DRAFT

CCBN: a Cross-Chain Block Notarization Protocol

Proposing a Decentralized Notarization Approach to Thwart Double Spends Made via Secret Mining 51% Attacks

DRAFT

Contributors: Edward Iskra, Hang Yin, JK

Abstract:

Various approaches to mitigating the risk of Double Spends created via short-term Secret Mining 51% attacks have been proposed. Most have shortcomings in safety and effectiveness or else work against the principle of Decentralization. We present here a fully decentralized scheme for the public notarization of blocks to secondary blockchain(s) to establish a public PoW-weighted reference that can be used to arbitrate a chain split in favor of legitimate miners over a Secret Miner, or else to alert blockchain users that there is an ongoing chain split and an imminent attack. Either way, losses due to Double Spends are avoided.

Overview:

Under CCBN, BTG blocks will be notarized to an independent blockchain (from our perspective, this becomes a ***Notarychain***), creating an external block timing reference. These ***Notes*** are complete BTG block headers along with their mined solution hashes, which cannot be easily faked. They will be recorded in transactions sent to a single easy-to-monitor address on the Notarychain, and each Note's age can be measured by the passage of blocks on the Notarychain. The small notarization cost will be inconsequential but will inhibit spam.

If a Secret Mining attack releases many blocks at once, BTG nodes with CCBN will not immediately switch to the longer chain. Instead, they will calculate an age-based ***Weight*** for the blocks in each chain and will only abandon the current chain for one with greater Weight. The formula for weight is quadratic, based on each note's ***Depth*** in the Notarychain raised to a power. A Note in the most recent Notarychain block (the tip) has a Depth of 1, and one recorded nine blocks earlier has a Depth of 10. If the Weight formula uses a power of 2, the formula for a single block's Weight would simply be its Note's Depth squared - so the Note at Depth 10 would have a Weight of 10 ^ 2 = 100. The Weight of a chain is the sum of the Weights of the Notes of its blocks.

Because of the quadratic formula, blocks Notarized earlier gain Weight much faster than blocks Notarized later. The honest chain, being Notarized as soon as blocks are found, will therefore gain Weight quickly, while an attacker's chain being mined in secret gains no Weight. Even if the attacker mines a very long chain of blocks, if they did not Notarize them until broadcasting them on mainchain, the honest chain will have a great advantage in Weight because of earlier Notarization.

An attacker who mines in secret, without Notarizing, will find that their chain is simply not accepted by BTG nodes. On the other hand, if an attacker Notarizes their blocks as they find them, they lose their secrecy.

The network of BTG Nodes can watch the Notarychain in real-time and a chain fork can be detected even if the attacking blocks have not yet been published on mainchain. This puts a Double-Spending attacker into a dilemma: if they Notarize their blocks as they mine them, then nodes can see the fork being created. Their Secrecy is lost. Parties to large transactions (like Exchanges) can see the fork and the attacker's deposits will not be accepted. On the other hand, if the attacker does not Notarize their blocks and remains Secret, then the Exchange will accept the deposits... but when the attacker later releases their Secret blocks, they won't have sufficient Weight and will be rejected by the network, preventing the double spend from working.

Either way, the attacker can not profit on a Double Spend.

A determined Secret-Mining attacker may try to continue mining and notarizing their chain after broadcast, but BTG's effective DAA (Difficulty Adjustment Algorithm) will limit how many blocks of advantage they can gain, and the quadratic growth in the Weight of the honest chain ensures that the attacker is unlikely to ever "catch up."

Summarizing: CCBN rewards blocks Notarized immediately in public with more Weight so that Secretly mined blocks cannot become the mainchain. If an attacker tries to accumulate Weight by Notarizing their blocks as they are mined, the "Secret" part of Secret Mining is lost, everyone can see the threat, and transactions are not accepted. Either way, attempts to Double-Spend are thwarted. Relying on Notarizations posted to an independent blockchain is an effective decentralized method for establishing block age in a way that is impervious to any attacks on the mainchain.

