Melmint: trustless stable cryptocurrency

Yuhao Dong
University of Waterloo
yd2dong@uwaterloo.ca

Raouf Boutaba
University of Waterloo
rboutaba@uwaterloo.ca

Abstract

Decentralized cryptocurrencies have gathered increasing interest in the past few years, raising hopes of a new era of non-sovereign electronic money. Unfortunately, cryptocurrencies perform poorly as actual money due to their unacceptably volatile purchasing power. “Stablecoins” aiming to reduce this volatility, on the other hand, tend to peg to an external currency like the US dollar, gravely weakening the decentralization that makes cryptocurrencies so attractive.

Melmint is a mechanism for issuing a trustlessly stable cryptocurrency, Mel, designed for the prototype Themelio blockchain but easily portable to others. Mel is defined without any reference to external pegs such as the US dollar, eliminating the need for oracles and other trusted third parties. This solves a major open problem in the field. We use Elasticoin, an existing proposal to reduce cryptocurrency volatility, as a building block for a trustless monetary policy that gives Mel robustly stable purchasing power. We evaluate Melmint through both theoretical economic arguments and stochastic market simulation, an approach not seen in the existing literature. In all our tests, Melmint is shown to be exceptionally robust in both mundane and extreme economic conditions.

1 Introduction

Blockchain-based decentralized cryptocurrencies, pioneered in 2008 by Bitcoin [18], are becoming increasingly widespread. Apart from many websites and payment processors accepting cryptocurrencies as a trustless and irreversible payment medium, cryptocurrency trading has become a significant financial market with a combined market capitalization of over 100 billion US$ [5]. This growth is despite significant regulatory uncertainty and pressure, as decentralized cryptoassets pose a certain threat to the enforcement of regulations such as capital controls and KYC-style financial reporting obligations. Demand for easy-to-use, electronic money that is entirely independent of centralized, government-backed monetary authorities will likely continue to support the growth of cryptocurrencies.

Despite their rising popularity, however, blockchain cryptocurrencies do not actually see a lot of usage as *money* — an asset that’s simultaneously a store of value, unit of account, and medium of exchange [14]. Cryptocurrency payment processors (such as BitPay [1]) typically convert payments immediately into fiat currencies, few people store personal savings in cryptocurrency, and prices are generally not quoted in cryptocurrency terms. At best, cryptocurrency is used as a “hot-potato” payment intermediary; at worst, it is used entirely as a speculative asset sitting on exchanges.

This state of affairs is mostly due to the *extremely volatile value* of cryptocurrency. Cryptocurrency exchange rates can fluctuate as much as 15% in a single day [5], greatly increasing the risk of long-term holding and hindering usage as money. Volatility is in turn caused by entirely demand-agnostic currency issuance — for example, Ethereum simply mines 2 ETH per block [27] — which causes the fluctuating demand of cryptocurrencies to directly translate into large changes in price.

Much of the existing work on solving this problem focus on *stablecoins*, or cryptocurrencies that are pegged to an external value-stable asset, generally a fiat currency like the US dollar. Stablecoin schemes include centralized currencies like Tether [22] and TrueUSD [24] that act as fiat-denominated IOUs against a trusted bank as well as semi-decentralized systems such as MakerDAO [21] which attempt to hold a peg through algorithmic monetary policy involving complex on-chain financial assets.

A problem common to all stablecoins targeting an exchange rate to an asset external to the blockchain, though, is that there is no trustless way of measuring this value on the blockchain. All stablecoins, even “decentralized” ones like MakerDAO, rely on trusted *price oracles*. In addition, coins tied to external assets are inherently vulnerable to shocks in the price of that external asset.

In this paper, we present Melmint, the first *trustless* value-stable cryptocurrency issuance scheme we are aware of in the literature. We define a hypothetical unit of account known as the DOSC, indexed to the value of one *d*ay *o*f *s*equential *c*omputation on an up-to-date processor. DOSCs have remained surprisingly stable in value even though processor speeds have increased several orders of magnitude over the past few decades. Building upon the existing work of Elasticoin [7], which can trustlessly measure the value of a DOSC using non-interactive proofs of sequential work [4], we then create a mechanism that pegs a cryptocurrency to the DOSC. Melmint allows decentralized cryptocurrencies to maintain a long-term stable value without any trusted issuers or data feeds, solving a major open problem in cryptocurrency design.

2 Background and motivation

In this section, we take a look at the background of the cryptocurrency volatility problem. We first examine why cryptocurrency prices are so volatile and some less obvious problems this volatility causes, and then we take a look at existing work that attempts to stabilize cryptocurrency values. Finally, we argue that existing approaches are all inadequate and that a new mechanism is badly needed.

2.1 The problem of volatility

Ever since their inception, cryptocurrencies have been exceptionally volatile. In fact, they're probably some of the most volatile non-derivative financial assets in existence — Bitcoin on average fluctuates by more than 3% every day [23], orders of magnitude higher than fiat currencies, even though it has by far the most market liquidity of any cryptocurrency.

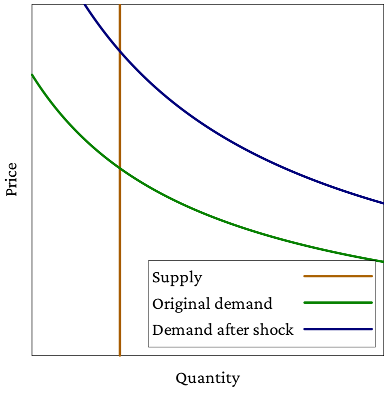


Figure 1: Inelastic cryptocurrency supply

Such extreme volatility is due to a combination of volatile demand and *perfectly inelastic supply*. Cryptocurrencies generally are issued on fixed schedules that totally ignore market conditions, leading to a situation illustrated in Figure 1: changes in demand cause sudden and large changes in price. However, volatility causes two serious problems:

- *The cryptocurrency ceases to be useful as money*. Since cryptocurrency units no longer have a stable purchasing power, they cannot fulfill the duties of a currency well. It becomes impractical to do business, store wealth, etc denominated in a cryptocurrency. This is a well-acknowledged problem [3, 13, 19] with volatile cryptocurrencies.
- *Cryptoeconomic mechanisms are destabilized*. More insidiously, with a volatile on-chain unit of value, it's much harder to design safe cryptoeconomic systems. For example, when analyzing Bitcoin's security it's common to assume that a rational self-interested entity wants to maximize profits denominated in bitcoins [8, 20]. Yet with almost every action inscrutably influencing the price of a bitcoin one way or another, it's very hard to be sure of any cryptoeconomic proofs. It also makes it hard to design cryptoeconomic rewards with defined sizes, an approach considered neglected [2] in current designs.

