ECIP Comparison for
51% Attack Resistance

Author

Kostis Karantias (IOHK)

Contributions from:

Brian McKenna (IOHK)

Dimitris Karakostas (IOHK)

Stevan Lohja (Director of Developer Relations, ETC Cooperative)

Table of contents:

Introduction

The recent 51% attacks have put Ethereum Classic in a precarious position.

These attacks have dented confidence in the ecosystem and challenged the community's ability to collectively address a very real issue while representing an existential threat to its future viability. Some exchanges require 91,000 block confirmations to accept a transaction. If the 51% attacks continue, the possibility of rapid and widespread de-listing on exchanges could become worryingly real.

IOHK has a long association with ETC and its community. To mitigate against these attacks and secure the ETC Network, the team at IOHK has carried out an analysis of the different options and proposals from across the ETC community.

As well as providing our own recommendations, we wanted to ensure the ETC community had visibility and understanding of the options available. Utilizing a curated series of Crowdcast presentations on each solution, each developer team was given a showcase to present their proposals. The goal of this comparative analysis - curated by IOHK but created collaboratively - is to further arm the ETC community with the knowledge and understanding to determine the right steps to take. This will help us all arrive at the most appropriate solution for network security, in the near-, mid- and longer-term.

IOHK believes that challenging times call for collaboration, not 'competition'. So this paper also serves to educate the wider cryptocurrency community on how to mitigate against future attacks on other proof of work chains.

It should be noted that all of these solutions come with trade-offs and they should only be considered as temporary. Ultimately, the root cause of the attacks will be mitigated by innovation, growth and belief in the ecosystem. ETC needs to have a clear vision on how to grow and attract the best and the brightest to create an innovative and deliverable roadmap. This roadmap should be aligned with the ETC ethos, whilst ensuring that ETC can compete with other platforms. With this, ETC will be able to mature and return to being a market-leading smart contract platform and beacon for innovation serving as one of the most secure and functional proof of work blockchains.

Goals

Before deep-diving into the ECIP comparison, let us first understand the goals of Ethereum Classic. Ethereum Classic aims to implement a robust transaction ledger. This theoretical analysis research paper strictly defines the properties that a robust transaction ledger needs to possess: **persistence** and **liveness**. These properties are crucial for any type of a transaction ledger regardless of implementation (i.e. whether it is proof of work, proof of stake, etc.) The robust transaction ledger is parameterized by *k* (the depth parameter) and *u* (the waiting time). A high-level definition of these two properties is as follows:

1. **Persistence**: If an honest party reports a transaction more than *k* blocks away from the end of the chain (i.e. the transaction is stable), then from that point, every honest party should report that transaction in the same position of the chain.
2. **Liveness**: If a valid transaction is submitted to the network for at least time *u*, it should subsequently be reported as stable by all the honest parties.

It is a proven fact that both persistence and liveness suffer when the adversarial mining power in the proof of work surpasses 50%. In the recent year, Ethereum Classic double-spending attacks have been conducted by creating large reorganizations of the chain. These kinds of attacks are called **persistence violations**. On the other hand, **liveness violations** can occur when the adversary is able to mine practically all blocks on the chain, which can result in withheld transactions or empty blocks, for instance. Adversaries can also perform attacks such as selfish mining, which provides them with disproportionately many mining rewards. This leaves honest miners with less block rewards further causing damage to the network as rational honest miners would eventually stop being network participants.

Considering that persistence and liveness are currently not guaranteed within the Ethereum Classic network, we are looking to implement protocol changes that will re-establish persistence and liveness under current network conditions.

Proposals

Checkpointing

The first proposal is to implement a checkpointing mechanism, based on the work of Karakostas et al.

Checkpointing suggests parameterizing the system with k_c parameter. Every k_c block gets irreversibly "checkpointed", which means that no one can ever drop or revert it. A trusted federation of nodes can *choose* the block to issue a checkpoint, which means they can decide which block becomes the canonical chain that all parties should follow. This trusted authority must *run continuously* and is responsible for *publishing* the checkpoint to the network. An example of a checkpoint is a simple signature on a standard block or a specially-crafted block.

Checkpointing ensures that the protocol is unaltered with regards to mining. The mining rewards are not affected. The checkpointing federation can only issue checkpoints on blocks that have valid proof of work and cannot mint blocks on its own. Because of the BFT protocol that the federation needs to run in order to decide which block to checkpoint, we can tolerate < 1/3 of the federation being malicious.