CCBN only comes into play on mainchain during a chain split and has no impact on normal mining. During a normal (chance) chain split, if both forks are notarized, CCBN causes no change to normal Nakamoto Consensus. All nodes observing CCBN during a split converge on the same answer almost immediately, so mining always proceeds smoothly. Legacy nodes that don't run CCBN may temporarily follow a Secretly Mined fork (or an unnotarized fork) under the Longest Chain rule, but if the majority of miners and mining pools use CCBN, the honest chain

will continue to grow and legacy nodes will eventually re-converge to the same (correct) mainchain as the CCBN nodes.

This means that CCBN is essentially a Soft Fork whose only effect is to provide a secondary consensus mechanism for resolving chain splits. Because it's a soft fork, it will not create any change in mining, it will not require legacy nodes to be upgraded, and will not result in a chain split or new "forked" coin. As long as a majority of mining pools are using CCBN nodes, the protection against Secret Mining is in place. As long as Exchanges use CCBN nodes, their protection against Double-Spends is in place.

All this is accomplished in a fully decentralized and transparent manner - no power is delegated to masternodes, notarynodes, federations, consortia, or other "elevated status" parties, so CCBN is completely consistent with the principles of Decentralization.

In addition, CCBN's use of independent blockchain networks to communicate information makes natural chain splits, Sybil attacks, and other node isolation attempts obvious. A BTG node that is fully isolated from the BTG network can still be aware of the real mainchain's status by observing notarizations.

Several refinements, described later, increase the protection offered by basic CCBN and allow Exchanges to enjoy excellent protection from Double-Spends with fairly modest deposit confirmation requirements.

Discussion:

It's important to recognize that a prolonged 51% public mining attack, commonly called a "51% Attack," is fundamentally different than a short-term 51% private mining attack, which we here call a "Secret Mining" attack. A 51% attack means taking over an entire network to unfairly collect all the rewards or to censor transactions, and can be done over a long term, while a Secret Mining attack is used in short-term attacks in order to Double-Spend. The Double-Spend is accomplished by delayed broadcast of a secretly-mined chain in order to revert a large transaction made on the mainchain (most commonly a deposit to a cryptocurrency exchange.) While both the 51% Attack and the Secret Mining Double Spend attack exploit the 51% attack vector of Nakamoto Consensus, they are fundamentally different attacks in terms of how they work and their economics.

The presence of intermediary "hashpower markets" which centralize miner hashpower and sell it to the highest bidder allows attackers to purchase large amounts of hashpower for very short periods of time, but due to market economics, large purchases typically entail paying above-market rates for that power. (Buying a large share of the market's available power necessarily drives up the market price.) The end result is that buying enough hashpower for long enough to run a 51% attack on a coin like BTG ends up costing more than the rewards earned from the mining, for a net loss of funds if one only considers the gains from the mining.

To be economically viable, a conventional 51% Attack requires sustained control of large amounts of network hashpower *at a cost that is lower than* the gains provided by the earned mining rewards. This is generally not possible using hashpower markets where buyers pay a premium for anything over modest amounts. There are also other inherent disincentives to such an attack discussed elsewhere (including the fact that taking over a chain is a publicly visible event likely to devalue the coin on the markets, reducing the attacker's gains.)

The economics of the short-term Secret Mining 51% attack to achieve a double-spend are entirely different: with a well-chosen target, the value gained from the Double-Spend can be *much* larger than the attack cost, making it economically viable to dramatically “overpay” to purchase hashpower just long enough to conduct the attack. The general cost of hashpower in a given chain tends to follow the value earned from mining the chain, and an “overpayment” to buy mining power may mean paying as much as double that rate. For an attacker with sufficient funds to use in Double-Spends, the gain from defrauding exchanges via reversed deposits can be *several orders of magnitude greater* than the cost of mining to mount the attack. This means the mining can be performed “at a loss” and still be part of a profitable Double-Spend attack.