Thus, volatility is neither a transient issue due to volatile cryptocurrency adoption nor a small inconvenience that can easily be abstracted away. We believe that eliminating the extreme volatility of cryptocurrencies is crucial to their long-term success.

2.2 Externally-pegged stablecoins

Existing approaches at creating a stable cryptocurrency generally focus on pegging it to a real-world asset, generally the US dollar. In fact, "stablecoins" are generally defined simply as on-chain assets pegged to real-world ones. Let's examine some existing approaches to creating such pegged stablecoins.

Centralized currency boards The most straightforward family of stablecoins uses a trusted bank that promises to exchange each unit of on-chain cryptocurrency for a fixed amount of the off-chain asset to which the peg is maintained. Stablecoins in this category include Tether [22], TrueUSD [24], and many others. This arrangement is known as a *currency board* system, and it is used by many robustly pegged fiat currencies [25]: the Hong Kong dollar, for example, is essentially an IOU issued by the Hong Kong Monetary Authority for 0.128 US dollars.

Currency board stablecoins have an advantage in that as long as the bank is trustworthy, no economic shock of any size can disturb the peg. Even if all users suddenly dump the pegged coin, the bank always has enough assets to sell to maintain the peg. Unfortunately, such stablecoins suffer from the obvious flaw of *counterparty risk* — if the institution providing the backing is untrustworthy, everything collapses. This is not a far-fetched possibility: unsound fiat currency boards such as that of the Argentinian peso [12] have undergone total collapses, and the risk of unsound backing [10] is a significant factor hindering the adoption of Tether.

"Decentralized" stablecoins Many other stablecoins projects exist that eliminate counterparty risk altogether by eschewing a trusted bank. They instead use some form of *on-chain algorithmic monetary policy*: a control loop typically implemented in a smart contract autonomously adjusts the money supply to target an exchange rate. No trust is required in any centralized issuer to achieve a stable peg. The exact mechanism used varies wildly from system to system; the most popular such stablecoin, MakerDAO [21], uses a sophisticated mechanism centered around maintaining reserves of significantly more than $1 worth of ETH for every $1 coin issued so that the peg can be maintained even when drops in the value of ETH wipe out a large percentage of the value of the reserves.

There are two significant problems with all non-currency-board stablecoin proposals though. First, *issuing an asset A pegged to asset B without holding asset B* is profoundly difficult, yet such algorithmic stablecoins must be able to peg a currency to, say, US dollars without the ability to hold any dollars. The challenge is comparable to that of a central bank attempting to peg a currency to the USD without any foreign exchange reserves, or a commercial bank investing depositors' dollars entirely in assets like commodities and foreign-currency bonds whose values are decoupled from that of the dollar. A way of doing either task safely would prove very profitable in the existing financial world; the fact that nobody engages in such business is strong evidence that it's in some way uninsurably risky.

More importantly, even stablecoins without a central issuer require trusted *oracles* to feed in the current price of the stablecoin, which is crucial to driving the algorithmic monetary policy. Although mitigations such as using the median of multiple oracles do exist, measuring facts external to the cryptoeconomic mechanism — the "oracle problem" — is one of the major fundamental issues with smart contracts in general [29]. Attempts at making decentralized oracles, such as SchellingCoin, typically fall to clever game-theoretical attacks that collapse their security entirely.

Thus, both currency board and "decentralized" stablecoins fundamentally still rely on centralized trust. We conclude pegging to blockchain-external currencies is probably not the right path towards trustlessly eliminating cryptocurrencies' drastic volatility.

2.3 Elasticoin: low volatility through elastic supply

Surprisingly, there aren't a lot of detailed proposals for endogenously stabilizing a cryptocurrency's value in the literature. For many years, the only things we had were vague suggestions of econometrically measuring on-chain activity [13, 26] to calculate the desired money supply. The most concrete proposal was probably a blog post by Vitalik Buterin [3] that attempted to construct a complex model to deduce the dollar-denominated price of Bitcoin from blockchain-endogenous metrics, though he admits that the model can be gamed.

Elasticoin [7] is the first detailed proposal for trustlessly reducing cryptocurrency volatility. Its core concept is to *fix the cost of minting a coin* to that of a certain quantity of "wasted" sequential computation time on the fastest processor available. Such a "proof of wasted time" can be trustlessly validated by combining non-interactive proofs of sequential work [4] with a continually updated on-chain speed record.

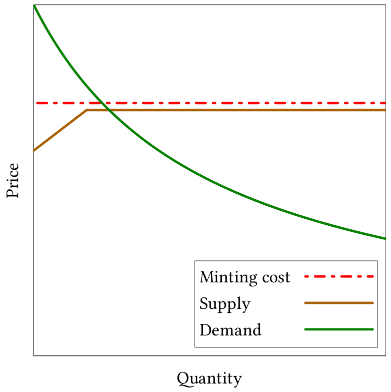


Figure 2: Supply and demand of an Elasticoin-issued currency

Figure 2 illustrates the supply and demand curves for newly-issued Elasticoin. Supply is very elastic, since whenever demand pushes the price of a coin above the cost of creating it, anybody can mint coins and take a risk-free profit. This one-sided arbitrage effectively establishes a limit to the price for the issued cryptocurrency.

