Know Your Assumptions

A Treatise On Assumptions in Cryptocurrencies and DeFi.
Revision 0.1

Alexander Chepurnoy, Amitabh Saxena

February 1, 2023

Abstract

1 Introduction

This article is going to provide affirmative answers to few important questions which seem to be largely ignored in the cryptocurrency industry, despite the fact that the questions are very basic ones, and, in our opinion, existence of the industry is heavily relying on them. Also, we systematize knowledge about assumptions cryptocurrency and DeFi protocols are based on. We suppose that understanding explicit or implicit assumptions protocols are based on is critical for understanding security and degree of decentralization of protocols properly.

- Traditional financial institutions are doing privacy invasive checks known as KYC/AML practices. In bankless world of cryptographically-powered decentralized finance it is natural to expect reversion of control, that is, now users are checking protocols for their properties, including cryptographic and other assumptions behind them. However, there is no explicit notion of it in the wild. Thus we would like to introduce Know-Your-Assumptions (KYA) practices as an alternative to KYC/AML practices in the world of decentralized money and financial tools, where users of the protocols do have real power, not centralized entities enforcing users to comply with their always-changing rules by giving out private data (always being leaked after). Some supporting notes:
- In traditional finance, an investment is always associated with not just a projected profit, but also with risk profile. In opposite, in DeFi only APY (estimated yearly interest rate at the moment, usually not averaged over long enough period and not cleared from noise even) is being sold, while the public is totally unaware about risks.
- As a consequence, it would be desirable for investors to quickly compare security of different protocols (from L1 blockchains to complicated DeFi stacks and inter-chain bridges) by assumptions they are based on.

- For protocol researchers and developers, it would be good to explicitly state assumptions their protocols are relied on in their papers and specifications.

- It is common, especially around so-called Bitcoin maximalists, to say, that, instead of trust in central authority, they trust math. However, we need to challenge that saying. Is Bitcoin protocol relies on math only? What does trust in math mean exactly?

In cryptography, starting from around 1980s, it is common to require provable security for all the constructions, starting from theoretical primitives, like one-way functions and pseudorandom generators, to very complex protocols. Provable security basically means that security goals of a construction are defined formally, as well as assumptions the construction is relied on, and then there is a proof that goals are indeed satisfied while assumptions hold. A proven construction can be then used as an assumption for a higher level protocol. In the very core there are some assumptions we can not reduce to other ones, so modern cryptography relies on a belief that in the world we are observing some very basic assumption holds, but those assumptions are well tested and centuries of math experience have not disproven them. We are trying to follow this approach in the inter-disciplinary field of cryptocurrencies, thus results would be not so theoretically strong, but nevertheless, useful.

We start with Bitcoin, the most well-studied and probably simplest cryptocurrency. Then we consider Ergo as well as some applications on top of it, such as ErgoDEX (now Spectrum), SigUSD, oracles, Dexy etc.

2 Bitcoin As A Digital Gold

We suppose that most of Bitcoin investors likely believe that Bitcoin is valuable digital commodity which has properties of *digital gold*, and electronic peer-to-peer cash at the same time (even if sometimes these concepts are contradicting).

Security of Bitcoin is based on following assumptions:

- Hash function SHA-256 is not broken. By being not broken we mean that known security properties of any hash function, namely collision resistance, second preimage resistance, and preimage resistance do hold.

- Digital signature scheme is not broken. As digital signature scheme used in Bitcoin is vulnerable in the presence of quantum computers, it means that the Bitcoin protocol also relies on assumption that quantum computers capable to solve Discrete Log Problem for an elliptic curve of 128-bit security will not appear.

- Majority of mining hashrate is *honest*, i.e. following the Bitcoin protocol. This is the most tricky part, as the protocol is defined via a reference client implementation, and it is not very clear what modification of pretty massive client codebase could be considered as dishonest. The best thing

we can do here is to work with simplified models of the protocol. The most famous one is presented in GKL15 paper [1]. For the first time in the space, the paper had shown that the Bitcoin-like Proof-of-Work based protocol can achieve some formally defined properties under assumptions of hash function collision-resistance and also majority (or 33% for fairness property) of hashrate power being *honest*, and the protocol defines what does it mean to be honest: a peer must build on a longest chain, use protocol's validation and input contribution rules. Real Bitcoin protocol, however, is much more complex than the GKL15 model. Continuing longest chain could not be strictly enforced by the Bitcoin protocol. For many years there was a tenet in the folklore that this behavior is just the most rational choice, but then some papers did show that rational behavior could be quite different from expected in some scenarios (see [2], [3], [4]). Also, while in the real protocol following validation rules seems to be necessary (otherwise, a mining node's block will be rejected by validating nodes), SPV mining was observed in the wild even [5] (and, theoretically, is a big concern when talks about increasing block size are happening, due to validation dilemma [6]). For input contribution, it is hard to say where honest behavior ends (especially for other protocols which are more feature-rich, where miners can profit a lot from reordering transactions and other games [7]). Please note that the field is still pretty green, so we do not know a lot about deviations from the protocol and how to react to them. And for some aspects of the Bitcoin protocol, e.g. miners behavior when emission per block will be much lower, we can only build theories, but then the reality which will be observed in pretty distant future could be quite different.

As you can see, even for Bitcoin assumptions are tricky, and then it is hard to say how successfully the protocol we have (in form of client code) may achieve its security properties, as even basis is not fully understood yet. However, as Bitcoin network works for more than 13 years to the moment of writing this treatise, many theoretical issues do not happen in practice (so maybe not of a concern), and issues that are not known likely are even more theoretical.

