(This document is a translation of original document written in Catalan, may include some errors)

Pascal Coin: Crypto currency without need of historical operations

IMPORTANT: THIS DOCUMENT IS A TRANSLATION OF ORIGINAL PASCAL COIN WHITE PAPER WRITTEN IN CATALAN LANGUAGE. MAY INCLUDE SOME TRANSLATION ERRORS

Definition:

The "Pascal Coin" is a cryptographic virtual currency [1] operated via a P2P protocol that works through account numbers instead of doing it through public keys. This allows the amount of predictable account numbers and thus efficiently storing their balance, thus avoiding the need to consult past operations, so operations only need to check their current balance. The way to operate with account numbers is with the cryptographic keys. Each account number has unique PIN (cryptographic key) that can be modified at any time. (A PIN can be used to operate more than one account, but an account only has one valid PIN)

Introduction:

This document is intended to define a variant of cryptographic currency based on knowledge provided by BitCoin [1]. Before reading this document is recommended to have a deep technical knowledge of BitCoin to understand the concepts presented here.

History:

In 2009, thanks to the invention of BitCoin, a new method for creating a virtual currency transactions without a central authority was created. This method included two new elements that were the basis of BitCoin (and therefore of all new virtual coins that have been created since then)

- "BlockChain" [1] [5] The block chain is the source of where the data is stored and all historical transactions are secure transactions by storing them in a Merkle Tree [2] [5] that prevents an operation can be carried out for duplicate or without the corresponding funds or double spent.

- Mining [1] [3] [5]: The mining process is unique and is the basis for the creation of the block chain and the generation of new currency. No mining there is no way to guarantee blocks consistency. Also, this method provides gratification to BitCoin generating new currency.

But although it is similar to a normal transaction, Bitcoin has some aspects that distinguish it from a traditional bank.

Firstly account numbers correspond to a cryptographic key [4], so they are not easily usable, unless electronically, because they size is 25 bytes, stored between 26 and 35 characters (Base 58). This means that if a person wants to make a transaction the account number is necessary

to be able to read electronically because manually is difficult to enter an account number without mistakes (but provides mechanisms to detect errors because it contains a checksum)

To verify a transaction needs to mention the previous transaction from which income is able to spend. That is, it is not to look at the balance of a bank account transfer but upon ... and obviously see that this transfer has not been previously spent. Although the mechanism ensures perfect operation involves always have all historical transactions affecting BitCoin address, and consequently, a miner must have all historical transactions.

Another very important aspect is the fact that if someone loses a key to operate your BitCoin, never be able to withdraw the funds you have in this address (except generate the same private key). Although this may seem secondary certainly be destabilizing for currencies such as Bitcoin, because the majority of addresses that were generated in its first two years of life (theoretically were created by the author the BitCoin, Satoshi Nakamoto) have never been used. Certainly only the creator knows if they are lost or not. What would happen if suddenly is discovered today that can be used? Most likely there would be a massive devaluation of its previous price due to ignorance of this factor or wrongly assuming that these BitCoins were lost.

Also keep in mind that BitCoin only operates with elliptic curve cryptographic keys "secp256k1" [4]. While today this key is considered safe, no guarantee that incoming years this key can be vulnerable and lose efficiency.

The alternative proposed by PascalCoin

PascalCoin proposes an alternative to the basic operation of Bitcoin, through which change several aspects for working on the new virtual currency:

- PascalCoin operates with traditional numbered account (like a Bank) instead of doing it in directions that correspond to elliptic curve public key "secp256k1"
- PascalCoin don't need to search transactions received for an account, but directly query its balance (stored in a safe box) and therefore does not need to have the historical transaction to work. The historic transactions stored within the chain block only serves to ensure that the safe box was not handled inappropriately.
- PascalCoin provides a standard way to include a concept of a transaction (Payload [5]) that only the intended recipient can read it. This is achieved because the public key is known for everybody and can be used to encrypt messages that only the owner of the private key can decrypt.
- PascalCoin provides a method set by protocol to retrieve coins that are not used instead (lost key). This method only applies if after a certain time the owner does not make any operation with the account private key.
- The operations that can be performed not depend on programming language (like BitCoin) due they are fixed by the protocol. In version 1 of the protocol only supported the following:

  - Transaction from 1 to 1
  - Change key to operate with an account
  - Recover money stored in an unoperative account (lost key)

- The possibility of creating new types of operations for new revisions protocol is open (Upgrade)
  - The way to know whether the miners will accept new versions of protocol is including protocol version upgrade proposal to chain blocks. Miners will choose its criteria to begin accepting this new protocol mined blocks, assuming the risk of creating orphan blocks [5].