Elasticoin reduces volatility in two ways. The most obvious one is that broadly stagnant or growing demand will result in a stable price very close to the cost of minting. Less obviously but far more importantly, Elasticoin *cuts off speculative demand*.

Cryptocurrency demand has been analyzed as broadly consisting of two parts: transactional demand CD_T from people seeking to use the currency to buy goods or hold as a short-term store of value, added to speculative demand CD_S based on rational expectations of higher values in the future. Anecdotally, demand for most cryptocurrencies is dominated by CD_S, but with Elasticoin, $CD_S \approx 0$ in a steady-state economy because there is no expectation of higher future values at all.

Thus, Elasticoin both flattens the supply curve and dampens movements in the demand curve, achieving low volatility in normal conditions without stablecoin-like oracles or financial instruments. Unfortunately, this is not enough to create a truly stable cryptocurrency that has rock-solid value even in abnormal economic environments.

2.4 Supply elasticity is not enough

The major problem with Elasticoin is that even though supply elastically expands when demand for currency is high, when demand is low supply cannot contract. This is illustrated by the "knee" in the supply curve in Figure 2. In fact, starting from a steady state where quantity supplied matches quantity demanded and the price is close to the ceiling, any drop in demand will cause a commensurate drop in price. Furthermore, if economic shocks cause demand to suddenly decrease, the price may become so far away from the minting cost that even the CD_S-damping effects of Elasticoin become irrelevant. Elasticoin's volatility would simply degenerate to be similar to that of a traditional cryptocurrency.

Elasticoin does make an important contribution in creating a *one-way* "peg" between a cryptoasset and a trustlessly measurable value unit, but it's clear that a different approach is needed to truly achieve our goal of a trustless stable cryptocurrency. Specifically, not only do we need high supply elasticity for newly minted coins, but also a way of reducing the number of coins in circulation when there's no demand for new coins.

3 Design

We will now discuss the design of Melmint, our solution to the trustless stable cryptocurrency problem. We first establish the context of Melmint's formulation, as a component of the prototype Themelio blockchain. We then describe a variant of Elasticoin we use for establishing a trustless value unit, the DOSC (day of sequential computation). Finally, we detail the Melmint algorithm itself.

3.1 Context: the Themelio blockchain

In this paper, we will largely discuss Melmint within the context of its original intended application — Themelio [15], an in-development "layer-0" blockchain focused on simplicity and robustness. This is because Themelio has trustless currency stability as one of its major goals.

Themelio is a proof-of-stake, UTXO-based[1][28] blockchain with a distinctive and relevant feature — it has two separate built-in cryptoassets. *Sym*[2] has a fixed supply and is staked to participate in the consensus process, while *Mel*[3] is used as the circulating currency that all in-protocol fees and rewards are paid in. Mels are intended to have a stable value.

As its name suggests, we created Melmint as a proposal for a mel-minting procedure for Themelio. It is, however, easily portable to many other blockchains, as we will discuss in 3.5.

3.2 Establishing a trustless value unit

As a building block for Melmint, we introduce a new built-in cryptocurrency, the DOSC. DOSCs are created using the Elasticoin algorithm targeting a minting cost of 24 hours of sequential computation per DOSC. This is similar to the way mels are currently minted.

DOSC UTXO balances, however, are subject to a 0.1% per block *punitive demurrage*. That is, every DOSC existing in the blockchain shrinks to 0.999 DOSC every time a new block is created. At the 30-second block interval of Themelio, a newly minted DOSC will be reduced to a mere 5% of its value after a day.

Thus, the DOSC is a "perishable" asset utterly useless as money, but in exchange, the circulating supply of DOSCs will be overwhelmingly dominated by newly minted DOSCs. The current value of a DOSC, therefore, cannot deviate far from the cost of a "day of sequential computation" of the most efficient minter. This is an important property that we will exploit in designing Melmint's core mechanism.

3.3 Melmint's core mechanism

The main objective of Melmint's core mechanism is to hold the price of 1 mel around 1 DOSC. There are two separate, simultaneous processes: an auction that establishes the value of a *sym* in DOSC, and a Mel-Sym exchange guarantee that backs every mel with 1 DOSC worth of Sym.

DOSC-sym auction We continually auction newly created syms for DOSC to establish the DOSC/sym exchange rate. To do so, we first divide time into an infinite number of *auction epochs* $E_0, E_1, \ldots$ where E_i lasts the 20 blocks (around 10 minutes) from block $20i$ to $20(i+1)$. At the start of each auction epoch E_i, we calculate δ_i, the number of new syms to sell, based on T_i, the total amount of syms in circulation:

$$\delta_i = \max\left(1, \frac{T_i}{2^{22}}\right)$$

This formula results in the supply of syms growing slowly at approximately 1.2% a year.

After establishing how many syms to auction off, we start the auction itself. The auction has two phases. During the first phase lasting 10 blocks, bidders may submit *bid transactions* to the blockchain offering to buy all δ_i syms in exchange for a certain number of locked DOSCs.

The second phase lasts for the remaining half of the epoch. Users can no longer submit any bids, but they may submit *buyout transactions*. A user Alice uses a buyout transaction to "buy out" a specific bid transaction from Bob by sending δ_i to Bob; Alice then gets the DOSCs offered by Bob, and Bob's bid will no longer be considered in the auction. This improves price discovery by incentivizing other users to buy out unreasonably high bids.

At the end of the entire auction epoch, the highest bid that hasn't been bought out is automatically accepted and the DOSCs locked within are destroyed. All other bids are rejected, with the associated DOSC balances returned to their owners. We record the DOSCs/mel price of the highest bid in epoch E_i as p_i; this gives us a dynamic feed of the current DOSC-denominated value of a sym that updates every 10 minutes.

