Bitcoin Candy Whitepaper

Contents

Abstract

Bitcoin Candy is a Hard fork of Bitcoin Cash with quantum resistance as its main feature. In this whitepaper, we will introduce its technical specifications and future developing plans.

1. Main Tech Specification (How it is different from BCH or Bitcoin)

1.1 Introduction

Bitcoin Candy is a hard fork of Bitcoin Cash and it inherits most of Bitcoin Cash merits. Bitcoin Cash is the first hard fork of Bitcoin which supports increasing block size rather than segwit as Bitcoin's scaling solution. Bitcoin candy was inspired to bring quantum-resistance into cryptocurrency world. It is developed from bitcoin ABC 0.16.2.7 version but also introduces other features which will be covered in the following section.

1.2 Main Features

The main features of Bitcoin Candy are listed in the following table:

Fork Height	512666
Fork Time	Feb 2, 2018 6:41:33 GMT+8
PoW algorithm	EQUIHASH(144,5)
Total Supply	21 billion
Whether Pre-ming	Yes, 1% pre-mining
Block Interval	2 min

Replay protection	Enabled, secure for both Bitcoin and Bitcoin Cash
Block Size	32M
DAA	CDY-adapted LWMA

1.3 PoW Algorithm

Originaly， Bitcoin Candy used Equihash <200,9> as its PoW algorithm which is one of the most common GPU mining PoW algorithms and it's widely used in Zcash, ZenCash， Bitcoin Gold etc.， We used this PoW algorithm since it's GPU-mining instead of ASIC mining and can make the network a more decentralized one.

However, since this algorithm is widely in use. There are lots of miners or pools control plenty of hash power of this algorithm which makes small coin like Bitcoin Candy vulnerable: They can easily implement 51% double-spending attack， selfish-mining， miners jumping on and off constantly and so on.

After being attacked for several times，we decide to use a less common PoW algorithm：**Equihash <200,9> with personal string: CandyPoW.** Ergo, other Equihash hash power can't switch to our network without replace their mining client and change their configuration. Also, it prevents upcoming ASIC miners.

1.4 DAA

1.4.1 Introduction

DAA, also refered as Difficulty Adjustment Algorithm, is one of the most important aspects of a PoW. A good DAA should reflects the hash power of the entire network as precisely as possible. However, in practice, there always exists delay: the DAA takes some blocks to catch up the real-time hash-rate. When the hash power fluctuates frequently, it's almost impossible for a DAA to catch up. There are two important metrics to evaluate a DAA's performance:
(1) Fast response: when hash-power changes, DAA changes difficulty correspondingly in a short period.

(2) Stability: difficulty doesn't fluctuate too much because the time to generate a block is usually unstable even if network's hash-power is stable.

A DAA is usually hard to have both two features, it has to make a tradeoff between these two：Stability comes at a price of losing the speed of response and vice versa. For example, bitcoin's DAA is stable but very slow, it adjusts difficulty every 2016 blocks. Some DAA adjust difficulty every block, for example: Digishield. Still, it takes several dozens of blocks to adjust its difficulty to the right value. In the meanwhile, it loses stability which makes it more vulnerable to big miners.

For bitcoincandy, the situation is even more urgent than other ordinary coin. Bitcoin candy is new and small. A big miner(whale miner) can easily surpass the constant hash power which makes our network vulnerable. To deal with this situation, we need to introduce a bitcoincandy adaptive DAA。

1.4.2 Bitcoin Candy Adaptive-DAA

Bitcoin Candy's DAA need to suffice the following conditions:

(1) Fast response:

If the DAA's reponse to hash power change is slow for bitcoin candy, big miners can take advantage of it. A big miner jumps on when the difficulty is low, since the DAA is slow, it takes dozens of blocks before difficulty gets high enough. To deal with this situation, we add an emergency response mechanism：

if the last 10 blocks are mined within 5 minutes, the difficulty will double (the current block difficulty) in the next block; if the last 10 blocks are mined within 10 minutes, the difficulty of the next block equals to 1.5 × current block's difficulty; if the last 5 blocks are mined within 90 seconds, the difficulty of the next block equals to 4 × current block's difficulty.