Advantages

1. Deterministic finality allows exchanges to lower their confirmation times significantly with confidence, improving one of the major issues Ethereum Classic faces today
2. The underlying proof of work protocol is not replaced, rather it is augmented. Miner rewards are not altered in any way
3. The proposal is formally described and proven
4. This is one of the two solutions that achieve liveness

Concerns

1. The protocol depends on a centralized federation, and can tolerate at most ⅓ of the participants being malicious.
2. As implemented in Mantis the protocol requires every federation party to be honest.
3. If at least one third of the federation members become malicious, some attacks can be performed *regardless* of the honest mining majority
4. Depending on the authority's performance, it may take several seconds before a checkpoint can be issued. With the growth of the federation, its performance worsens

Effort: IOHK has a working prototype implementation on Mantis. This implementation requires every federation party to be honest.

Timestamping

In the same research paper by Karakostas et al., it is proposed that the same security guarantees can alternatively be provided by using a timestamping service. Naturally, since any secure decentralized ledger offers reliable timestamping, this allows us to base the security of Ethereum Classic on the security of some other ledger. By doing this, we avoid relying on a federation and thus, this solution is completely decentralized. For the purpose of illustration, we present a solution based on Bitcoin, while stressing that the scheme can be implemented using any secure decentralized ledger.

Ethereum Classic miners need to run a full Bitcoin node to retrieve and verify timestamps as well as own some bitcoins to create timestamping transactions; in September 2019, this cost was $3.6 per block.

We assume the following parameters:

- k_c: the checkpointing interval. Note that k_c needs to be large enough, such that a timestamp has enough time to become stable on Bitcoin; in practice, it should correspond to *at least* 60 minutes.

- *g*: the ETC block number that marks the beginning of the timestamped period.

The process below explains how to mine in ETC using a timestamping feature. To produce a new block with number *j* (if $j = g + i*k_c$) for some integer *i* (i.e. every k_c blocks), the miner should:

1. Retrieve the timestamp of the ETC block $g + (i - 1)*k_c$ from Bitcoin. This will be the hash of the oldest BTC block containing a part of the previous ETC checkpoint's header.
2. Insert a special transaction (or the header field) in the new ETC block. The transaction should contain the timestamp of the previous ETC checkpoint (as described in step 1).
3. After creating the new ETC block, publish its headers on Bitcoin. For this, a miner should use a series of OP_RETURN transactions.

If $j \neq g + i*k_c$, the miner should create a block as usual.

To choose between two ETC chains (C^1 and C^2), a miner should:

1. Find the common ancestor block B_C of the two chains.
2. Find the first block B_t^1, B_t^2 after B_C in each chain. These are supposed to be timestamped on Bitcoin (i.e. the first block after B_C with $g + i*k_c$ number for some integer *i*).
3. If one of the two blocks is not timestamped, then pick the other chain; if timestamp (B_t^1) < timestamp (B_t^2) (i.e. if B_t^1 has an older timestamp than B_t^1), then the miner should pick C^1; otherwise - pick C^2.
4. If B_C is the latest timestamped block on both chains, the maxvalid rule applies.

In practice, a miner can choose on which chain to mine before a timestamp becomes stable. Specifically, they may assume that the Bitcoin miners follow a first-come-first-in-the-block rule, such that the first timestamping transaction they observe on the Bitcoin network will eventually be the chosen one.

This proposal requires a soft fork to add the special field/transaction (step 2) that will be used to ensure liveness.

We remark that as proposed, helpful users are not directly incentivized to submit transactions to the Bitcoin network and may seem altruistic. However, if a miner wishes to ensure that their block will not be reorganized and that they will indeed obtain their rewards, they can timestamp their block with a minor discount on the block reward. Changes to the scheme to compensate users or miners who timestamp transactions corresponding to the RSK proposal can be incorporated as well.

Advantages

1. The security of the scheme relies solely on Bitcoin, therefore, it does not assume any (semi)centralized authority for protection
2. Retrieving the Bitcoin timestamp and inserting it into the ETC chain (see step 2 below) ensures liveness with some probability (as opposed to simply publishing a hash of the chain on Bitcoin, which ensures *only* persistence; this means that with the timestamping mechanism, it is possible to ensure both persistence and liveness)
3. The underlying proof of work protocol is not replaced. Miner rewards are not altered in any way
4. The proposal is formally described and proven

Concerns

1. The scheme is not practical because each checkpointed ETC block needs to be posted in full in the Bitcoin chain
2. The fee market on Bitcoin may cause blocks not to get timestamped in a timely manner
3. Full nodes need to be connected to the Bitcoin network

Effort: There is no planned implementation for timestamping.