[1] That is, transactions are similar to Bitcoin, unlocking script-encumbered *unspent transaction outputs* and creating new ones, rather than transferring money from one account to another as in Ethereum

[2] from Greek *metakhi*, meaning "equity share"

[3] Capitalization of "Mel" vs "mel" is similar to that of Bitcoin vs "bitcoin" — the currency itself is "Mel", but the unit is "mel", e.g. "a cup of coffee costs 2 mels"

Stabilizing mel value With a $\{p_i\}$ price feed, we can now stabilize the value of the mel. We guarantee that anyone can destroy 1 DOSC worth of syms to obtain 1 freshly minted mel, or destroy 1 mel to obtain 1 DOSC worth of new syms.

More specifically, at any time during an auction epoch E_i anyone can submit a *mel minting* transaction that destroys $(1+\epsilon)\ell/\varkappa\tilde{p}_i$ syms to create ℓ mels or a *sym minting* transaction that destroys $\varkappa(1+\epsilon)\tilde{p}_i t$ mels to create t syms. $\tilde{p}_i$ is a smoothed DOSC/sym exchange rate estimator derived from the median of the values $p_{i-5},\dots,p_{i-1}$ from the past 5 most recent epochs, while $\epsilon = 2^{-6}$ is a 1.56% minting fee that prevents wasteful arbitrage exploiting the inherent imprecision and time lag of $\tilde{p}_i$. $\varkappa$ is a "devaluator" that is generally equal to 1; it's used in emergency situations to devalue the mel when attempting to maintain the peg threatens systemic collapse (see 3.4).

Discussion Taken as a whole, Melmint effectively pegs each mel to 1 DOSC worth of syms, stabilizing the price of 1 mel to around the cost of wasting one day of sequential computation to create 1 DOSC. This is because arbitrage opportunities exist that push mel prices towards 1 DOSC no matter whether mels are too expensive or too cheap.

During periods of increasing demand, the price of a mel would rise until it reaches $1+\epsilon$ DOSC, at which point it becomes profitable for anyone to mint DOSCs, buy syms with them, then exchange them for newly printed mels. This increases the supply of mels until the price decreases such that no arbitrage is possible, establishing an Elasticoin-like hard ceiling on the mel price.

When demand decreases, mels may depreciate until they are no longer worth $1/(1+\epsilon)$ DOSC. At this point, a different kind of arbitrage becomes profitable — anyone can buy mels for the equivalent of less than $1/(1+\epsilon)$, exchange them for newly minted syms, and sell the syms for more than $1/(1+\epsilon)$ DOSC. This process will be repeated until enough mels are destroyed that the price increases back towards 1 DOSC.

One important observation is that through the sym-minting process, *Melmint backs the value of a mel by expropriating sym holders*. When new syms are created in exchange for destroying mels, this directly dilutes the value of one sym. In the long run this is balanced by the mel-minting process destroying syms and raising their value, but in any case this means that people holding syms contribute reserve capital to back Mel value, and the total market capitalization of Sym is a good estimate of the "implicit reserves" that Melmint has to defend the peg. We will show in 4.1 that these reserves are almost certainly many times the amount of circulating mels, ensuring the stability of the mel-DOSC peg.

3.4 Recovering from emergencies

In extreme circumstances, however, the peg might become impossible to maintain, at least in the short run. For example, a general panic in cryptocurrency markets may cause the value of a sym to drastically plummet, until the implicit reserves derived from metholder expropriation can no longer support a 1 DOSC/mel value. This will in turn incentivize a fatal run on the mel. Everyone owning mels would wish to immediately exchange them for 1 DOSC worth of syms, since those "first in line" would be able to get 1 DOSC worth of value while those left behind would have nothing. The sym would then rapidly hyperinflate, ruining the value stabilization mechanism and possibly leading to the collapse of Themelio as a whole.

We intend Melmint to issue a long-term value-stable cryptocurrency, not something rigidly tied to the rather esoteric unit of DOSCs. The DOSC/mel peg is not worth dying for! Instead, Melmint includes an "circuit breaker" to devalue the peg when impending insolvency is detected, as well as a mechanism to gradually recover the original peg at a pace that the market can bear.

Emergency devaluation At every block height, we calculate how much the sym supply has grown since either 86400 blocks (30 days) ago or the last emergency devaluation, whichever is closer. If this number is greater than 20% of the entire sym supply, we consider there to be a serious threat of sym hyperinflation. An *emergency devaluation* is immediately triggered, reducing $\varkappa$ to $3/4$ its previous value. Emergency devaluations may follow one after another if reducing $\varkappa$ once still does not stop a dangerous rate of sym inflation. Eventually, $\varkappa$ should reach a value where the market is in equilibrium where the "20% rule" will no longer be triggered, and the Melmint peg will operate at this devalued exchange rate.

Peg recovery Melmint does not treat a devalued peg as permanent — as our subsequent evaluations will show, in a steady state there should always be enough implicit reserves to back a 1 DOSC/mel exchange rate. Thus we engage in tentative *peg recovery* after devaluation. At the end of every auction epoch where $\varkappa < 1$, we compare the number of mels created to those destroyed within the epoch. If the mels created exceed those destroyed, we know that the market will bear a slightly higher peg, so we increase $\varkappa$ to $\min(1, 1.001\varkappa)$. But if more mels are destroyed, we decrease $\varkappa$ to $0.9991\varkappa$. This effectively establishes an upwards-biased "crawling peg" that should eventually restore $\varkappa$ to 1 if implicit reserves are sufficient to support such a peg.

3.5 Porting to other blockchains

Given the almost entirely blockchain-agnostic description of Melmint above, it is straightforward to implement Melmint on smart-contract blockchains such as Ethereum and EOS. None of the core functionality of Melmint uses any blockchain-specific "black magic".

The main subtlety is the definition of a sym: syms are intimately tied to consensus participation and stakeholder rewards in Themelio, while a non-Themelio deployment of Melmint is clearly unable to issue any cryptocurrency with such powers. A "useless" sym with no inherent value will not work, as such a token would not provide nearly enough implicit reserves.

The most important property of the sym in Themelio is that its value is largely based on revenues from transaction fees, and thus is proportional to the total economic value transacted within the mel-using ecosystem as a whole. As we will see in 4.1, this *fee-based sym valuation* is crucial to Melmint's robust stability.

