Bobtail Prototype Extending Storm on Bitcoin Unlimited

Submitted by: George Bissias
Date: 7 Nov 2019

Motivation

Two of the most fundamental attacks on proof-of-work (PoW) systems like Bitcoin are the doublespend and selfish mining attacks. In the original Bitcoin whitepaper, Satoshi Nakamoto showed that for an attacker with 10% of the total hash rate, we must wait at least 5 confirmations to be sure that the attacker's chances of doublespending a transaction are below 0.1%. This analysis gave rise to the 6 block confirmation convention, which leads to a practical limitation for merchants both in person and online. And despite mitigations, doublespends do occur in the real world, albeit in cryptocurrencies with relatively low total hash rate.

Selfish mining attacks involve an attacker who withholds his blocks in hopes of causing honest miners to produce wasted work. In their paper, Eyal and Sirer showed that attackers with just over 20% of the total hash rate can actually increase their effective hash rate by engaging in selfish mining. The net effect to the system is a tendency toward mining pool centralization and potentially a decrease in honest miner participation due to lost profits.

Our recent work introduced Bobtail, a modification to the Bitcoin consensus protocol that dramatically lowers the risk of both doublespend and selfish mining attacks. As a side benefit, Bobtail also ensures that blocks are generated at more regular intervals when hash rate remains consistent.

Background

Bobtail

A block can be mined in Bitcoin whenever a miner generates a hash of the block header that falls below the target value. We refer to such a hash as partial-proof-of-work, or proof for short. The lowest proof constitutes Bitcoin's mining statistic. Bobtail is an update to the Bitcoin consensus protocol whose mining statistic is the average of the lowest k hash values, or proofs. This average is required to fall below the target in order to mine a block. Miners work collaboratively to assemble the proofs required to meet the target. Proofs constitute samples from the uniform distribution. The Bitcoin protocol uses the lowest order statistic (OS) from this distribution while Bobtail uses the average of the lowest k. Because the lowest k OSes constitute a larger sample than the lowest OS alone, it turns out that the Bobtail mining statistic has lower variance than Bitcoin's.

Due to the lower variance Bobtail mining statistic, we have shown that the probability that an attacker with as much as 35% of the total hash rate will succeed in a doublespend attack with a single confirmation is less than 1%. This compares to a 44% chance of success for a similar attack in Bitcoin. We have also shown that the risk of selfish mining in Bobtail is virtually eliminated: attackers with less than 45% of the total hash rate actually stand to lose money. Finally, when hash rate remains stable, 95% of blocks will arrive within 12 minutes in Bobtail compared to 40 minutes for Bitcoin.

Storm

Storm is a solution for zero-confirmation transactions or so-called preconsensus for the Bitcoin Cash protocol. It works by having miners report partial PoW through delta blocks.

A delta block contains a block header, a set of delta-block ancestors and a set of transactions named the delta set. The PoW in the header can be weak or strong. The PoW must minimally be below the weak threshold. If it is also below the strong threshold (i.e. the target) then it constitutes a valid (strong) block. Storm's delta blocks are formed by referencing any number of compatible ancestor delta blocks. The aggregate of a delta block and all its ancestors is called a merged block. The transaction set implied by the merged block is the union of all delta sets in the constituent delta blocks.

Weak PoW (WPoW) for a merged block is the quantity of delta blocks comprising the merged block. Multiple delta blocks can achieve the highest WPoW at any given time. These are called the weak tips. When there exist multiple strong blocks competing in a block race, WPoW is used to decide the winner. The idea is that a miner is thus incentivized to build on the weak tips in order to improve his chances of winning a block race. Assuming that miners do in fact build upon the weak tips, and that they broadly disseminate the associated delta blocks, then merchants can gain some confidence that a transaction appearing in the delta set of the weak tips will appear in the next block.

Cooperation between Bobtail and Storm
Bobtail provides strong mitigation for doublespend and selfish mining attacks that span multiple blocks. It is therefore highly effective against attacks on expensive purchases that are protected by multiple confirmations. However, Bobtail's protection against zero-confirmation doublespends is no better than Bitcoin today. For this reason, a combination with the Storm protocol would greatly improve Bobtail's overall doublespend protection. Moreover, Bobtail can potentially leverage Storm's existing prototype implementation for Bitcoin Unlimited.

On the other hand, Storm does not provide doublespend protection beyond zero-confirmation transactions. So it too can benefit from a combination with Bobtail. Additionally, the Storm protocol does not provide direct incentives for delta block creators. We feel that this is critical for keeping the protocol incentive compatible. For example, one of the problems with Storm is that it is not clear that there exists strong enough incentive for miners to release or propagate delta blocks because they can enjoy an advantage in the event of a block race if they do not. Indeed Storm's author acknowledges that it might be necessary to reward delta blocks to solve this problem. A benefit of incorporating Bobtail is that it already contemplates how to reward partial PoW such as that comprised by delta blocks.

Objectives
This project will deliver a prototype of the Bobtail protocol as a branch of the Bitcoin Unlimited codebase. Our intent is to leverage the existing Storm prototype to speed implementation as well as maintain the benefits of Storm preconcensus. It will also benchmark the increase in network overhead required by a Bobtail deployment. A detailed roadmap breaks down into the following five parts:

1. Create hybrid protocol
    - How do Bobtail proofs coincide with Storm delta blocks?
    - What is the best way to incorporate Bobtail proof rewards into the Storm protocol?
    - What are potential interactions (either positive or negative) between the zero- and multi-confirmation doublespend protections offered by the two protocols?
    - If for some reason it is determined that Bobtail is not compatible with Storm, then we will proceed with the implementation of the Bobtail prototype without maintaining Storm compatibility.
2. Implement Bobtail-specific data structures and logic (Likely some overlap with Storm code):
    - Proof set: partial PoW for miners, which includes Merkle root, payout address, and a pointer to the supporting proof
    - Transaction set: transactions necessary to validate proof sets
    - Proof queue: the Bobtail paper offers a threshold for what proofs are worth keeping / disseminating; maintain priority queue according to this threshold.
3. Update consensus logic
    - Mining code must generate and hash Bobtail proof sets and maintain a proof queue
    - Validate blocks according to the Bobtail protocol
    - Provide facility for validating proof and transaction sets
4. Update the network protocol to support (likely some overlap with Storm code)
    - BOB_PROOF message for proof set dissemination
    - BOB_TRANS message for transaction set dissemination
    - Modify block propagation to support extra elements from Bobtail
5. Testing
    - Develop python rpc-framework test to validate Bobtail prototype
    - Benchmark increase in network overhead due to Bobtail

- Benchmark increase in network overhead due to Dobtail

Project Duration

The total anticipated project duration is 6 months (based on part-time work). This will break down by part as follows.

Part 1: 1 month
Part 2: 1.5 months
Part 3: 1.5 months
Part 4: 1 month
Part 5: 1 month

*In order to leave time to complete work on Graphene v2.2, the project will begin on or after December 29, 2019.

Budget

The total requested budget for this project is $34,100, or $5,683 per month of the anticipated project duration. The budget will break down by part as follows.

Part 1: $5,684
Part 2: $8,525
Part 3: $8,525
Part 4: $5,683
Part 5: $5,683