# Chainweb Protocol Security Calculations

Monica Quaintance, monica@kadena.io Will Martino, will@kadena.io

WORK IN PROGRESS - DRAFT v7

## 1 Introduction

In this paper we consider the Chainweb protocol[1] and the inherent security in its configuration. Herein we provide an analysis of the scenario of an adversary attempting to generate various configurations of an alternate chainweb, or braid, faster than the honest network; the proof is analogous to the probabilistic double-spend attack analysis in the Nakamoto Bitcoin paper[2] expanded to the additional dimension of multiple chains under the Chainweb protocol. We present three versions of the proof in successive levels of strictness that demonstrate the viability of Chainweb as an extremely secure protocol.

## 2 Overview of the Chainweb Protocol

Chainweb is a parallel-chain architecture which can combine hundreds to thousands of Proof-of-Work blockchains, increasing throughput to 10,000 transactions per second and beyond. The Chainweb network transacts a single currency using trustless Simple Payment Verification (SPV) cross-chain transfers. Chains incorporate the Merkle roots of each other to enforce a single super chain offering an effective hash power that is the sum of each individual chain's hash rate. Additional detailed explanation of the Chainweb design is presented in the general paper.

The Chainweb network is comprised of multiple independent peer blockchains minting distinct coins of the same currency. Each chain incorporates a subset of other peers' Merkle roots in its own block hashes. A Chainweb braid arranges chains in a graph configuration such that a given chain samples a fixed number of peer chain Merkle roots for each block (its *degree*, or $d$) but can reach any other chain in some fixed maximum number of hops

[1] Martino, Quaintance, Popejoy 2018

[2] Satoshi Nakamoto 2008, https://bitcoin.org/bitcoin.pdf

(the *diameter* of the graph, or $\Delta$). With each increase in block height more chains are referenced transitively until with generation of block height equal to $\Delta$ the entire network has been referenced.

Solutions to the *degree-diameter problem* define valid graph configurations for a Chainweb braid (such as the Peterson graph) where the desired degree and diameter determine the size of the network (the *order* or $\Omega$).

## 3 Objective of the Adversary

In our analysis we consider the double-spend attack in which an adversary works in secret to construct an alternate version of the network, or braid, in which she has not spent her coin. Even such an adversarial braid is generated it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the adversary. An adversary can only try to change one of her own transactions to take back money she recently spent.

As in the Nakamoto paper, a prudent recipient of coin will wait until at least $z$ block height has been added to a chain before accepting confirmation of a transaction in a given block. For a single chain this practice results in generation of $z$ blocks; for a Chainweb braid this practice will result in the network generating a number of blocks equal to $\Omega \cdot z$ blocks, or a *layer* of the braid for every increase in block height.

Our analysis will consider three scenarios, each one a tighter bound on the probability associated with the strategy of the adversary. The first scenario is one in which an adversary attempts to recreate the entire braid in parallel to the network. The second scenario describes the adversary creating only the blocks that directly reference the block that contains her fraudulent transaction, or the Merkle cone (defined below) of her fraudulent block, for arbitrarily large values of $z$. The final scenario describes the case in which an adversary generates the Merkle cone of her fraudulent block for a value of $z$ equal to $\Delta$, the diameter of the graph.

This final scenario will present the best probability for the attacker to dominate the network with regard to the fraudulent block, yet we will show for several graph configurations that even under this tight bound the likelihood of success is so infinitesimally small as to effectively be impossible.

# 4 Scenario 1: Full-braid replacement

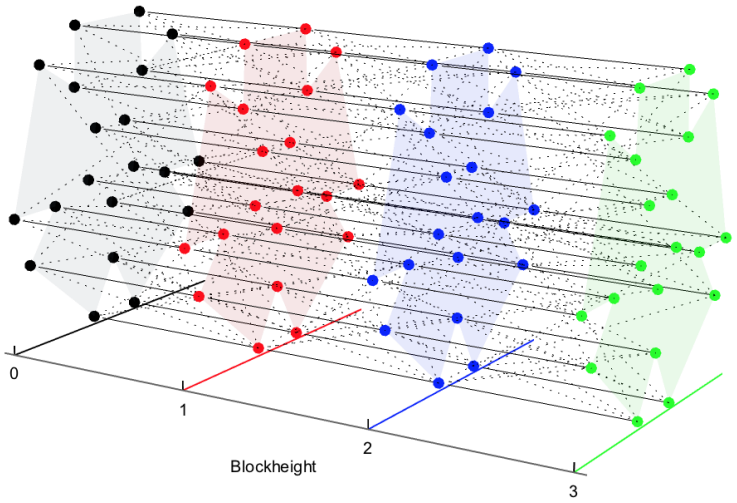


Figure 1: Full Chainweb Braid

Here we consider the scenario of an adversary trying to generate an entire alternate fork of the Chainweb, or braid, faster than the honest network. For every increase in block height, the number of blocks generated across a chainweb is equal to the mass $\mu$ of a layer, which for the full braid is the order $\Omega$. As in the bitcoin analysis, a receiver will wait until the transaction in question has been added to a block and $z$ blocks have been linked after it. In the chainweb case waiting $z$ additional blocks is equivalent to waiting for $z$ full layers of the chainweb to be generated, or $\Omega \cdot z$ blocks, which in the case of full-braid replacement must also be generated by the adversary.