Fortunately, replicating this on other blockchains is fairly easy: one can simply have sym-holders split a small percentage fee on each mel-denominated transaction to simulate Themelio's transaction fees. This will then make the total market capitalization of syms proportional to mel-denominated economic activity.

4 Evaluation

In this section, we first analyze the stability of the system using both qualitative arguments and quantitative data from real-life financial markets. We then examine the security of the system against attack and the cryptoeconomic incentives involved. Finally, we compare Melmint to the existing literature, showing that no previous system has achieved both trustless operation and robust stability.

4.1 Stability of implicit reserves

We start with a rough but very conservative analysis of Melmint's stability. As we've previously mentioned, the market cap of syms acts as an implicit reserve that is drained when syms are inflated to buy and destroy mels. Thus, the ratio of the total value of all circulating syms to that of all circulating mels — the *implicit reserve ratio* — must be above 1 to guarantee stability.

Let us estimate what this ratio would be in a realistic blockchain economy. Mets derive their value by "taxing" mel transaction activity through fees, block rewards, etc. This process generally extracts some small fraction of the total economic value transacted in mels. In Bitcoin, the proportion of the total transaction volume captured as miner revenue is around 2-10%, a number curiously similar to the percentage of GDP raised by a wide range of premodern taxes on vital commodities, such as salt taxes in imperial China [9]. As a safe estimate, let us assume that the revenue r captured by metholders is 2% of mel-denominated economic activity Y: $r = 0.02Y$.

We can now estimate the market capitalization of syms through a discounted cash flow model: given a discount rate of d, the total value of all syms Θ would be:

$$\Theta = \sum_{i=0}^{\infty} r(1-d)^i = \frac{0.02Y}{d}$$

Assuming a typical discount rate of $d = 0.03$, this gives $\Theta = 0.67Y$. This is then also the upper limit of the total value of syms we can safely issue. But examining existing economies, we see that $0.67Y$ is generally well above the amount of currency in circulation. In Table 1, we list the ratio of money supply to annual economic activity for the US economy and for Bitcoin. For the US, we use the M1/GDP ratio, while for Bitcoin we divide the total number of bitcoins by 360 times the daily on-chain transaction volume. We see that in both cases the ratio is well below our conservative estimate of 0.67.

Economy	5% percentile	Median	95% percentile
USA (since 1960)	0.105	0.149	0.245
Bitcoin (since 2014)	0.107	0.228	0.495

Table 1: Ratio of money supply to yearly economic activity

Thus, we would expect Melmint's implicit reserves to be more than enough to withstand even the largest "runs" on the mel.

4.2 Stochastic market simulation

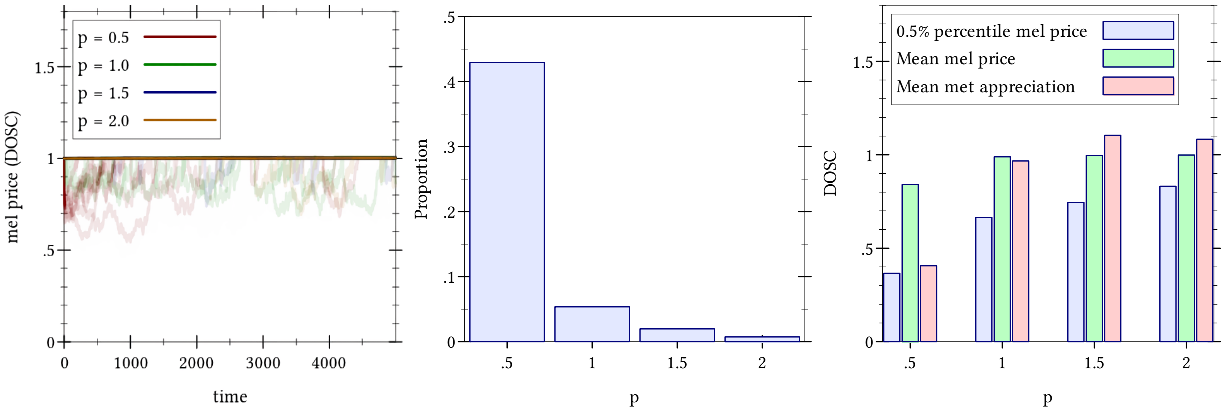
We've shown that "on average" Melmint should give a very robust peg, but how would it behave in a wide variety of extreme economic conditions? We build a stochastic simulation of a cryptocurrency market to investigate Melmint's behavior.

Setup We simulate the Melmint mechanism on a simple market model containing the following four variables varying with time:

- Total *sym supply* T in circulation
- Total *mel supply* M in circulation
- Current *sym price* p in DOSCs
- Current *mel price* q in DOSCs

We then simulate fluctuating demand for both syms and mels: every simulated day, both the current sym price and the quantity of mels demanded randomly changes by at most 1%. Random adjustment of the sym price is done by simply changing p, while we run the Melmint mechanism to create or destroy mels until the quantity demanded is sym.

Running the Melmint mechanism also changes the sym supply and therefore price; this is harder to model since it depends on the demand curve of syms. As a simplifying assumption, every time the mechanism changes the amount of syms by a factor x, we change the sym price by a factor $x^{-1.1}$: $T \leftarrow Tx \implies p \leftarrow px^{-1.1}$. Thus, decreasing sym supply increases the sym price but also the sym market capitalization due to market expectations of further contraction, while increasing sym supply does the opposite. We chose a small exponent of x so that indefinite sym inflation will deplete our implicit reserves — keeping sym market capitalization constant would



(a) 10 runs of the simulation

(b) Proportion of time spent unpegged

(c) General stability statistics

Figure 3: Market simulation results

instead allow an infinite amount of value can be raised from inflation, since $\sum_{x=0}^{\infty} 1/x$ does not converge.

Finally, when the implicit reserve ratio Mq/Tp falls below 1, demand for mels rapidly vanishes and we attempt to destroy 5% of all outstanding mels every day. This simulates a panic where the implicit reserves backing mels fail, and a run on mels happens.