In essence, a conventional 51% Attack is an attack on the chain, while the Secret Mining attack is really an attack on the recipient of a large transaction (where the chain is just a means to invalidate a deposit transaction.)[1]

It’s important to note that a successful Double Spend of coins deposited to an exchange generates no profit for the attacker - they have merely regained control of the previously-deposited coins, which makes for a zero sum and no gain. In order to profit, the attacker must trade on the exchange and/or withdraw different coins than those which were deposited. The reason that exchanges may fail to prevent such withdrawals is that they cannot know when a 51% attack’s block reorganization is imminent. If the privately mined blocks are kept Secret until the trading and withdrawal are completed, the exchange has no knowledge or recourse until it is too late. CCBN solves this.

Previous proposals[2] have addressed the “delayed broadcast” aspect of such a Secret Mining attack by imposing a “penalty period” before blocks are accepted by the chain which would

---

[1] It’s also potentially possible for an attacker to try to profit by taking a large “short” position on the crypto markets in hopes that their Secret Mining attacks cause reputational damage to a crypto project, causing a drop in price and a gain on the “short” position - but the point still holds: in these attacks, the actual mining aspect of the attack can be performed “at a loss” because the real profit comes from activity against/on the Exchanges. In practice, short Secret Mining attacks have had little impact on short-term market prices, so the “short position” attack is unknown on BTG, whereas Double Spend attacks have been observed.

[2] See *A Penalty System for Delayed Block Submission by ZenCash*, https://www.horizen.global/assets/files/A-Penalty-System-for-Delayed-Block-Submission-by-ZenCash.pdf

otherwise cause an immediate reorganization. The hope is that the exchanges will gain enough time during the delay period to halt withdrawals, or else that the delay period raises the cost to mount such an attack high enough to dissuade attackers. In our analyses, neither of these are likely enough to be true to afford sufficient defense.

Other proposals include various types of “checkpoints” - choosing a block count after which blocks are considered immutable. Such approaches introduce some risk of chain splits.

The CCBN proposal outlined here takes a different approach, addressing the “Secret” aspect of an attack by giving extra weight to blocks which are shared in a more highly public and verifiable manner: Notarization to another chain. Notarization is not a new concept, but most existing notarization schemes rely on a federation, syndicate, masternodes, or other types of “authoritative” nodes,[3] which are generally incompatible with a fully decentralized system.

The CCBN protocol leverages an independent blockchain as a Notarychain to share block information through an independent network of nodes, allows for a fully decentralized notarization scheme, and does not require any special authorities.

High-Level Description of CCBN

1. **Recording**: BTG block headers with hashed solutions are written (“Notarized”) to another chain (making it a “Notarychain” for BTG.)[4]
2. **Monitoring**: Miners and Exchanges run CCBN-enabled Full Nodes which monitor the Notarychain as well as the BTG mainchain. (Optional for others.)
3. **Weight**: Blocks gain Weight based on depth of their Notarizations in the Notarychain via quadratic formula (such as weight equals depth squared.)
4. **Mainchain Split / Reorganization**: if a new chain fork appears on the BTG mainchain, CCBN nodes do not switch to the longer chain immediately - instead, they compare the Notarized Weights of the two chains. Nodes *only* switch if the new chain is longer *and* has greater Weight.
5. **Notarization Chain Split Detection**: if Notes appear in the Notarychain for blocks which are not on the main blockchain, CCBN nodes report the potential fork immediately, alerting users to stop respecting deposits and payments until any risk is past.

To understand exactly how this prevents Double-Spends from Secret Mining 51% attacks, and what potential risks remain, we need to understand exactly how the attacks work.

[3] Komodo’s dPoW is one such system; see: *Delayed Proof of Work (dPoW) Whitepaper*, https://github.com/SuperNETorg/komodo/wiki/Delayed-Proof-of-Work-(dPoW)-Whitepaper

[4] A Full BTG block header with Solution hash is currently 241 bytes.