Given the independent nature of the cryptographic hash, we make the simplifying assumption that from a probabilistic perspective mining the blocks of a layer either simultaneously or sequentially will result in the same likelihood of aggregate success; in reality, a rational adversary will prefer to distribute her hash rate equally across chains to reduce the likelihood of self-collision (adversary mining the same block with more than one compute resource).

As in the Calculations section of the Nakamoto paper, the race between the honest network and an adversary can be characterized as a Binomial Random Walk, and the probability of an adversary catching up from a given deficit is analogous to a Gambler's Ruin problem beginning from the fraudulent block and proceeding to the confirmation block. The general

probability an adversary catches up with an honest chain or chains is:

$p$ = probability an honest node finds the next block

$q$ = probability the adversary finds the next block

$q_{\mu(z)}$ = probability the attacker will ever catch up from $\mu(z)$ blocks, i.e. $z$ layers, behind

$$q_{\mu(z)} = \begin{cases} 1 & p \leq q \\ (q/p)^{\mu(z)} & p > q \end{cases}$$

Our analysis will consider the case in which the adversary contains less than 51 percent of the network hash power. Assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = \frac{q}{p}\mu(z)$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress she could have made by the probability she could catch up from that point:

$$\sum_{b=0}^{\mu(z)} \frac{\lambda^b e^{-\lambda}}{b!} \cdot \left(\frac{q}{p}\right)^{\mu(z)-b}$$

Or, 1 minus the probability that this doesn't happen:

$$1 - \sum_{b=0}^{\mu_z} \frac{\lambda^b e^{-\lambda}}{b!} \left[1 - \left(\frac{q}{p}\right)^{\mu_z - b}\right]$$

For the adversary, generating the full braid in parallel is clearly very difficult. However, in reality the adversary will choose to mine only a subset of the braid, as detailed in the further proofs.

# 5 Scenario 2: Merkle cone replacement for arbitrarily large values of $z$

For Scenario 2 we examine the probability of a successful double-spend attack for an adversary attempting to generate a particular referential sub-section of the braid called a Merkle cone for arbitrarily large values of $z$. For this analysis, therefore, we must define the term Merkle cone.

## 5.1 Merkle Cones and referential graph expansion

Given the referential property of Chainweb, for an adversary to successfully replace a block in the braid she must not only generate subsequent blocks on her own chain but also generate blocks on sibling chains that reference the bad block. If the fraudulent block is generated at block height $l = 0$, then at block height $l = 1$ there are $d+1$ blocks containing a hash of the Merkle root of the fraudulent block which form a referential layer.

It follows that the expansion ratio of the space that references the Merkle root of the fraudulent block can be represented by the set of vertices of a bounded d-regular expander graph with each moment of expansion expressed in terms of block height.[3] The vertex space of our expansion graph here is finite and bounded by the size of the braid $\Omega$.

This expanding set of referential blocks begins at the fraudulent block and grows over time until it reaches the entire network after $\Delta$ number of layers. The space created by the expanding layers that reference the Merkle root of the fraudulent block defines a cone-shaped subsection of the full network braid, which we define as a Merkle cone [Figure 4].

The number of blocks in any given Merkle cone layer $l$ is represented by $\sigma(l)$. The number of referential blocks from inception to the current layer $l$ is the Merkle mass, or $\mu_2$. Therefore, a fully-propagated Merkle cone has a mass of $\mu_2(\Delta)$ blocks. For an adversary's alternate chainweb to be accepted, she must offer at least a fully-propagated Merkle cone as an alternative to the network, but for this section 2 analysis she must continue to generate layers for an arbitrarily large value of $z$.