Sample transfer:

In BitCoin to make a transaction must indicate target BitCoin address that can exist or not, for example:

Transaction from: ABCDEF...XYZ to ZYX....CDBA amount X

To make the transaction needs to have the private key of the address ordering the transaction, which is the same key that generates public address "ABCDEF XYZ ..." and also need to indicate the previous transaction received by this address that will be used to spend. (No in-tx, no out)

In PascalCoin indicate the account number of the source and destination address, no keys, only numbers. Necessarily have to exist both your source and destination account number:

Transaction from: 1234 to 4321 amount X

To make the transaction PascalCoin also needs the private key of the 1234 address but in this case is not necessary to know previously transaction received at 1234 to be spend, only its current balance.

Advantages of this method:

The fact that the transaction does not indicate previously transaction implies that we don't need to look at the historical stored transaction, so is more quickly to check any operation because the balance is known and stored in each account, freeing disk space for miners.

Another advantage is that you can use different types of private keys and not only the key "secp256k1" used in Bitcoin, that improves safety

The chain blocks and mining not only generates new currency but also generate account numbers with which they can operate.

Because the public key is always known (in Bitcoin not always, unless already carried out an operation to an address) in PascalCoin can encrypt messages using public key target so that only the intended recipient can decrypt the message. So there will be more similarity between a traditional bank account number where each operation may have been a concept.

Disadvantages of this method compared with BitCoin:

Because you can only operate with account numbers quantity is limited to the number of units produced, while in BitCoin address quantity is virtually unlimited.

Each time you generate a new block to calculate a new output value for the next block will grow in size sequentially and a calculation for the new checksum is needed (See "Safe Box Hash"). The calculation, although fast, is a negative factor as compared to BitCoin is a process that will become increasingly expensive.

New item: Safe Box

In PascalCoin safe box is where you save amount of each address, and also includes elements to perform operations there safely.

Internally the Safe box is a set of blocks (called "AccountBlock"), where each block has been generated through each block of the block chain. Each "AccountBlock" also contains account numbers generated plus a checksum to ensure integrity block has not been tampered with.

One of the distinguishing features of the safe is incorporating a general checksum (hash) that guarantees the status of all accounts at a particular time. This value is known as "Safe Box Hash" and is used to undermine a new "BlockChain" and guarantees not to be tampered with the safe that was used for calculations.

General operation of PascalCoin

First block (zero block):

At one zero point the "Safe Box" is empty and there is no account inside.

Miners seek a new block for the "BlockChain" satisfying the condition of "Proof-of-work" [5] considering the "Safe Box Hash" is an initial fixed value (hardcoded) . Once created across the first account block, this creates N accounts (each "AccountBlock" consists of N accounts).

By modifying the safe will calculate the new "Safe Box Hash" to be used by the next block "BlockChain."

The number of account numbers of the safe box will always be equal to the quantity of mined blocks * N, where N corresponds to a constant defined by protocol.

X Block:

While the miners seek a new block that meets the requirements of the proof of work, and taking as starting the "Safe Box Hash" of the block X-1, will receive operations that will be incorporated into Merkle Tree if they meet the requirements. In no case will look at the historical chain of blocks to check the veracity of an operation as it will only check if the balance / account status of the "Safe Box" corresponds to the requested operation. When receiving a valid operation that modifies the state of Merkle Tree operations and therefore the basis for calculating the "Proof of Work".