RSK

Rootstock (RSK) is a smart contract platform, which is connected to the Bitcoin blockchain utilizing sidechain technology. RSK does not have native tokens, instead, it is fueled by SmartBitcoins (RBTC) that are 2-way pegged with bitcoin (1 RBTC = 1 BTC). The 2-way peg

system is federated. RBTC is used for rewards and mining fees. Currently, approximately 40% of the Bitcoin mining power merged mines RSK blocks.

Merged mining allows Bitcoin miners to mine the RSK sidechain. At the end of the Bitcoin block coinbase transaction, there used to be a tag “RSKBLOCK:” followed by a hash. This hash was the hash of the RSK block that had been merged mined. However, this has changed so that the hash following the tag commits to an authenticated dictionary. Then, the RSK block that is mined can be validated to be included into this dictionary with a prespecified key. An authenticated dictionary with a prespecified key is necessary to eliminate the creation of multiple RSK blocks pegged by attackers to one Bitcoin block.

The proposal is to implement checkpoints (ETC block hash/height) within Bitcoin blocks and verify them with Bitcoin (SHA-256) proof of work. Each ETC block can be then assigned with a score, which is calculated as the sum of all the nominal difficulties of Bitcoin blocks covering the ETC block. A **hit** is defined as a Bitcoin block that commits to the ETC block. A hit can have multiple **hit confirmations**, which are consecutive blocks that do not commit to any ETC block. Hit confirmations need to form a chain from the hit that can be traversed with previds. The score of an ETC block is the sum of the difficulties of both its hits and hit confirmations.

In order to include ETC block information in Bitcoin blocks, RSK proposes to rely on the RSK-based smart contract called Universal Merged Mining (UMM). Merged miners query this contract to discover additional information to include in merged mined Bitcoin blocks, inside the authenticated dictionary. Merged miners should find ETC block information in this contract and add it to their merged mined Bitcoin blocks to generate hits. Every day, a blind auction takes place in the RSK UMM contract from interested participants who wish to have their own string of data included in merged mined blocks. 24 1-hour slots are auctioned.

The information about hits and hit confirmations of each block is encoded in a VisibilityProof that is added to future ETC blocks. This is how participants in the ETC network can view the score of each block without extra external communications. Miners who add a VisibilityProof in their blocks are rewarded. The reward comes from the dilution of all other rewards so that no changes in coin issuance need to be made.

The mechanism is set in place so that alarms can be triggered on specific conditions. The description hints towards heuristics, as in “the score of the last 100 blocks is less than 80% of the score of the previous 100 blocks”. In this scenario, the node can enter into a failsafe mode and stop confirming transactions. Double-spending attacks are mitigated because a victim node enters failsafe mode before the first spending of the attack is confirmed. This prevents individuals and automated systems from acting upon the confirmation until the attack attempt is over, which is also locally detected by the node.

Advantages

1. The security of the scheme relies solely on Bitcoin, therefore, it does not assume any (semi)centralized authority for protection

Concerns

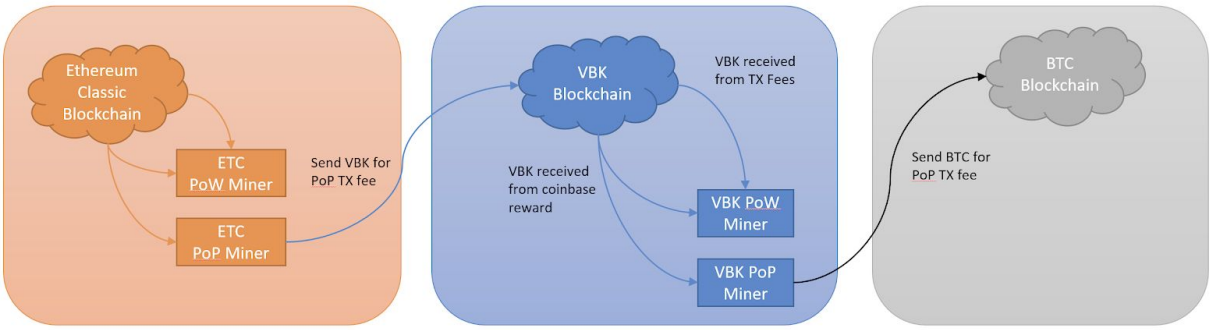
1. Miners need to be connected to the RSK network
2. Miners need to own RBTC and participate in auctions for UMM time every day. (It is not clear what happens if participants are always outbid, as the auction is blind.)
3. No formal claims are made nor is a proof provided
4. Liveness is not addressed

Veriblock

Veriblock is a blockchain that implements Bitcoin’s proof of work (PoW) security by a theoretically unbounded quantity of additional blockchains. It offers protection to other ledgers against double-spending attacks. Veriblock is an account-based PoW blockchain that relies on Bitcoin’s security with what they call proof of proof (PoP). PoP works by relaying Veriblock blocks to Bitcoin. Below is the lifecycle of a Veriblock transaction.