[3] Hoory, Shlomo; Linial, Nathan; Widgerson, Avi (2006), "Expander graphs and their applications" (PDF), Bulletin (New Series) of the American Mathematical Society, 43 (4): 439?561, doi:10.1090/S0273-0979-06-01126-8

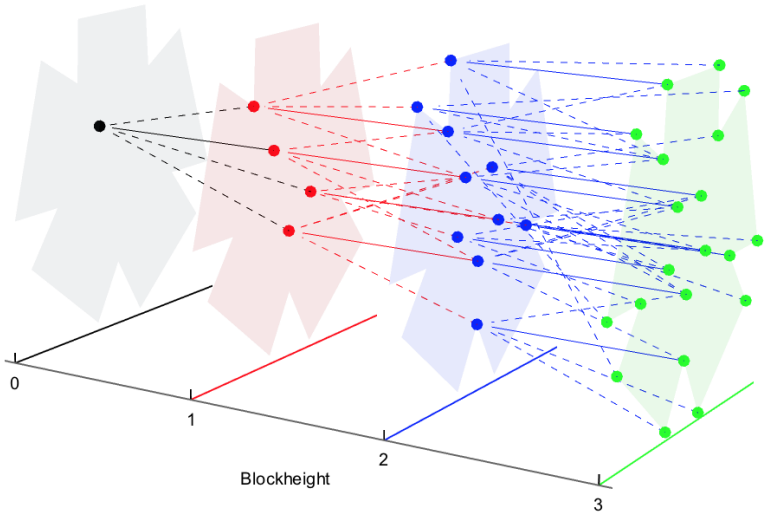


Figure 2: Merkle Cone

### 5.2 Adversary versus network resource allocation

For $l \geq \Delta$ the adversary must mine a layer of size $\Omega$, since every chain contains a reference to the fraudulent Merkle root. For $l < \Delta$, the adversary mines a Merkle cone that is a subspace of the entire braid, and the proportion of the network mining the same space as the adversary is equivalent to the proportion of the network encompassed by the Merkle cone. Therefore, the probability for the adversary and the network of mining a given block in the Merkle cone layer $l$ is as follows:

$r_p$ = resources available to the honest network

$r_q$ = resources available to the adversary

$r_p + r_q = 1$

$$\begin{bmatrix} adversary: & q = r_q \\ network: & p = r_p \frac{\sigma(l)}{\Omega} \end{bmatrix}$$

We let this ratio of the likelihood of success for the adversary versus the network be bounded by 1 and defined as the hash ratio:

$$\rho(l) = \frac{r_q}{r_p \frac{\sigma(l)}{\Omega}}$$

It follows that for arbitrarily large values of $z$ any advantage to the adversary in mining only her Merkle cone as opposed to the full braid swiftly approaches zero, leading her probability to converge upon the probability of mining the entire braid, or the result from Section 1. This feature of Chainweb is the heart of its long-term security as an inviolable record of transactions.

As an aside, we note that in practical application an adversary must wait for the header blocks on peer chains to become available before proceeding on to further layers. Therefore we have given the adversary a theoretical advantage in this proof by assuming that headers are always available to the adversary when they become necessary for future progress.

## 6 Section 3: Merkle cone replacement for $z = \Delta$

Clearly there are limitations from a practical perspective in waiting an arbitrarily large block height before accepting a transaction as having a high likelihood of confirmation. This section presents the case in which the recipient waits $\Delta$ layers such that the adversary must generate only a Merkle cone of $mu_2(\Delta)$ to offer a viable replacement.

To begin, we then substitute the hash ratio into the expected value function:

$$\lambda = \rho(l)\mu_2(z)$$

Given that 1) the subset of blocks the adversary must mine and 2) the proportion of resources the network allocates to the Merkle cone both change with layer in question, the probability distribution modeling the progress of an adversary is a composite discontinuous function of $z$ individual poisson point processes.

For an adversary to catch up to the network if she is mining the first layer, she must successfully finish her current layer as well as all subsequent layers until she completes layer $z$. Therefore, the probability of an adversary successfully catching up is the sum over all blocks in the cone of the product of three terms:

1. The familiar formula of being at a particular block $b$ and the likelihood of completing the rest of the blocks, restricted to the current layer $l$

$$\frac{\lambda(l)^b e^{-\lambda(l)}}{b!} \cdot \rho^{\sigma(l)-b}$$

2. The probability she has completed the blocks of all previous layers before $l$

$$\prod_{k=0}^{l-1} \rho(k)^{\sigma(k)}$$

3. The poisson point process of finishing future layers after $l$ up to and including $z$

$$\prod_{j=l+1}^{z} \frac{\lambda(j)^{\sigma(j)} e^{-\lambda(j)}}{\sigma(j)!}$$

Aggregating over all layers $l$ in the Merkle cone gives us the following final composite function modeling the adversary's likelihood of catching up from any given block $b$ in the cone, times the probability of being in that block:

$$\sum_{l=0}^{z} \left[ \sum_{b=0}^{\sigma(l)} \left( \frac{\lambda(l)^b e^{-\lambda(l)}}{b!} \rho^{\sigma(l)-b} \cdot \prod_{k=0}^{l-1} \rho(k)^{\sigma(k)} \cdot \prod_{j=l+1}^{z} \frac{\lambda(j)^{\sigma(j)} e^{-\lambda(j)}}{\sigma(j)!} \right) \right]$$