How Double-Spends through Secret Mining Work

A Secret Mining attacker uses more hashpower than all the honest miners combined, usually accomplished by renting power on a market. The attacker then does the following:

- mines a Secret chain (forking off of the Mainchain) - these blocks are mined, but not broadcast (not public)
- Spends the same BTG coins on Mainchain and on the Secret chain:
    - on Mainchain, sends the coins to an Exchange, while
    - on Secret chain, sends the same coins to their own wallet
- waits for Exchange to accept deposit on mainchain (while continuing to mine in Secret)
- trades BTG on Exchange for BTC (or other coin) and withdraws the BTC
- finally broadcasts Secret chain to cause chain switch ("reorganization"), effecting a Double-Spend of the originally deposited coins

The final step only works because the Secret Chain, having been mined with more hashpower than the honest chain, has more blocks than the mainchain. The standard rule is that the longest chain wins.[5]

How does the attacker profit from this? They deposit *x* BTG, and then effectively cancel that deposit, regaining their *x* BTG. This is no gain (the sum is zero). However, between the deposit and the cancellation, they traded for and withdrew another coin (BTC). This constitutes the lost value for the Exchange.

Thus, a successful attack depends on the following:

1. The exchange must accept the BTG and allow trading/withdrawal of BTC
2. The attacker must later cause the chain to switch, invalidating the BTG deposit
3. The gain from #1 must be larger than the cost of causing #2.[6]

[5] Technically, the Secret Chain has *more work* than the honest chain, not *more blocks*. The node calculates the total "work" in each chain based on blocks and their mining difficulty and follows the chain with the most work. For the attack lengths that are relevant here, the outcome is the same either way, and the common phrase "longest chain" is used for simplicity.

[6] The attack is run just long enough for the Exchange to accept the deposit, which limits the cost of renting power for the attack. If an Exchange demands five more blocks, the attacker simply mines for five more blocks. Exchanges compete for business and cannot demand excessive numbers of blocks for Confirmation of a deposit. Meanwhile, there is no inherent limit to the size of the attacker's deposit, subject to the Exchange's risk controls.

How CCBN Protection Works Against this Attack

Scenario One: attacker mines Secret chain, **does not Notarize**, spends same coins on mainchain and Secret chain, waits for Exchange to accept deposit on mainchain, trades and withdraws, and releases Secret chain.

> **Result**: when miner releases their Secret chain, the network *does not switch* to the new chain because the new chain has no Weight. Public mining continues normally on existing mainchain and the **attempted 51% Attack fails**.

Scenario Two: attacker mines Secret chain, **Notarizes blocks immediately**, spends same coins on mainchain and Secret chain, waits for Exchange to accept deposit on mainchain

> **Result**: Nodes see chain split in Notarizations on the Notarychain and Exchanges immediately freeze deposits, even though nothing is visible on mainchain. With their deposit frozen, the attacker cannot trade (or withdraw). Exchanges do not resume clearing deposits until any potential split is resolved on Mainchain. Regardless of how the split is resolved, Exchanges are at **no risk from Double-Spends**.

Summarizing Scenarios One and Two: if the attacker does not Notarize, the 51% Attack will fail when the nodes ignore the blocks. If the attacker Notarizes, the Double-Spend will fail when the Exchanges ignore their deposits.

Scenario Three: attacker mines Secret chain, **does not Notarize immediately**, spends same coins on mainchain and Secret chain, waits for Exchange to accept deposit on mainchain, trades and withdraws, releases Secret chain, and **then begins Notarizing and continues mining** the attacking chain in hopes of gaining enough Weight to cause a chain switch.

> **Result**: the honest blocks Notarized earlier gain Weight much faster than the attacker's blocks Notarized later (because of the quadratic weighting). Delayed Notarizations from the Secret chain are at a tremendous disadvantage. Simple rules for recognizing Notarized depths (explained below) make it extremely unlikely the attacker can gain more weight than the honest chain. With sufficient blocks of delay (the length of the Exchange's Confirmation requirement), the **odds of an attack succeeding become tiny despite enormous cost**.

In addition, Notarization to two or more Notarychains simultaneously ensures it is **prohibitively expensive for an attack to have even a tiny chance of succeeding**.

Unlike the first two scenarios, the likely outcome of the Third scenario is not always obvious. The remainder of this Whitepaper focuses on this Third scenario and the defensive strength of CCBN in various configurations. We should first formalize the variables and methodology for Weight.