Technical specifications of objects used by Pascal Coin

Account

Each account contains:

Name	Type	Description
Account number	Unsigned 32 bits	Ordinal number for a single account. Value unchanged forever.
Public EC Key	Public Key (type, X and Y) (Between 66 and +200 bytes)	The public key is the PIN of an account. This value can be changed at any time.
Balance	Unsigned 64 bits	Account balance, calculated from the operations of the block chain
Updated block	Unsigned 32 bits	Number of the last block of the block chain that has operated this account. This value let you know when an account has been inoperative in order to rescue its balance
N Operation	Unsigned 32 bits	Incremental value that indicates how many transactions were made with this account and ensures that orders of operations are unique and not duplicable

Block Account

The account numbers are grouped in blocks "Block Account", where each block is generated every time a miner creates a new block to the block chain with operations so that transforms affected blocks. Protocol version will specify how many accounts are generated for each Block.

The content of each "Block Account" is as follows:

Name	Type	Description
Block number	Unsigned 32 bits	Block number that is equivalent to the Block Chain. This value is unchanged forever.
Accounts	Array of N Accounts	Fixed Array (size N) with account numbers that are generated by this Block
Timestamp	Unsigned 32 bits	Unix Timestamp when generated. This value is unchanged forever.
Block Hash	32 bytes	Hash value of this block. It changes every time the chain block change any of account included in this block account. This ensures the integrity of this block.

Finally safe box contains a checksum that is the hash of all “Block Hash”. This value is known as "Safe Box Hash"

The Block Chain

The Block Chain is similar Bitcoin, but adding the calculation of the "Safe Box Hash" indicating the state it had safe box before applying the operations included in the Block Chain Merkle Tree.

The "BlockChain" contains the following fields:

Name	Type	Description
Block number	Unsigned 32 bits	Block number generated by miners. correlative
Account key	Public Key (type, X and Y) (Between 66 and +200 bytes)	Each of N "Accounts" generated will have this public key
Reward	Unsigned 64 bits	
Fee	Unsigned 64 bits	
Protocol version	Unsigned 16 bits	Protocol version
Protocol available	Unsigned 16 bits	Protocol number that can apply the miner owner of this block. Serves for future protocol upgrades
Timestamp	Unsigned 32 bits	Unix Timestamp
Compact target	Unsigned 32 bits	
Nonce	Unsigned 32 bits	
Old Safe Box Hash	32 bytes	Value of "Safe Box Hash" from which operations are carried out under this block
Operations Hash	32 bytes	Merkle Tree Hash
Proof of Work	32 bytes	

(Protocol may include additional fields, like a Payload…)