VeriBlock Transaction Lifecycle	
Stage	**Description**
UNCONFIRMED	The transaction is in the VeriBlock blockchain mempool
N-CONFIRMED	The transaction is confirmed by N VeriBlock blocks
N-BTC-REFERENCES	The transaction is in a VeriBlock block which was referenced N Bitcoin blocks ago, and early attack detection metrics can begin to function
N-BTC-FINALITY	The transaction has achieved Bitcoin finality; and N BTC blocks would need to be forked to challenge it

The ETC block header can be submitted in a special transaction to the Veriblock blockchain. This requires transaction fees.



The ETC block header will change to include:

1. PoP transactions in Veriblock from ETC
2. Veriblock headers for potential SPV verification of the Veriblock chain
3. PoP transactions in Bitcoin from Veriblock

The Veriblock chain verification is not specified and the chain weighting approach is not described in the ECIP. However, more details on the approach can be found in the Veriblock whitepaper, which is non-standard and requires further analysis in order to determine its security. The use of magic numbers is not encouraging.

The ECIP suggests a “transaction offloading” mechanism to mitigate the loss of liveness caused by a majority attacker. However, the mechanism is described neither in the ECIP nor in the whitepaper.

Concerns

1. Every node needs to be connected to 2 extra networks (Veriblock & Bitcoin)
2. This solution proposes a drastic change to the header and adds significant processing overhead
3. PoP miners need to pay fees that are reimbursed when PoP blocks are mined on ETC. Reimbursements are of arbitrary amounts and thus coin issuance needs to change potentially
4. The fee market may make the solution unreliable
5. Liveness is not addressed

PirlGuard

Pirl is a semi-centralized, Ethereum-based system. It consists of "masternodes", which run PoW and monitor the system for attacks, being rewarded with Pirl tokens. In case of attacks (see here), masternodes and developers can enforce penalties to ban dishonest parties (for example, attackers) from participating in the network consensus. Changes performed by the attacker will be reverted.

Pirl suggests a heuristic to make 51% attacks harder (see the "peer penalties" here). It forces an attacker to spend more hashing power if they want to revert some blocks in the honest chain of a party. Specifically, if a reorganization of less than k_p blocks is suggested, then it is allowed. However, a reorganization of $n > k_p$ blocks can only happen if the contesting chain also contains Θ(n^2) “penalty” blocks on top of the blocks normally needed to be a candidate for reorganization.

Advantages

1. Ease of implementation

Concerns

1. No explicit description of the protocol other than code snippets or formal proof of security

2. Use of penalty blocks in a variable difficulty environment instead of penalty difficulty opens the protocol up to potential attacks
3. This protocol is subjective, and nodes in the network may have different views of the state. This is a departure from classical blockchain protocols and has not been studied in the literature

MESS

*Modified Exponential Subjective Scoring (*MESS) is a modified version of Vitalik Buterin's *Exponential Subjective Scoring* (ESS). It is proposed by ETC Labs in consultation with Chainsafe and OpenRelay. MESS aims to make larger chain reorganizations more difficult.

MESS requires the full node logic to change as follows. When a reorganization is proposed, the lowest common ancestor (LCA) between the proposed chain and the current chain is located. The time difference between the two blocks is calculated as the difference between the respective block timestamps. We will refer to the part of any of the two chains starting from the LCA as the subchain. The goal is to make reorganizations harder as this time difference grows. The existing protocol for Ethereum Classic compares the total difficulty of both subchains and declares the winner subchain to be the one with the most total difficulty.

MESS changes this by giving an advantage to the local chain depending on how old the LCA block is. Thus, the aforementioned time difference is taken into account. Assume a function *f(dt)* which provides the relative advantage of the local subchain. The local subchain's difficulty is multiplied by this relative advantage and then the two subchain difficulties are compared.

More formally, instead of comparing *local_subchain_td* with *proposed_subchain_td* we now compare *f(dt) * local_subchain_td with proposed_subchain_td.*[1]

The proposed relative advantage function is a smooth ramp function which starts at 1 and tops off at 31 when its input reaches 25132 seconds (or ~7 hours). From that point, the relative advantage remains constant.