Determining the Weight, W

The weight of a chain of Notarizations, W_{chain}, is the sum of the notarized weights of the blocks in a well-formed blockchain, starting from the first forked block, FFB, and ending at the chaintip, tip:

$$W_{chain} = \sum_{i=FFB}^{tip} W_{block(i)}$$

The weight of a single block in the Notarychain, W_{block}, is a function of the Depth on the Notarychain of the Notarization, $Depth$. The formula for Weight is then:

$$W_{block} = Depth^{WP}$$

Where the Weighting Power, WP, is a constant of 2 (or more.) The result is quadratic growth in the weight of a block as it ages when more Notarychain blocks go by.

Adjusting the Weighting Power, WP

The Mainchain's weight advantage comes from Notarizing blocks sooner than the Secret chain. This means we can make the Weight advantage larger by increasing WP to a larger number, making an attack more difficult. However, modeling this shows that while a larger WP makes a successful attack require more blocks, it does not necessarily make a successful attack impossible if the attacker is willing to continue to overpay to continue 51% mining.

This is relevant when one considers how a lengthy attack might defeat the CCBN defense. While waiting for the required Confirmations to pass, the attacker mines a larger number of blocks than Mainchain, and after Confirmation, they can all be Notarized immediately. These Notes are all newer (and thus lower Weight) than the honest Notes, but there are a greater number of them. As both chains continue to mine and all of the Notes in question become "old", the relative difference in the age of the oldest Notes becomes less significant than the difference in number of Notes, and the total weight of the Attacker's blocks eventually becomes greater than the honest blocks.

This can be overcome by using an exponential instead of a quadratic formula, replacing $Depth^{WP}$ with $Depth^{Depth}$. However, this creates another problem: if an attacker can mine and notarize a single block on the Notarychain a block before the mainchain, that single block will forever have a notarized Weight larger than all subsequent mainchain blocks combined!

Obviously, we cannot use an exponential, but this brings to mind a danger with quadratics if using *too high* a Weighting Power: a vandal can mine one or two blocks ahead of mainchain, notarize them, and then stop. If the Weighting Power is *very* high, it may take a long time for the honest chain's Weight to overcome it. This act of Notarization "vandalism" forces all Exchanges to close their wallets for fear of attack, even though the vandal has stopped mining. This disrupts activity while everyone waits for many blocks to pass before the chain can be considered safe again. Repeated acts of such "vandalism" can interrupt business so much that Exchanges are tempted to turn off CCBN and expose themselves to risk of attack. The higher the WP, the greater the threat to the liveness of the CCBN mechanism.

So, a low WP offers less defense against 51% attacks, while a very high WP can threaten liveness. Proper selection of an appropriate WP for a given Notarychain is important to hit the desired balance. Modeling shows diminishing returns from increasing WP - that is to say that the defense improves at a slower and slower rate as WP gets higher and higher. In most cases, a number moderately greater than 2 or 3 improves defense significantly without much threat to liveness.[7]

Recognition Rules

We can further improve the defense when we analyze possible failures. Note that even a very high WP merely delays an attack's success instead of preventing it if the Exchange chooses a very low number for the Confirmation requirement (much less than 10). This problem arises because the attacker can notarize many blocks immediately after the required Confirmations have been met; afterwards, they may need to mine for a long time for those many blocks to age, but it is conceivable. We can prevent immediate notarization of a large number of blocks by *Metering* the rate at which they are Recognized as valid Notarizations. Of course, we don't want to restrict Notarizations recorded before an attack is made - those blocks are necessarily honest blocks and the Notarizations should be respected immediately, so the Metering rule only comes into play *after* an attack has begun.

Metering the Recognition of Blocks after the Chain Split is Detected

To recognize where the attack begins in the Notarychain, we watch for a Conflicting Notarization - that is, a second Note in the Notarychain for the same mainchain blockheight.The first Note for that height comes from the honest mainchain, while the second (Conflicting) Note comes from the attacking chain. Call the Notarychain block in which the Conflicting Notarization occurs the $CBlock$. Before the $CBlock$**,** Notes are recognized immediately and assigned Depth based on the Notarychain block they are in, with one recognized in the actual $CBlock$, but all further Notes are assigned Depths at a metered pace. We are replacing $Depth$ with Recognized Depth, or $RDepth$. We want to Meter out the blocks at pace which is reasonable for an honest chain.