Security Guarantee

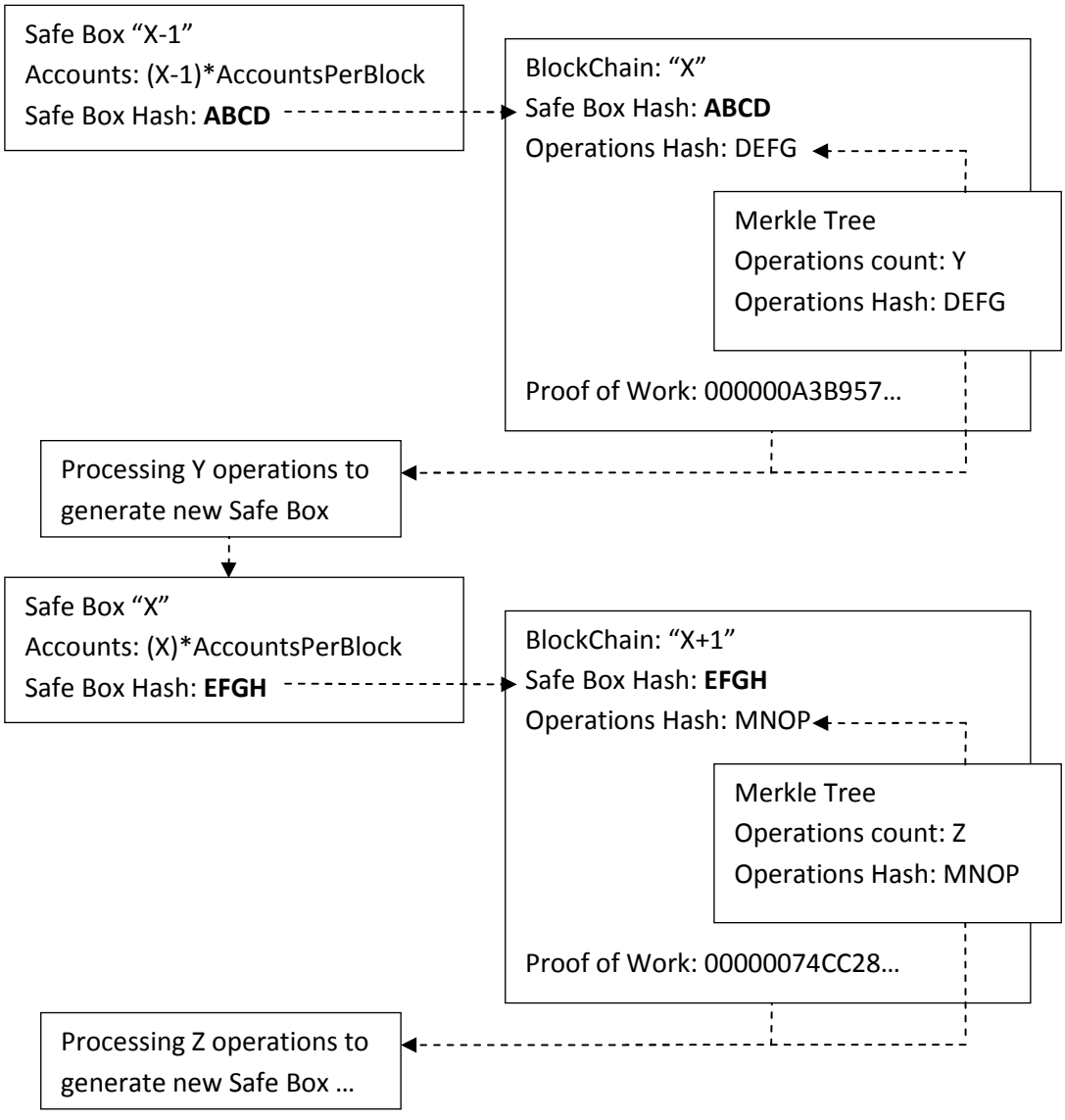
Safe Box integrity

To check the Safe Box is intact and correct must perform the following operations:

- For each "Block Account" calculate its Block Hash
- Finally calculate the "Safe Box Hash", that is a hash of all the "Block Hash"
- The value of "Safe Box Hash" is unique to each "Safe Box" depending on the number of units they contain, so that you cannot manipulate. Because the "BlockChain" is generated with its initial "Safe Box Hash", unique for each block chain

Generation of "Block Chain" from previous Safe Box

To ensure that no Safe Box manipulates, its "Safe Block Hash" is included on each Block Chain.



Duplicate operations control

Each account contains a field called "n_operation" as an incremental integer, so that when a new operation is sent it indicates the new value of "n_operation" and has to be last +1. Since operations are signed, it is possible to create two operations with the same "n_operation" but only the first accepted by the miner will be valid, discarding the second to have a value that does not correspond to +1.

Hard Coded Safe Box

One advantage of this method allows not necessary to view the history of operations. So we may want to have a starting point in the block chain that is different from the block 0, space saving and downloading in future implementations.

This is accomplished by a "Hard Code" any of SafeBox generated in a previous predetermined time. This "Hard Code" will be implemented as an initial value and can be found that is true as follows:

To determine a "Safe Box" of "BlockChain" "X" is just right to access "BlockChain" X has generated it and calculate it. If result of "Safe Box Hash" is the same can ensure it is correct. (And so on until the block "0" original).

References

[1] – Satoshi Nakamoto, “BitCoin: A Peer-to-Peer Electronic Cash System” https://bitcoin.org/bitcoin.pdf

[2] – R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[3] - A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002

[4] – Certicom Corp, “SEC 2: Recommended Elliptic Curve Domain Parameters” http://www.secg.org/sec2-v2.pdf

[5] – BitCoin Foundation, “Bitcoin Wiki” https://en.bitcoin.it/wiki