(2) Stability:

to avoid difficulty fluctuates too much, we choose LWMA with N = 45, the description of LWMA can be found in:

https://github.com/zawy12/difficulty-algorithms/issues/3

LWMA is relatively fast without losing stability, we can choose a smaller N for faster response but it will lose stability. We choose N = 45 as a compromise.

(3) Avoid Difficulty dropping too fast:

If the difficulty drops too fast, imagine difficulty is high enough now to keep the whale miner away. But with the whale miners gone, it only takes one or two blocks for difficulty drops low enough to attract whale miners coming back again. To avoid this situation, we add a threshold:

the difficulty can only drop 30% each block at most.

1.5 Scaling solution

Because of the limitation of block size and block interval , bitcoin and ethereum have undergone scaling problems for a long time: low tps, e.g, 7 transactions per second for bitcoin at most; high transaction fee.

To address this problem, bitcoin uses segwit along with lightning network (not widely in use yet).

For bitcoin candy, we solve scaling problem by:
(1) Increasing block size to 32MB (also implemented on Bitcoin Cash)
(2) Short block interval: 2 minutes instead of 10 minutes.

1.6 Replay Protection

The replay protection is enabled by setting SIGHASH_FORKID bit in the signature's sighash type. And the following digest algorithm:

In order to ensure proper activation, the reference implementation uses the SCRIPT_ENABLE_SIGHASH_FORKID and SCRIPT_ENABLE_CHANGE_FORKID flag when executing EvalScript .

Digest algorithm
The proposed digest algorithm computes the double SHA256 of the serialization of:
nVersion of the transaction (4-byte little endian)
hashPrevouts (32-byte hash)
hashSequence (32-byte hash)
outpoint (32-byte hash + 4-byte little endian)
scriptCode of the input (serialized as scripts inside CTxOuts)
value of the output spent by this input (8-byte little endian)
nSequence of the input (4-byte little endian)

hashOutputs (32-byte hash)

nLocktime of the transaction (4-byte little endian)
sighash type of the signature (4-byte little endian)
Items 1, 4, 7 and 9 have the same meaning as in the original algorithm[3].

hashPrevouts
If the ANYONECANPAY flag is not set, hashPrevouts is the double SHA256 of the serialization of all input outpoints;
Otherwise, hashPrevouts is a uint256 of 0x0000.....0000.

hashSequence
If none of the ANYONECANPAY, SINGLE, NONE sighash type is set, hashSequence is the double SHA256 of the serialization of nSequence of all inputs;
Otherwise, hashSequence is a uint256 of 0x0000.....0000.

scriptCode
In this section, we call script the script being currently executed. This means redeemScript in case of P2SH, or the scriptPubKey in the general case.
If the script does not contain any OP_CODESEPARATOR, the scriptCode is the script serialized as scripts inside CTxOut.
If the script contains any OP_CODESEPARATOR, the scriptCode is the script but removing everything up to and including the last executed OP_CODESEPARATOR before the signature checking opcode being executed, serialized as scripts inside CTxOut.
Notes:
Contrary to the original algorithm, this one does not use FindAndDelete to remove the signature from the script.
Because of 1, it is not possible to create a valid signature within redeemScript or scriptPubkey as the signature would be part of the digest. This enforces that the signature is in sigScript .
In case an opcode that requires signature checking is present in sigScript, script is effectively sigScript. However, for reason similar to 2, it is not possible to provide a valid signature in that case.

value
The 8-byte value of the amount of bitcoin spent in this input.

hashOutputs
If the sighash type is neither SINGLE nor NONE, hashOutputs is the double SHA256 of the serialization of all output amount (8-byte little endian) with scriptPubKey (serialized as scripts inside CTxOuts);
If sighash type is SINGLE and the input index is smaller than the number of outputs, hashOutputs is the double SHA256 of the output amount with scriptPubKey of the same index as the input;