Normal case simulation We run a 5000-day simulation of Melmint, with an initial state of $T = M = 1000$, $q = 1$. We vary our starting sym price p to simulate different "safety margins"; for each p we run the simulation 100 times to determine average behavior. The results are summarized in Figure 3.

We note that with $p = 0.5$, almost half of the simulation is spent with a broken peg. This is of course expected since the implicit backing is only 50% of the necessary value. We also note that the sym depreciates to less than half of its original value due to the large amount of sym inflation triggered by attempts to maintain the peg. Qualitatively we see in Figure 3a that most of the dips in price come from the $p = 0.5$ case.

As p increases above 1, the peg becomes extremely robust. A negligible fraction of the time is spent with a broken peg, and even during emergency devaluations the mel price is very close to the peg. Finally, at $p = 2$, a value that gives an implicit reserve ratio close to our previous predictions, the peg can be considered always solid. We don't see any sign of depegging in Figure 3a associated with $p = 2$.

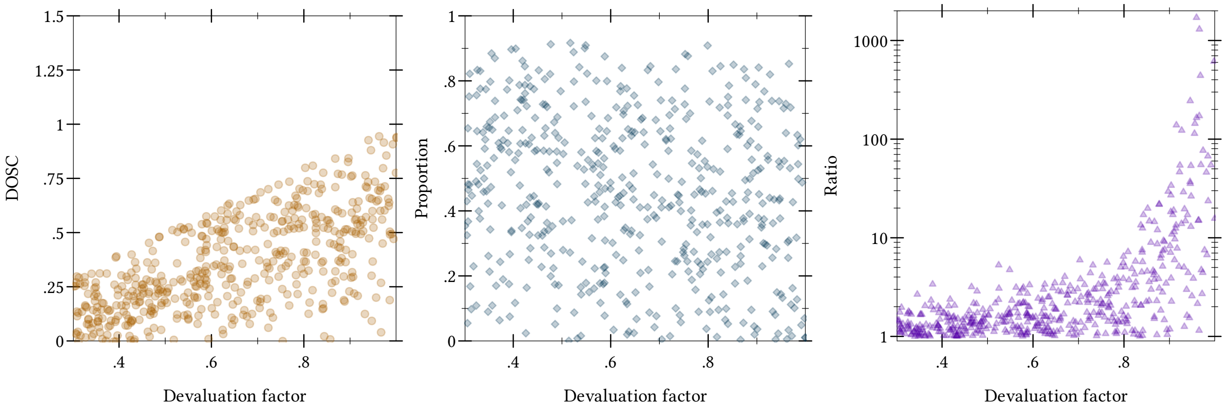
Devaluation stress test We've shown that Melmint behaves quite robustly under randomly varying conditions. Let's now investigate in more detail a crucial component to Melmint's robustness — emergency devaluation and peg recovery. In particular, we examine the effect of varying the *devaluation factor*, or how much x should drop when we detect a panic. We set the devaluation factor to 3/4 in 3.4, but there is a tradeoff involved — a number closer to 1 makes the devaluation during a panic smaller, but increases the chance that the devaluation is insufficient and multiple devaluations with lots of sym inflation will happen.

To test peg recovery, we use the same random-walk model as we did previously, except that each day demand for syms has a 51% chance of decreasing instead of 50%, while the demand for mels does not show this downwards bias. This simulates a an economy with rapidly dwindling demand for syms — a "the cryptocurrency is dying" scenario. Repeated panics driven by insufficient implicit reserves will almost certainly occur. We run a 5000-day simulation 500 times, each time with a random devaluation factor between 0.3 and 1, and plot the results in Figure 4.

In 4a we see a clear linear relationship between devaluation factor and the minimum price the mel falls to. Of course, this is because the devaluation factor directly controls how much we devalue during crises. A more interesting observation is that it does not seem like harder devaluation factors significantly decrease the occurrence of multiple devaluations: the proportion of samples with the minimum mel price corresponding to a single devaluation doesn't seem to change with devaluation factor. This is corroborated by 4b, showing that devaluation factor doesn't affect the time spent unpegged.

Figure 4c illustrates the dramatic increase in sym issuance as the devaluation factor approaches 1. In fact, we see *superexponential* growth on the log-linear plot. In reality, Melmint would not be stable at all for factors greater than around 0.8, since the sym supply inflating hundreds of times or more would almost certainly destroy the usefulness of the system.

Finally, in Figure 5 we illustrate the danger of sym hyperinflation by testing the exact same mel and sym demand patterns against two devaluation factors. When we use a factor of 0.75, the mel persistently depegs, but the sym retains a significant fraction of its value. But attempting to strongly hold the peg results in the mel depreciating anyway, while the sym totally collapses in value. This shows that a sizeable re-peg



(a) Minimum mel price

(b) Proportion of time spent unpegged

(c) Maximum sym inflation

Figure 4: Devaluation tests

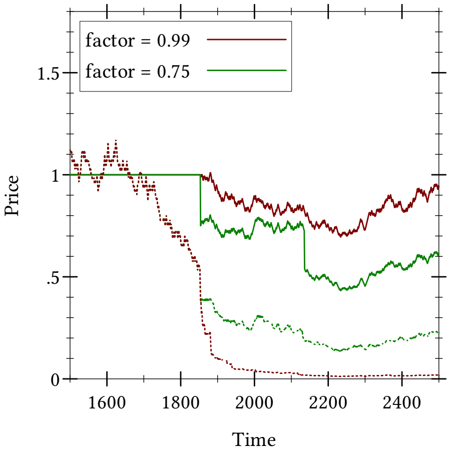


Figure 5: A specific test instance. Mel prices use solid lines while sym prices use dashed ones.

during emergency devaluation is necessary yet sufficient to give Melmint robust stability even during crises that force a depegging.