[7] Estimating the magnitude of the threat to liveness of a given WP is an area for further research.

The simplest metering (linear) would be to Recognize one Note for every *n* Notarychain blocks, where *n* is the expected ratio of Notarychain blocks to BTG blocks. (We are measuring time in Notarychain blocks.) For example, if we are recording Notes to the Litecoin blockchain, we expect an average of 4 Litecoin blocks (average 150 seconds per block) to every 1 BTG block (average 600 seconds per block), so the ratio is 4/1, and *n* = 4. We would therefore Recognize only one new Note for every 4 Notarychain blocks that go by; any remaining Notes would be queued for Recognition when more Notarychain blocks pass.

Modeling this kind of linear metering shows an increase in the effectiveness of the defense, but there are still a significant number of failures in the blocks soon after the attack begins when Exchanges choose a low Confirmation requirement. Close inspection shows that many of these failures happen when the mainchain happens (by natural chance) to run slowly after the attack begins. The attacking chain, with many Notes queued up, will get a Note recognized at every opportunity, but when the honest chain runs slow it will miss opportunities for Notarization and fall behind. Essentially, the failures happen because of natural variance as honest blocks are found. We can significantly limit this by replacing our linear metering with metering based on the Standard Deviation (*σ*) of the block time for BTG. We call this SD-Based Metering.

- In SD-Based Metering, we accept a single Note at the $CBlock$.
- The first Note after the $CBlock$ will be accepted after a time equal to the mean BTG block time (*μ*) plus two times the standard deviation (*σ*) of two BTG blocks.
- The second Note after the $CBlock$ will be accepted after the mean time for two BTG blocks plus two times the standard deviation of two BTG blocks.
- Generalizing, the *n*th Note after the $CBlock$ will be accepted after the mean time for *n* BTG blocks plus two times the standard deviation of *n* BTG blocks, $\mu_{nBTG} + 2\sigma_{nBTG}$

In general, we will find a given block within mean plus two standard deviations ($\mu + 2\sigma$) for such a chain the great majority of the time. For a single block or two, this may seem like a large variance, but as the series of blocks becomes longer, the value will get closer and closer to the mean for *n* blocks.

Let's call this the Chain Normalization Delay function, CND. In terms of n Notarychain blocks:

$$CND = (\mu_{nBTG} + 2\sigma_{nBTG}) / \mu_{Notarychain}$$

where μ_{nBTG} is the mean time to find n BTG blocks, σ_{nBTG} is the standard deviation of the time to find n BTG blocks, and $\mu_{Notarychain}$ is the mean time to find a block on the Notarychain.

Once we include the Chain Normalization Delay for recognizing Notes after the $CBlock$, there are few instances where CCBN will fail for a normally functioning blockchain because of random chance. Unlike our previous scenario where we simply raised the Weighting Power, WP, we are not simply delaying failures - we are actually eliminating most of them. With a modest WP and a modest Confirmation requirement, an Exchange can be extremely safe. However, we need to make one last adjustment before moving on.

Dealing with Selfish Mining

If used as described above, CCBN could be used to perform Selfish Mining[8] with a γ (gamma) of 1.0! A Selfish Miner could Notarize their unpublished blocks; whenever the honest chain finds a block, the Selfish Miner could publish their block and always win the “race” because CCBN would give the Selfish Miner’s block more weight.

To prevent such abuse of CCBN to enhance Selfish Mining, we introduce a two block delay for the application of CCBN. In other words, for block reorganizations of one or two blocks (which can also happen by natural chance), CCBN is not invoked to determine Weight; rather, the node follows the normal Longest Chain rule. For a block reorganizations of 3 or more blocks, CCBN is invoked.

This delay effectively mitigates against the risk of enhancing Selfish Mining.

Now we can put together all the Recognition rules.