Otherwise, hashOutputs is a uint256 of 0x0000.....0000.
Notes:
In the [original algorithm][3], a uint256 of 0x0000.....0001 is committed if the input index for a SINGLE signature is greater than or equal to the number of outputs. In this BIP a 0x0000.....0000 is committed, without changing the semantics.

sighash type
The sighash type is altered to include a 24-bit fork id in its most significant bits.
ss << ((GetForkID() << 8) | nHashType);
This ensure that the proposed digest algorithm will generate different results on forks using different fork ids.

We use SIGHASH_FORKID = 111 which is different from BCH's SIGHASH_FORKID to enable replay protection from BCH as well as Bitcoin which doesn't use SIGHASH_FORKID at all.

2. Quantum-Resistant Solution

2.1 Post-Quantum Cryptography Background

In modern times, public key cryptography is a widely-used technology to secure electronic communication over an open networked environment such as the Internet, without relying on a hidden or covert channel, even for key exchange. Specifically, it can be used to create a digital signature which later can be utilized to validate each other's identity, non-interactively encrypting data, avoiding fraud and so on.

In bitcoin and other blockchains, a public key cryptography key-pair is mainly used in account system to keep users' asset safe and represent user's identity in the network. A user can have multiple public-private key-pairs.

However, with the emergence of quantum computers, popular public key cryptosystems nowadays (include bitcoin) will be broken eventually according to Shor's algorithm. Bitcoin and ethereum etc. use secp256k1 whose security are based on elliptic curve cryptosystems will also be broken. In other words, a hacker can steal anyone's bitcoin if he has access to a powerful quantum computer.

To eliminate post–quantum threat, block-chains need to come up with a new account system which is quantum-resistant.

2.2 How to choose an appropriate Quantum-resistant Public Key Algorithm for block Chain

In the post-quantum cryptography, there are four main directions:
(1) Lattice-based cryptography;
(2) Hash-based Cryptography;
(3) Code-based Cryptography;
(4) Multivariate Public Key Cryptography(MPKC for short).
The math problems behind all these cryptosystems are all quantum-resistant, they all can be a potential solution to imminent quantum threat.

However, these cryptosystems are usually impractical: they are either insecure (insecure trapdoor function design, or vulnerable to side channel attack etc.) or two slow, or the key size is too large. So far, not a post-quantum public-key cryptography algorithm is secure, efficient and compact like ECDSA.

To find an appropriate post-quantum cryptography algorithm for blockchain account system, following conditions need to be satisfied:
(1) As compact as possible
Block-chain consists of multiple blocks which are chained together by their hashes and nearly infeasible to tamper. Every full-node store one copy of these blocks, and the size of these blocks grows as time elapses. To reduce the burden of a full-node to store all these blocks, also increase the scalability of the network, the public key size and signature size should be as compact as possible.

(2) Efficient
The speed to generate a signature of a transaction and to verify its correctness should be efficient. Otherwise, it will affect a transaction's propagation since it will take a long time before every node accepts it as every node needs to check its validity before accepting it.

(3) Secure
Security is the top priority for a blockchain account system since it relates to users' property safety. Without security, hacker can steal a user's money and all other efforts will be in vain. Thereby, we should choose a secure (better if a provable security is provided under strong computational hardness assumption) post-quantum algorithm for account system.

However, in the existing post-quantum cryptography algorithms, besides quantum-resistance,

there is hardly an algorithm that can compete with ECDSA on efficiency, security and size. For example， lots of algorithms have key size over 10KB (or even 1MB for code-based schemes), BLS used in HCash is vulnerable to side-channel attack, XMSS hypertree used in QRL is kind of slow.

Anyway, to select an appropriate for Blockchain account system, we need a secure (better with a strict security proof under strong computational hardness assumption), fast and relatively compact (key and signature size) algorithm.

