

# Proposal for a Merit Caching Algorithm

Luke Parker

August 11, 2018

## Abstract

Proof of Work requires both the current state and extended periods of time to function, which is unacceptable for instant transactions on a cryptocurrency network. If a miner only has to sign a transaction, effectively instant transactions can be achieved. The issue with this is a lack of reputable verification, as this isn't anymore secure than just removing mining altogether, opening the network up to double spends. This also doesn't declare what address new coins should go to.

By using Proof of Work to generate Merit, we can cache work, removing the need for the current state. This is accomplished via a separate blockchain, which would be mined to generate Merit. This non-transferable Merit would approve transactions on a form of Directed Acyclic Graph. As signing can be done almost instantly, this system allows for the security of Proof of Work to be applied to an effectively instant transaction system.

**License** Merit Caching was designed by Luke Parker for use by Ember. Merit Caching's design and whitepaper, but not any implementations or pieces of code included/shipped with this whitepaper, are for the public domain. Any included implementations and code, whether it be pseudocode, compilable code, or otherwise, are licensed under the MIT License, copyright Luke Parker as of 2018.

# Contents

<b>1</b>	<b>Employed Data Structures</b>	<b>3</b>
1.1	Blockchain . . . . .	3
1.1.1	Blocks . . . . .	3
1.2	Lattice . . . . .	4
1.2.1	Transactions . . . . .	4
1.2.2	Verifications . . . . .	4
<b>2</b>	<b>Mining and Verifying Transactions</b>	<b>5</b>
2.1	Mining . . . . .	5
2.2	Verifying Transactions . . . . .	5
2.3	Incentive . . . . .	6
<b>3</b>	<b>Attack Vectors</b>	<b>7</b>
3.1	Double Spends . . . . .	7
3.2	51% Attack . . . . .	7
3.3	51% Attack Part Two . . . . .	7

# 1 Employed Data Structures

Merit Caching is dependent on a dual storage structure, with one being a blockchain, and the other being an asynchronous structure, such as a Directed Acyclic Graph, or Lattice (as used in this paper). This is due to the Lattice being needed for instant transactions, yet a time-proven, secure, and reliable blockchain is the mineable structure. If we mined the Lattice on a transaction by transaction level, we'd have to wait to generate the work, as work's security is provided by the time needed to calculate it, and therefore transactions wouldn't be instant.

## 1.1 Blockchain

A blockchain is chunks of data (blocks) in a linear fashion, or a linked list. No existing blocks can be edited, and all new blocks always appear at the end, making the 'x-axis' of the database time. This means someone who downloads the blockchain years after its creation can see how it was formed over time and verify its state.

Merit Caching considers the blockchain with the most accumulated work valid, just as most Proof of Work systems do. This is because the greater work symbolizes greater processing power and the chain is therefore the most secure. New blocks are added only if the miners can generate a hash which beats the difficulty, a property that exists to require high amounts of work, and is generated based on how often blocks are added. This also allows semi-consistent spacing of blocks.

### 1.1.1 Blocks

In blockchain based cryptocurrencies, blocks contain transactions. In the Merit Caching algorithm, blocks contain a list of verifications the Lattice has of its transactions. By taking these verifications from

the Lattice to the Blockchain <sup>1</sup>, we can provide a master overview of the Lattice to be used for syncing and deciding block rewards.

## **1.2 Lattice**

A Lattice is a modified form of a Directed Acyclic Graph. Each address has their own blockchain, and only they can write to it. This allows definitive block ordering, even at high speeds with severe network lag, asynchronous writing to the database (if the accounts are different), and easy comprehension by humans.

### **1.2.1 Transactions**

Transactions are composed of two parts: send and receive blocks. When a user sends cryptocurrency to another user, they create a send block. For this data to be added to the other user's blockchain safely, as rapid incoming transactions could overwrite the indexes the user is trying to send cryptocurrency with and therefore stop them from spending money, the other user must create a receive block that declares the index.

### **1.2.2 Verifications**

Verifications are created by Merit Holders, and include a transaction hash. What they do is covered below.

---

<sup>1</sup>Blockchain is capitalized as it is used as a proper noun.

## 2 Mining and Verifying Transactions

### 2.1 Mining

Merit Caching involves mining the Blockchain via a Proof of Work algorithm, and any algorithm, from SHA256 to Scrypt to Lyra2 to Argon2, satisfies Merit Caching's needs. Successfully mining a block generates Merit, which is a non-transferable value that decays over time. This is because Merit mined years ago should not still be generating profit so small it's 10 decimals down and clogging the Lattice/Blockchain with its verifications.

### 2.2 Verifying Transactions

When a new send block is added to the Lattice, everyone with Merit can sign it or not sign it. If they do, they are putting their Merit behind it. A transaction only becomes valid when it has 50+% of all Merit behind it. This means if a double spend attempt is broadcasted, and both are at 49.9% because they were sent out at the same time to different parts of the network, the person with the last 0.2% will only sign one, stopping the double spend before it's confirmed. If a Merit Holder is malicious and signs a double spend, they will lose all their Merit for this action. The Merit behind the transactions will be recalculated, and one may need to cross the 50% threshold again.

## 2.3 Incentive

Merit Holders will earn coins from the Blockchain after a Mining Period. A Mining Period is a set amount of blocks from a transaction's first appearance on the Blockchain to the block where if it still isn't verified, it is orphaned as it will never be verified. When a Mining Period is finalized, Merit Holders will get coins based on:

- Percent of the Merit placed behind the transaction that was theirs;
- Percent of transactions in this period they helped verify.

## **3 Attack Vectors**

The network starts on the Blockchain and feeds back into it. As everything goes through it, Merit Caching successfully avoids most vulnerabilities usually found on a Proof of Stake system, yet keeps the ones found with Proof of Work blockchain systems.

### **3.1 Double Spends**

As covered above, double spends can be submitted, yet only one will be verified without a malicious Merit Holder. When that happens, the malicious Merit Holder will lose their Merit and the transactions will revert to both being pending or only one being verified. This would be resolved in a matter of seconds, and there's no advantage to the Merit Holder. If it's a high value transaction, and a service wants to be absolutely sure, they can wait for the next block to mined. That said, they only really have to wait a few seconds for the double spend to appear on the same node and be flagged/resolved.

### **3.2 51% Attack**

As the chain follows the longest chain rule, the chain can be rewinded if an attacker has 51% of the mining power. Since verifications are based on weight created by the Blockchain, editing the Blockchain will alter transactions on a mass scale, potentially not just reverting transactions to pending but invalidating them entirely.

### **3.3 51% Attack Part Two**

As miners control what verifications are included in a block, any miner who gets 51% of the mining power can selectively verify transactions.