Summarizing Note Recognition and Weight Calculation for CCBN

Consolidating the processes described above, we end up with the following method to properly ***Recognize*** Depths for each Note on a Notarychain:

1. If a potential reorganization of more than two blocks is detected on mainchain (based on traditional Nakamoto Consensus), then CCBN is invoked to determine whether to switch. Otherwise, the Longest Chain rule is followed.
2. Identify the $CBlock$ in the Notarychain:
    a. The $CBlock$ is the first Notarychain block with a Conflicting Notarization within it.
3. Identify the Notarizations of the contending chain segments:
    a. Each is a well-formed chain of Notes in the Notarychain, both beginning at the BTG block height of the Conflicting Notarization and extending to the most recently seen Note in the Notarychain.
4. Find each block’s $RDepth$.

[8] See *Majority is not Enough: Bitcoin Mining is Vulnerable*, aka the “Selfish Mining” paper, https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf

a. For Notes before the $CBlock$ and for the first note in the $CBlock$, Recognize the Depth for each Note equal to the depth of the Notarychain block where it is recorded.
b. Beginning at the $CBlock$, meter out additional Recognized Depths using the CND (Chain Normalization Delay). Some notes are likely to initially be queued with no depth, unable to be Recognized until more Notarychain blocks pass.

Now that each CCBN Note has a Recognized Depth, $RDepth$, we can find the Weight of each block:

$$W_{block} = RDepth^{WP}$$

And find the Weight of each chain:

$$W_{chain} = \sum_{i=FFB}^{tip} W_{block(i)}$$

This process radically improves the level of protection against a 51% attack for a given Weighting Power, WP when the number of Confirmations is relatively small (<10), even when the attacker can mine with many times the hashpower of the honest miners[9]. The right WP to use is best determined by careful modeling of attacks using simulated blockchains. It's critical that the simulated blockchains accurately model the actual DAA (Difficulty Adjustment Algorithm) in use for the mainchain. It's also critical to have accurate figures for the standard deviation of the blockchain.

It's also worth noting that for a chain like BTG, which has a per-block DAA, one should not use the common formula for the standard deviation of an exponential distribution (which is generally used to estimate Bitcoin's standard deviation.) Because the DAA compensates for short-term fluctuations, BTG's standard deviation is significantly smaller than Bitcoin's for chains longer than just a few blocks.

It's also important to note that some of the strong protection afforded by CCBN depends on a well-functioning DAA. Because the DAA will be raising the Difficulty, the attacker will not be able to extend their block lead indefinitely, even if they can muster many times the mainchain's hashpower. A chain where the Difficulty can go without adjustment for hundreds or thousands of blocks (like Bitcoin) would still gain protection from CCBN, but it would require a higher number of Confirmations for the same degree of extra security.

[9] The <10 figure is true for BTG because BTG has an effective DAA. For a blockchain with a slow-to-respond DAA, protection is substantially reduced. Other projects will need to carefully model CCBN to determine what constitutes a "relatively small" number of Confirmations for their chain.

Notarizing to Multiple Notarychains

Notarizing to more than one Notarychain at the same time further improves the protection. This happens because any blockchain can run faster or slower than normal by natural chance. Depending on which chain runs faster/slower, the defensive power of CCBN may be reduced.

When Notarizing to two or more Notarychains, the basic CCBN rule changes:

> **Mainchain Split / Reorganization**: if a new chain fork appears on the BTG mainchain, CCBN nodes do not switch to the longer chain immediately - instead, they compare the Notarized Weights of the two chains on each Notarychain. Nodes *only* switch if the new chain is longer *and* has greater Weight on *all* Notarychains.

In this way, a single "misbehaving" notarychain doesn't weaken the defense of CCBN. Interestingly, modeling has shown that notarizing to both a fast and a slow chain provides better protection than two fast or two slow notarychains, as they guard against slightly different failure modes.[10]

Obviously, using more than one Notarychain also protects against one of the Notarychains suffering a catastrophic failure, and allows time for the community to adopt another Notarychain.

Closing Observations

Effective Defense: CCBN, with the recognition rules in place (including the SD-based Chain Normalization Delay), with well-selected Notarychains and properly tuned weighting powers, provides highly effective protection against Double-Spends powered by 51% attacks. Modeling shows that if an Exchange requires merely 6 confirmations, even an attacker willing to mine for 100 blocks at several times the mainchain's hashrate will only succeed 1% of the time.