2.3 Our choice

After doing some research about the candidates of standard quantum-resistant public-key cryptographic algorithms listed by NIST 2018 1st round, we select Falcon as our post-quantum solution. It has following excellent qualities:
(1) Provable security against Quantum computers and traditional computers;
(2) Fast signing and verification: over 1000 tps on an ordinary PC.
(3) Relatively small size: under 128bit security level, Flacon has public key size of 897Bytes and private key size of 2*897Bytes, 617.38Bytes signature size.

Falcon is a lattice-based signature scheme, it is based on NTRU lattice with fast fourier sampling and GPV framework to build a hash-and sign signature scheme. Detail description of this algorithm can be found in:
https://falcon-sign.info/falcon.pdf

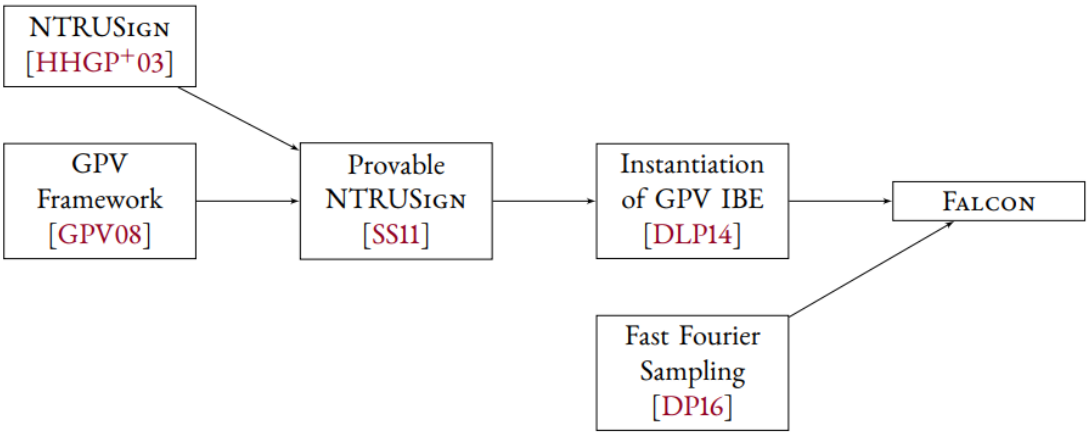

Figure 1 The genealogic tree of Falcon

After selecting Falcon as our quantum resistant algorithm, our account system can be constructed as:

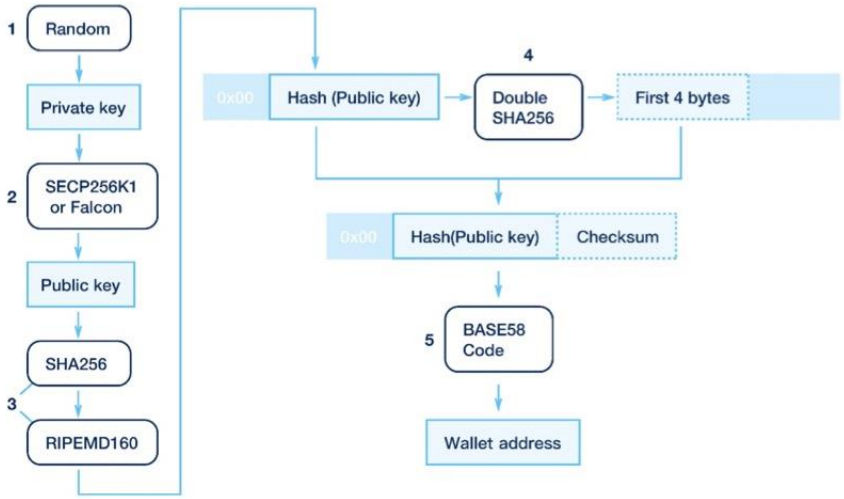

Figure 2 Quantum-resistant blockchain account system

3. How to burn the unclaimed Bitcoin Candy

3.1 A brief introduction of hard-fork coin

A hard-fork of a block-chain usually means some nodes in the network of this blockchain are going to apply a new set of rules which are incompatible with the previous ones. This essentially creates a fork in the blockchain: one path follows the new, upgraded blockchain, and the other path continues along the old path.

