location traverses from the root node to the desired leaf node using the provided path. During traversal, at each node, it is easy to find:

• The number of leaves in the tree, which is the root node; and

Number of leaves in left and right subtrees.

9.3 Number of Leaves in Subtrees

Considering the way the tree is built, we know that the left subtree is a perfect binary tree. Therefore, it must have 2 N leaves, where N is the largest integer such that 2 N is less than the total number of leaves. The following examples illustrate:

total number of leaves N 2N (leaves in left cotyledon must be less than total number of leaves)

13 3 8(<13)

16 3 8(<16)

18 4 16(<18)

after we know the leaves in the left subtree, it is easy to find the number of leaves in the right subtree because we know the total number of leaves – from the previous “iteration “, or if we are at the root node, this is all as input to the algorithm.

Moving Down the Path

Knowing the number of leaves in the left and right subtrees helps to narrow the range of leaf nodes that each movement expects, as follows:

• When we are at the root node, we know that the B1(position based on 0) is 0.12.

When we move to the right, narrow the range to (0+8).12, because there are eight leaves in the left subtree, B1 not one of them.

As we move to the left, narrow the range to 0.(12-5) because there are five leaves in the right subtree, not one of B1.

In general, this means that moving the right increases the beginning of the number of leaves in the left subtree, and the left moving the left reduces the end of the range by reducing the number of leaves in the right subtree.

9.4 Time Verification

Put the above algorithm in the context of TPAB signature and TPAB calendar block chain, we can verify the POSIX time of TPAB signature (since 1970-01-01) in the following way Number of

seconds since 00:00:00 UTC), POSIX publication time P (assuming TPAB signature has been extended to this publication).

TPAB the shape of the calendar hash chain in the signature, which is the path from the root of the tree constructed P time to the leaf corresponding to the signature time.

9.5 Algorithm Illustration

The following recursive function written in Python findPosition find the location of the leaf specified by the given path. Help function getLeftLeafCount used to get the number of leaves in the left subtree. note that the number of leaves in the tree is not explicitly given as a parameter as this can be calculated from the posMin and posMax.

```
def findPosition (posMin, posMax, path):
leafCountAtNode =posMax -posMin +1
leftLeafCount =getLeftLeafCount (leafCountAtNode)
rightLeafCount =leafCountAtNode -leftLeafCount
+ print “Total leaf count “+str (leafCountAtNode)2
“,+ left subtree “+str LeafCount
“, right subtree “+str (right Leaf Count +”; posMinright subtree “+str
str (posMin +”, posMax “+str (posMax)
# If no more moves left,posMin and posMax have to be equal
if len (path)==0:
assert (posMin ==posMax)
return posMax
move =path[0]
if move == R':
posMin +=leftLeafCount
elif move == L':
posMax -=rightLeafCount
else:
raise ValueError ('Invalid move'+move)
return findPosition (posMin, posMax, path [1:])
# Finds the number of leaves in the left subtree
# given the total number of leaves in the tree
def getLeftLeafCount (total Leaf Count):
# The answer is the largest power of 2that's
# strictly less than the total number of leaves
# Keep computing successive powers of 2until
# we reach or exceed the upper bound
leftLeafCount =1
while leftLeafCount <totalLeafCount :
leftLeafCount =leftLeafCount <<1
# Now leftLeafCount >=totalLeafCount ,so we have
# come one step too far,so we take one step back
leftLeafCount =leftLeafCount >>1
return leftLeafCount
```

10.Industry application scenarios:

(1) Compliance and audit-signing necessary records, transaction data, sensitive documents and other information to be audited in a timely manner, increasing the speed of audit and reducing the risk of non-compliance.

(2) Easy to tamper with data 2. to publish – published information such as test results, reports, business information, and analysis reports [10]. The integrity and authenticity of electronic data between customers and partners have proved to greatly increase trust and can use open new methods for business transactions and transactions.

(3) intellectual property protection - online publishing of content, whether in the form of a website or blog, image, audio, video, proof of its integrity can be more property protection when published.

(4) file archiving and sharing – although we are able to store original documents as electronic data, at the same time we may reduce credibility and integrity in stored procedures. If we sign the document without a key, the cost of archiving can be significantly reduced when the document has evidence of integrity.

10. Conclusions

The development of blockchain has become the focus of global attention in recent years, and it is another technological revolution after the Internet. Looking at the whole development history of blockchain, there are still many breakthroughs in blockchain technology, and there is still a gap between blockchain technology and large-scale application. The integrity and authenticity of electronic data between customers and partners greatly increase trust, and open new methods can be used for business transactions and transactions. Although we can store the original document as electronic data, at the same time we may reduce the credibility and integrity of the stored procedure. If we sign a document without a key, when the document has integrity evidence, the cost of archiving can be significantly reduced.

References

[1] Cha Jialing, Zhang Yuan. Application research of blockchain technology in the design of hospital patient diagnosis and treatment information system[J]. Modern Information Technology, 2020, 4(08):186-188+191.

[2] Ren Haowen, Yang Yaqi. Application of blockchain distributed technology in power demand side response management[J]. Telecommunications Science, 2019, 35(05):161-166.

[3] Liang Tianjing. Discussion on the application of blockchain in the security system-investigation and verification-verification [J]. Digital User, 2019, 025(036):97-98.

[4] Zhang Yawei, Zhang Wenyin, Wang Jiuru, etc. Framework design and analysis of digital asset management system based on blockchain[J]. Computer Science and Applications, 2019, 009(001):28-37.

[5] Huang Zhongyi. Exploration and application of blockchain technology in the field of industrial Internet platform security[J]. Cyberspace Security, 2018, 9(10):26-29+37.

[6] Gao Mengjie. Searchable medical data sharing scheme based on blockchain[J]. Journal of Nanjing University of Posts and Telecommunications: Natural Science Edition, 2019, 39(6):94-103.

[7] Hu Jie, Ge Changtao, Sun Yong, et al. Research on logistics information management framework based on blockchain[J]. Logistics Technology, 2018, 41(10):40-42.

[8] Jia Yinshi. Research on Network Copyright Transaction Based on Blockchain Technology[J]. Science Technology and Publishing, 2018, 000(007):90-98.

[9] Xu Chao, Chen Yong. Research on audit methods under blockchain technology [J]. Audit Research, 2020(3): 20-28.

[10] Zhang Lihua, Fu Donghui, Wan Yuanhua. Blockchain-based secure sharing of medical records[J]. Journal of East China Jiaotong University, 2020, v.37; No.175(05):125-130.