MESS does not come with a formal security proof, thus making it difficult to know what security guarantees it provides. Unfortunately, attacks to the protocol have already been discovered. Due to the use of block timestamps in performing chain selection, combined with the fact that no validation takes place for block timestamps, and that they can be

[1] This formulation is different but equivalent to the one presented in the ECIP. It is preferred here for readability.

effectively adversarially chosen, MESS is vulnerable to selfish mining attacks. The attacks have been described in detail by Sergio Lerner.

Advantages

1. Ease of implementation

Concerns

1. The use of block timestamps weakens the security of the protocol, as they can be effectively adversarially controlled
2. Authors do not formally define security for the protocol, nor provide a formal proof
3. Attacks against this protocol have already been described
4. No reasoning is provided behind the changes applied to the original ESS protocol (which is also not proven to be secure formally). One of the changes, namely - the use of block timestamps - opens the protocol to attacks
5. Liveness is not addressed
6. This protocol is subjective, and nodes in the network may have different views of the state. This is a departure from classical blockchain protocols and has not been studied in the literature
7. MESS provides *subjective* finality while other proposed solutions to ETC's 51% attacks provide more confidence when a transaction is absolutely final or make 51% attacks out-right impossible. Additionally, there is no indication that exchanges will be able to lower their confirmation times due to the adoption of this protocol

Effort: ETC Labs implemented MESS in Core-geth. ETC Labs announced that MESS is scheduled for mainnet at block 11,380,000 (~October 10). *ETC Lab's Core-geth has ~70% node share**

Comparison

	Fixes persistence	Fixes liveness	Danger	Monetary cost	Hard-fork necessary	Complexity	Remaining effort
Checkpointing	Proven	Proven	Assumes < ⅓ of the federation can be corrupt	Low	Yes	4/5	2/5
Checkpointing (Mantis impl.)	Proven	Proven	Assumes every federation member is honest	Low	Yes	4/5	2/5
Timestamping (Bitcoin)	Proven but impractical	Proven but impracti cal	Not practical to post full blocks on Bitcoin	High	For incentivization	5/5	4/5
RSK	Defends against some attacks	No	Fee market	High	For incentivization	5/5	4/5
Veriblock	Defends against some attacks	No	Fee market	High	Yes	5/5	3/5
Pirlguard	Makes some attacks more costly	No	Chain splits	Low	No	1/5	1/5
MESS	Makes some attacks more costly	No	Chain splits	Low	No	2/5	1/5

Monetary cost: Recurring cost the solution requires to function.

Hard-fork necessary: If the solution requires a modification in the consensus protocol, then it requires a *hard-fork*. However, each solution would have to be coordinated like a hard-fork whether the consensus protocol is modified or not.

Complexity: The technical complexity of each solution. 1/5 is least and 5/5 is most complicated.

Remaining effort: The effort needed for the solution to be functional in the Ethereum Classic network. The effort is proportional to complexity, except for cases where an implementation is already underway where we are only concerned about the complexity of completing the implementation.

Conclusion

Ethereum Classic is not the chain of preference in the Ethash environment and therefore, it is not expected that adversarial majority attacks will vanish in the near future. Therefore, security assumptions of the protocol are no longer applicable. Temporary mitigation from such attacks would provide security and confidence in the marketplace, and a safe environment for Ethereum Classic protocol developers to aggressively innovate, bringing Ethereum Classic back to a secure, decentralized proof of work blockchain. Therefore, while MESS seems to be reaching adoption in the immediate term, we believe that it will not provide robust security and there is no guarantee that it will prevent further attacks. Our analysis concludes that Checkpointing and Timestamping provide far greater, and importantly formally proven, security against all attacks. It is important that any 51% attack mitigation is truly robust enough to give absolute certainty to ETC holders, users, and service providers that their transactions will be secure.

While a sense of urgency is apparent in the community, we cannot stress enough that this is a very important crossroads and that no rushed decisions should be taken. Major stakeholders such as exchanges have shown incredible patience in the past years while Ethereum Classic has been under attack. It is not enough for us as a community to claim we've rectified all of the network's issues. If these attacks appear again in the future because we adopted an understudied solution, it is unlikely the same stakeholders will remain patient with the ETC community and may decide to cut their losses and depart from the network and the community.

For the longer-term health and success of Ethereum Classic, we need to look past these short-term solutions towards network growth, sustainability, and innovation to provide network security. A decentralized treasury would ensure two important things for the

future of the ecosystem. Primarily, it would provide a permanent ongoing source of funding for ETC while making the ecosystem more valuable on the whole. Secondarily, it would provide a democratic and transparent funding mechanism, which lets the ETC community determine its future growth, bringing the innovative vision required to truly compete.