> Conditions: notarizing to a 30-second Notarychain with WP = 4 and a 600-second Notarychain with WP = 6, attacker uses 3x normal nethash, mainchain drops to ½ normal nethash (for a 6x advantage to the attacker,) attacker mines for 100 mainchain blocks (about 150 attack blocks). If the exchange requires 6 confirmations, the attack will fail about 99.05% of the time.

Resistant to Attacks: While it's theoretically feasible for an attacker to employ 51% attacks against both the mainchain and all the Notarychains simultaneously, the cost of such an attack is dramatically higher than attacking a single chain - both in terms of hashpower and in terms of

[10] If the Notarychain runs unusually slowly *before* the CBlock (which limits the honest chain's ability to develop a lead in Weight), then a faster Notarychain does better than a slower Notarychain. If the Notarychain runs unusually quickly *after* the CBlock (which allows the attacking chain to fit in extra Notarizations), then a slower Notarychain tends to do better than a faster Notarychain. A blend provides the best protection.

the skill/infrastructure to attack multiple chains at once. This is especially valuable to a small new chain which can Notarize to a larger, hard-to-attack chain.

Resistant to Network Issues: Because CCBN leverages secondary blockchains run by independent networks of nodes, it has resistance to certain disruptions caused by transient network problems or malicious Sybil attacks on the mainchain's network. A CCBN node effectively sees BTG block headers being broadcast through two or more separate networks - the mainchain and the Notarychains - adding layers of network redundancy.

Resistant to Notarychain Issues: If Notarychain suffers a disruption, it does not affect the mainchain in any way - the mainchain's network *only* references the CCBN Weights in the rare event of two contending chains. During the short time a Notarychain may be disrupted, the protection of CCBN is reduced, but even this small exposure is limited by choosing Notarychains that use well-proven technology and are highly stable, and by performing Notarization to more than one chain simultaneously.

Deterministic: the protocol is strictly deterministic, meaning that all CCBN nodes will have the same outcome when given the same information. In contrast, time-of-arrival based schemes may have significant subjectivity and increase the risk of spontaneous (or malicious) chain split.

Immediate: All CCBN-observing nodes will assume the same state *immediately* after information is communicated, whether a block delay is caused by a 51% attack, a Sybil attack, or a transient network problem. In contrast, "penalty delay" systems may take many hours or days to fully resolve after a protracted attack or after an accidental chain split.

Not a replacement of Nakamoto Consensus: the fundamental consensus mechanism remains the same: longest chain wins. CCBN only comes into play to arbitrate chain splits which are longer than two blocks.

Soft Fork: CCBN is effectively an optional set of functions to be enabled on Full Nodes. It does not introduce any changes to the underlying blockchain or PoW protocols, so no changes are required for mining the chain, no changes are required for transacting on the chain.

CCBN functionality may be coded into the mainchain's full node code, or may be coded as auxiliary software.

Longest Chain versus Accumulated Work: we refer to the "longest chain" in this paper for simplicity. In practice, most blockchains respect the chain with the highest Accumulated Work. When referring to the mainchain, "Longest" refers to the leading contending chain of a split based on the rules of the given mainchain. The mechanics of CCBN are unaffected by this difference.

Selected References

Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf

A Penalty System for Delayed Block Submission by ZenCash, https://www.horizen.global/assets/files/A-Penalty-System-for-Delayed-Block-Submission-by-ZenCash.pdf

Bitcoin Cash ABC's rolling 10 block checkpoints, https://blog.bitmex.com/bitcoin-cash-abcs-rolling-10-block-checkpoints/

Delayed Proof of Work (dPoW) Whitepaper, https://github.com/SuperNETorg/komodo/wiki/Delayed-Proof-of-Work-(dPoW)-Whitepaper

Majority is not Enough: Bitcoin Mining is Vulnerable⁎, aka the "Selfish Mining" paper, https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf

Detective Mining: Selfish Mining Becomes Unrealistic under Mining Pool Environment, https://eprint.iacr.org/2019/486.pdf

A Deep Dive into Blockchain Selfish Mining, https://arxiv.org/abs/1811.08263