Start with Bitcoin Cash, at the end of the last year, lots of hard-fork coins of BTC emerged such as: Bitcoin Gold, Bitcoin Diamond, Bitcoin God, Bitcoin Private etc. Hard-fork coins have had their prosperity. However, lots of them have vanished from people's sight nowadays.

Bitcoin candy is the first hard fork of Bitcoin Cash which applies a new set of rules such as:(1)Equihash as PoW; (2) Shorter Block Interval;(3) Different DAA (4)Quantum Resistance solution(in the near future) and so on.

3.2 How to burn the unclaimed Bitcoin Candy

Since it's a hard fork of Bitcoin Cash, anybody owns BCH before hard fork height 512666 can claim CDY by the ratio 1:1000, i.e. if you have 1BCH in your wallet, you can get 1000 free CDY by importing corresponding private key to CDY wallet.

However, there are a huge amount of unclaimed CDY in the network. To make the community and our ecosystem a more healthy one. We are going to burn these unclaimed CDY.

We are considering two ways to recycle the unclaimed coins:

(1) **Option One**: Cut off the blocks before 512666, and fork at a certain height, transactions between these two heights are assumed valid by default. And transactions after the new fork height will considered invalid if they use UTXOs before 512666.

(2) **Option Two**: Simply invalidate all the UTXOs before 512666 when checking the validity of a transactions input.

4.Other Future plans (POS, block reorganization, etc.)

4.1 simple POS or simple master node

To maintain CDY blockchain network, someone should generate blocks. Mining uses electric power and someone should pay it. If ratio of reward pow get from block generation reward is small less of hash power will participate in block generation, I.e, less electric power. Thereby, CDY introduce POS or master node reward (Let it briefly POS reward). Block reward dilute share of coin holder, investor. By introducing POS reward the dilution can be reduced. If cdy blockchain has protection to 51% attack, POW hash-rate can be small. (Refer limit reorganization part)

4.2 block reward distribution

Block reward should be divided to miner, coin holder, promoter and developer. Direct cost for maintaining blockchain is electric power of mining rig and marketing cost. CDY premined 1% of coin. To increase user and to expand exchange site, marketing is important. For marketing, CDY set 5% of block reward and delegate to Bitcoin Candy Promoting

Association(BCPA). BCPA reward can be used to support creative writing like steemit. For developing coin software, cdy should make programmer join to cdy developing. After reward hardfork miner and pos reward ratio start from 1:2 and make it 1:5 after 3month and make it 1:8 after 6month. BCPA activity can make interrelation to real economy with cdy. BCPA activity would be chief property. BCPA divide reward manually to contributor's behavior. After reward fork, BCPA is more important than mining. Mining reward and BCPA reward is short term cost, so cdy holder should endure it. It is the reason of high POS reward.

	hardfork	3month	6month
miner reward	32%(1)	-> 16%(1)	-> 10%(1)
pos reward	62%(2)	-> 78%(5)	-> 84%(8)
BCPA reward	5%		
Dev reward	1%		

- Process 1 : Manual allocation except miner reward. Allocate Pos reward manually to holder with 1M and 10M coins. (Until coinex.com open withdrawal, 31%(half of pos reward) will be accumulated. If btc-alpha and coinex.com does not open withdrawal, 62%(all pos reward) will be accumulated. Accumulated reward will be distributed later. Nov 2018)
- Process 2 : Make automatic distribution by code to coin Holder with 1M and 10M.
- Process 3 : After making BCPA, Candy Team distribute to BCPA group every month or week. Continue manual process by candy team.