4.3 Cryptoeconomic security

We analyzed the stability of Melmint under the assumption that all the mechanisms work without disruption. Now, let's examine what would happen when adversaries deliberately attempt to destabilize the system.

We look at two kinds of attacks separately: *mechanism-level* attacks which attempt to manipulate a Melmint mechanism working on a perfectly trustworthy blockchain, and *blockchain-level* attacks that subvert Melmint by disrupting the underlying blockchain. In both cases, we assume that the purpose of the attacker is to de-peg the mel from the DOSC and destroy its value stability.

Mechanism-level attacks In a mechanism-level attack, an adversary attempts to disrupt the price of a mel by engaging in actions allowed by the protocol. One possible avenue is by simply attacking the peg directly. The attacker might try to sell or buy huge amounts of mels, affecting their price. However, this is extremely costly and ineffective, since the minting mechanism in 3.3 ensures that other people can use the mechanism to profit unboundedly off of market manipulation attempts, the peg would stay safe, and the attacker would lose a great deal of money. For example, if the attacker tries to buy up lots of syms to jack up the price, anybody can repeatedly turn in the DOSC-equivalent of syms to get mels and sell them to the attacker. In fact, the robustness of Melmint's mechanism rests on this sort of arbitrage.

Two crucial parts of Melmint, though, do not rest on such an obvious two-way arbitrage: DOSC minting and sym auctioning. Attacks against DOSC minting are not possible without breaking the Elasticoin mechanisms [7]. Against the sym auction, the attacker may want to cause either an overvalued or undervalued sym price to be measured. This would destabilize the system and cause fluctuations in mel prices.

Fortunately, the sym auction is protected by another, less direct kind of arbitrage. Attackers attempting to bid a below-market number of DOSCs for the newly minted syms will not succeed, as they will be immediately outbid by bidders offering higher prices. Bidding an above-market price, on the other hand, incentivizes others to use syms to buy out the attacker's bid. These other people can then use the stockpile of DOSCs obtained from the attacker to bid for syms again and obtain more syms than the attacker was bidding for, gaining a profit.

The only possible avenue of attack is to simply bid unprofitably high prices at a loss so often that the market is flooded with DOSCs and thus the DOSC/sym exchange rate falls, but like market attacks by sheer expenditure in general, this is extremely costly and possible only with overwhelming market power.

Blockchain-level attacks The attacker may subvert the underlying blockchain in order to attack Melmint. Here we do not consider attacks that totally break the blockchain (say, 51% attacks that cause double spending) as defenses against them should be handled by cryptoeconomic mechanisms within the blockchain itself. Instead, we consider "pathological" strategies that consensus participants (miners in Ethereum, stakeholders in Themelio) may follow that don't break the entire consensus but might damage the stability of Melmint.

In particular, we consider *frontrunning*, where the attacker has knowledge of blockchain transactions in advance of others, and *censorship*, where the attacker prevents transactions from reaching the blockchain. In both cases, we assume an extremely powerful attacker: one that has a consensus monopoly able to control what goes onto the blockchain, rationally seeks to maximize profits, and does not have external incentives such as bribes or threats. Under these circumstances, we do not want to create an incentive for this attacker to do any action that would damage Melmint's stability — thus, the cryptoeconomic incentive structure of the underlying blockchain would suffice.

In a frontrunning attack, the adversary uses its privileged position to observe Melmint transactions, such as DOSC-minting and sym auctions, before others see them. It's likely possible to extract some profit from this information — for example, by dumping mels just in advance of an emergency devaluation. But in line with existing economic research [6, 11], we expect any frontrunning to actually increase the efficiency of the markets established by Melmint, helping rather than harming its stability.

Censorship is a much more serious issue. It's quite obvious that with total control over what transactions go onto the blockchain, an adversary can destabilize Melmint as much as she wants — after all, Melmint operates entirely with blockchain-based inputs. However, will there be a mechanism-internal incentive for a blockchain-controller adversary to do so?

Although we haven't yet done a rigorous game-theoretical analysis, we conjecture that the answer is no, there's nothing to gain out of manipulating Melmint for a blockchain-controlling adversary. In Themelio's instantiation of Melmint where syms are controlling shares in both protocol revenue and consensus, there is a fairly intuitive argument that manipulation will not be profitable — manipulating the exchange rate would almost certainly cause syms to depreciate, and blockchain-controlling adversaries necessarily own a vast amount of syms. One might guess that large metholders would want to censor all sym minting to prevent their share from diluting, but in fact they don't have an incentive to do so assuming an efficient market, since any sym minting blocked would only accumulate and happen all at once at the end of the attack, and expectations of this would depreciate syms just as much as actual sym minting would.

	Trusted parties	Strong peg?	Reserves
Tether	Issuer	Yes	Bank deposits
MakerDAO	Oracles	Probably	On-chain collateral
Seign. Shares	Oracles	Uncertain	Seigniorage-based
Basis	Oracles	Uncertain	Seigniorage-based
Elasticoin	None	No peg	None
Melmint	None	Yes	Fee-based

Table 2: Comparison of Melmint to other systems

4.4 Comparison to existing systems

Finally, we compare Melmint with existing work on reducing cryptocurrency volatility. Table 2 compares three important aspects of a cryptocurrency issuance mechanism: the *trusted parties*, whether or not a *strong peg* to some stable index is achieved, and the implicit or explicit *reserves* backing the peg.

We give Tether [22] as an example of a traditional centralized stablecoin: it's issued by a trusted party, backed by fiat reserves, and maintains a robust peg assuming the issuer is reliable. MakerDAO [21] is the most popular stablecoin without a trusted issuer, relying on only a trusted oracle that publishes up-to-date exchange rates with the US dollar. It has a unique system roughly reminiscent of non-deliverable forwards used to trade non-convertible currencies in traditional finance; its reserves used to support the coin's value is an on-chain reserve of cryptocurrency almost always worth more than the issued coins.