Note that Bitcoin has known emission schedule if assumptions hold. Thus Bitcoin as a digital gold can be better than physical gold, as for physical gold assumptions behind hardness of its production are less known. Thus *while assumptions do hold*, Bitcoin is perfect digital gold.

Alex notes : fun facts, e.g. Bitcoin protocol is the code, but originally the code had unlimited emission.

3 Ergo

Bitcoin assumptions, as shown above, are already tricky, but for other cryptocurrencies situation is even worse. They may have additional cryptographic assumptions (such as hard problems in pairing and trusted setup in ZCash),

or more complex reasoning about peers behavior or consensus among them (we can refer to endless disputes about Proof-of-Stake here). For Ergo, since day one the focus was on achieving possible maximum set of features securely from modest Bitcoin's set of assumptions, or even to rely on less.

A peer-to-peer network based cryptocurrency assumes that it is possible to run protocol client on commodity hardware of today (e.g. a mid-priced laptop). It is also assumed that client should allow to be an own bank, which is in case of Bitcoin is about validating all the blocks from genesis. Validating all the blocks could be expensive, especially with larger block size and UTXO set size [8]. This puts limits on block size, expressiveness of contracts etc. Ergo allows a node to have security guarantees of a node which validates all the blocks but without doing that. Even more, there are multiple techniques supported:

- Ergo node may have full security guarantees without storing UTXO set at all, by checking cryptographic proofs of UTXO set transformations. Unlike other stateless cryptocurrencies appeared later, such as Mina, Ergo relies only on hash-based authenticated data structures [9], so security of bootstrapping is based on collision-resistance of hash function used, i.e. no new assumptions introduced. Still, mining nodes do need to store UTXO set to generate proofs (otherwise, users would need to generate non-expirable proofs attached to their transactions, and for that exotic solutions, usually with heavy proving machinery, are required).

- Ergo node may download UTXO set snapshot after downloading block headers, and then download and then apply full-blocks only after the snapshot. This bootstrapping method is the most popular in Ethereum community already, but for Ergo (and other Proof-of-Work cryptocurrencies), there is known reduction of security of such bootstrapping to collision-resistance of a hash function [10].

- Ergo node may avoid downloading all the headers even, using NiPoPoW proofs instead [11]. Along with stateless clients, this allows for log-space storage for all the parts of the blockchain [12]. NiPopoWs are also rely on collision-resistance of a hash function, and also on assumption that adversary (attacking NiPoPoW sub-protocol) has no more than $\frac{1}{3}$ of total mining hashrate. With new assumption in mind, bootstrapping non-SPV client with NiPoPoWs should be done with large enough suffix.

In regards with signatures, Ergo is using Schnorr signatures (which are simpler, and security is formally proven) on top of secp256k1 curve used in Bitcoin. Ergo also allows for building cryptographic protocols as applications, based on sigma protocols supported by the Ergo protocol. Then some applications are based on (stronger than discrete logarithm) DDH (decisional Diffie-Hellman) assumption, but in practice this shift for minority of UTXOs is minor (security is literally the same).

In regards with honest behavior of miners, Ergo is pretty much the same as Bitcoin here, with notable exception of storage rent, which will hopefully simplify issues with miners incentivization in future.

As we can see, Ergo is not different from Bitcoin in its basic security assumptions, both cryptographic and protocol. At the same time, Ergo brings solutions for L1 scalability with the same level of security, and support for feature-rich contracts and cryptographic protocols.

4 DeFi apps on Ergo

5 Oracles

Oracles are delivering data from external world.

6 ErgoDEX

7 Djed and SigUSD

8 Dexy

References

[1] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pp. 281–310, Springer, 2015.

[2] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pp. 154–167, 2016.

[3] K. Liao and J. Katz, "Incentivizing blockchain forks via whale transactions," in *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pp. 264–279, Springer, 2017.

[4] A. Judmayer, N. Stifter, A. Zamyatin, I. Tsabary, I. Eyal, P. Gaži, S. Meiklejohn, and E. Weippl, "Sok: Algorithmic incentive manipulation attacks on permissionless pow cryptocurrencies," in *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers 25*, pp. 507–532, Springer, 2021.

[5] "Some miners generating invalid blocks." `https://bitcoin.org/en/alert/2015-07-04-spv-mining`. Accessed: 2023-01-30.

[6] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proceedings of the 22Nd acm sigsac conference on computer and communications security*, pp. 706–719, 2015.

[7] S. Eskandari, S. Moosavi, and J. Clark, "Sok: Transparent dishonesty: front-running attacks on blockchain," in *Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, pp. 170–189, Springer, 2020.

[8] "Utxo uh-oh. . . ." `http://gavinandresen.ninja/utxo-uhoh`. Accessed: 2023-01-30.

[9] L. Reyzin, D. Meshkov, A. Chepurnoy, and S. Ivanov, "Improving authenticated dynamic dictionaries, with applications to cryptocurrencies," in *Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21*, pp. 376–392, Springer, 2017.

[10] T. Duong, A. Chepurnoy, and H.-S. Zhou, "Multi-mode cryptocurrency systems," in *Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts*, pp. 35–46, 2018.

[11] A. Kiayias, A. Miller, and D. Zindros, "Non-interactive proofs of proof-of-work," in *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pp. 505–522, Springer, 2020.

[12] A. Kiayias, N. Leonardos, and D. Zindros, "Mining in logarithmic space," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3487–3501, 2021.