4.3 coin inflation

At 2018 year, coin inflation is about 4%. But after 2024 the inflation is under 1%. Coin price is very volatile. Volatile price can't be controlled by small supply ratio. Volatile price is property of cryptocurrency. After 2024, inflation of coin will be set to 1% for cdy.

reference : bitcoin inflation(number : million)

year	coin(m)	increase(m)	inflation%
2017	15.750	0.657500	4.1746
2018	16.408	0.657500	4.0073
2019	17.065	0.657500	3.8529
2020	17.723	0.657500	3.7100
2021	18.380	0.328750	1.7886
2022	18.710	0.328750	1.7571
2023	19.038	0.328750	1.7269
2024	19.370	0.328750	1.6972
2025	19.700	0.164375	0.8344

4.4 target market, target usage

Cdy coin should be used in real world. It can be done by marketing and expand usable site, retailer. By making Bitcoin Candy Promoting Association(BCPA), CDY delegate the marketing behavior. The cost can be given by block reward. BCPA help valuable writing, bloging. BCPA expand retailer acceptance.

Cdy cannot compete with BTC, BCH as store of value and medium of exchange. Cdy should focus on niche market. Many companies will introduce cryptocurrency to their mileage and points. But that crypto is under manipulation of each company. If Cdy make it easy for points and mileage, company may introduce cdy as points.

Small office like coffee shop can discount price by giving bitcoin candy. Because crypto is volatile in price, given cdy is similar to small lottery. Consumer can sell it at current price or put sell order at twice price. At first, BCPA offer cdy tickets(ex 1000cdy 5000cdy) to office owner. If office owner want many cdy tickets, he could buy cdy in the market. Cdy ticket is for consuming. This cost is covered by dilution of share especially of investor and holder.

4.5 limit reorganization

Small hash coin is under 51% attack. Cdy blockchain was reorganized by attacker several times. Cdy hardfoked from bitcoin cash. So cdy is distributed well comparing other coins. It is backgroud of high POS ratio. Pow is most fair method of coin distribution. So keeping pow and defending 51 attack, it can be done by limit the reorganization depth. Cdy dev team suggest the depth 6. After 6 confirmation, the transaction in the block cannot changed, cannot double spend. By this method, protect coin exchange site.

- Process 1 : Manual manipulation by official pool and block explorer and Dev team nodes. Invalitate selfish mining block(hidden mining without broadcast to network). This method splits selfish miner chain from official chain.
- Process 2 : Automatic limit the reorganization by code. (https:// github. Com/ bitcoincandyofficial/ bitcoincandy/ pull/ 10)
- 16 Nov 2018 BCH_ABC announced Auto-finalize block once they reached a certain depth (by default 10). This approach is same idea, so cdy would accept this approach.

4.6 BCPA activity

Block reward would set 5% to BCPA(bitcoin candy promoting association). BCPA conduct marketing to real world. For example : add cdy to coin exchange site, make meetup in several city. For each activity need cost. This cost can be paid by block reward.

Another method of marketing is direct carrot to digital writing like steemit. BCPA member

find good writing in blog, thread. And make link about cdy coin carrot. If the writer reply with cdy coin address, BCPA give coin to him. This activity contribute to real world from crypto world. BCPA activity is core in success of cdy coin. And BCPA member also get paid cdy for his activity.

BCPA, mining and dev reward are short term cost. Cdy set coin number increasing to 1% finally. This short term cost is paid by coin traders, holders. Cdy make share of total coin almost same for Pos reward investor. But many coins are not claimed to cdy from bch, Pos investor share will increase. Minimum holding would be 1 million, and standard holding would be 10 million. Trader, investor can be BCPA member. Investing cdy and joining to BCPA activity would be best situation to cdy.

Cdy is small cryptocurrency. If cdy have good ecosystem and benefit to real world, cdy is precious and can grow gradually. BCPA is core part in cdy crypto.