Seigniorage Shares [19] and its now-defunct [16] descendant Basis [17] are the stablecoins closest in design to Melmint. Like Melmint, both couple the stablecoin with a secondary volatile asset ("shares" or "basebonds" or "syms") that is inflated and deflated to support the stablecoin's peg. Reserves are therefore implicit and roughly equivalent to the market capitalization of the secondary asset. Unfortunately, both seigniorage shares and Basis use secondary assets with value derived solely from expected future inflation (seigniorage) of the main coin. This makes the implicit reserves only sound when there is steadily and rapidly increasing demand; seigniorage in a normal fiat currency is usually a minuscule fraction of total circulating currency. In fact, when demand is expected to decline in the future, the secondary assets would actually have *negative* value! Thus, seigniorage-backed stablecoins may be fundamentally unsound, as several analyses of Basis have concluded.

Finally, Elasticoin [7] introduced the concept of a low-volatility trustless cryptocurrency, but it does not have a stable peg and has problems with volatility during periods of low demand. Melmint synthesizes Elasticoin with a Seigniorage Shares-like implicit reserve with a *fee-based* value that's tied to the total economic value of the system, rather than a self-referential monetary policy variable like seigniorage. Thus, Melmint achieves the "holy grail": no trusted parties, a strong peg, and robust reserves.

5 Conclusion

In this paper, we presented Melmint, a new cryptocurrency issuance scheme that robustly pegs issued coins to the DOSC, a unit measuring the cost of sequential computation for a day. This is done by combining Elasticoin, an existing algorithm for measuring the value of a DOSC, with a strong implicit reserve based on diluting shares of fee revenue. Unlike all other stablecoin proposals, Melmint operates without any trusted issuers or oracles while maintaining a robust peg. We show through both qualitative argument and extensive stochastic simulations that Melmint does indeed achieve its goals.

References

References

[1] *BitPay*. URL: https://bitpay.com (visited on 10/16/2019).

[2] Vitalik Buterin. *DRAFT: Position paper on resource pricing*. 2018. URL: https://ethresear.ch/t/draft-position-paper-on-resource-pricing (visited on 10/03/2019).

[3] Vitalik Buterin. *The search for a stable cryptocurrency*. Nov. 11, 2014. URL: https://blog.ethereum.org/2014/11/11/search-stable-cryptocurrency/ (visited on 08/15/2019).

[4] Bram Cohen and Krzysztof Pietrzak. "Simple proofs of sequential work". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 451–467.

[5] CoinMarketCap. *Cryptocurrency Market Capitalizations*. URL: https://coinmarketcap.com (visited on 10/15/2019).

[6] Jean-Pierre Danthine and Serge Moresi. "Front-running by mutual fund managers: a mixed bag". In: *Review of Finance* 2.1 (1998), pp. 29–56.

[7] Yuhao Dong and Raouf Boutaba. "Elasticoin: Low-Volatility Cryptocurrency with Proofs of Sequential Work". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2019, pp. 205–209.

[8] Ittay Eyal and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable". In: *Communications of the ACM* 61.7 (2018), pp. 95–102.

[9] Albert Feuerwerker. "The state and the economy in late imperial China". In: *Theory and Society* 13.3 (1984), pp. 297–326.

[10] John M Griffin and Amin Shams. "Is bitcoin really untethered?" In: *Available at SSRN 3195066* (2018).

[11] Thorsten Hens, Terje Lensberg, and Klaus Reiner Schenk-Hoppé. "Front-Running and Market Quality: An Evolutionary Perspective on High Frequency Trading". In: *International Review of Finance* 18.4 (2018), pp. 727–741.

[12] John F Hornbeck and Meaghan K Marshal. "The Argentine financial crisis: a chronology of events". In: Congressional Research Service [Library of Congress]. 2002.

[13] Mitsuru Iwamura et al. "Can we stabilize the price of a Cryptocurrency?: Understanding the design of Bitcoin and its potential to compete with Central Bank money". In: *Understanding the Design of Bitcoin and Its Potential to Compete with Central Bank Money (October 25, 2014)* (2014).

[14] William Stanley Jevons. *Money and the Mechanism of Exchange*. Vol. 17. Kegan Paul, Trench, 1885.

[15] Themelio Labs. *Themelio*. URL: https://themelio.org (visited on 10/13/2019).

[16] Nader Al-Naji. *Basis update*. 2018. URL: https://medium.com/basis-blog/basis-update-ae96e3565b1d (visited on 09/15/2019).

[17] Nader Al-Naji, Josh Chen, and Lawrence Diao. *Basis: A Price-Stable Cryptocurrency with an Algorithmic Central Bank*. 2017.

[18] Satoshi Nakamoto et al. "Bitcoin: A peer-to-peer electronic cash system". In: (2008).

[19] Robert Sams. *A Note on Cryptocurrency Stabilisation: Seigniorage Shares*. Aug. 27, 2019. URL: https://github.com/rmsams/stablecoins.

[20] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. "Optimal selfish mining strategies in bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 515–532.

[21] Maker Team. "The Dai Stablecoin System". In: (2017). URL: https://makerdao.com/whitepaper/DaiDec17WP.pdf.

[22] *Tether*. URL: https://tether.io (visited on 10/16/2019).

[23] *The Bitcoin Volatility Index*. URL: https://bitvol.info/ (visited on 10/01/2019).

[24] TrustToken. *TrueUSD*. URL: https://www.trusttoken.com/trueusd (visited on 10/03/2019).

[25] Alan Walters. "Currency boards". In: *Money*. Springer, 1989, pp. 109–114.

[26] Ethereum Wiki. *Hard problems of cryptocurrency*. URL: https://github.com/ethereum/wiki/wiki/Problems (visited on 08/15/2019).

[27] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[28] Joachim Zahnentferner. “Chimeric Ledgers: Translating and Unifying UTXO-based and Account-based Cryptocurrencies.” In: *IACR Cryptology ePrint Archive* 2018 (2018), p. 262.

[29] Zibin Zheng et al. “Blockchain challenges and opportunities: A survey”. In: *International Journal of Web and Grid Services* 14.4 (2018), pp. 352–375.