[

](https://ibb.co/F7vzbKS)

Introduction

Welcome to volume 3 of the Intents Newsletter. This issue indulges four topics; Intents, P2P, Distributed Systems, Restaking and Cryptoeconomic Safety.

In particular, featured is a brief discussion on how intents relate to binding conditional commitments from 50-year-old game theory. The newsletter also highlights a recent whiteboard presentation by @cwgoes at the modular summit, where he white boarded Anoma. The focus is on the p2p portion of his presentation. The distributed systems section highlights five recent papers including work on multiple concurrent block proposers as well papers featured at the Science of Blockchain conference 2024. The Newsletter concludes with two excellent articles on Restaking and Cryptoeconomic Safety. Without further prerequisites, let's jump in.

Acknowledgements

Thank you to all the brilliant researchers whose work is featured in this newsletter.

Table of Contents

Intents

[

](https://ibb.co/BrPc3gL)

Image: @vveiln

Program Equilibrium

In 1971, James W. Friedman published, [A Non-cooperative Equilibrium for Supergames

](https://www.jstor.org/stable/pdf/2296617.pdf). The key result established in the paper was that repeated interaction can achieve any feasible, individually rational payoff - that is, a payoff that a player can secure by playing their "safest" strategy, regardless of the actions of the other players.

Later, Moshe Tennenholtz in their 2004 classic Program Equilibrium

shows that in a program equillibrium of one-shot prisoner's dilemma games mutual cooperation can be obtained. More generally, Tennenholtz shows that the set of program equilibrium payoffs of a game coincides with the set of feasible and individually rational payoffs of it.

The idea of Program Equilibrium is that you can achieve any individually rational payoff even in a one-shot game

, if players instead of just taking actions themselves, they use commitment devices or programs whereby the players commit themselves to a strategy. Since this provides a credible guarantee for how players will act, then you can achieve any individually rational payoff.

We show that a set of program equilibrium payoffs of a game coincides with the set of feasible and individually rational payoffs of it.

If MyProgram = OtherPlayerProgram DO(coop); else DO(defect)

Program Equilibrium, is the closest recent research basis for intents. The research literature describes the problem pretty well, mathematically, in terms of what we want to get out of these systems.

Theorem 1:

There exists a program equilibrium for the prisoner's dilemma which yield's cooperation for both parties

Proof:

If both Players adopt this program, then the players have programs with the same syntax (they are equivalent to files), and therefore the condition IF

P_1 = P_2

holds, and DO(1,0) (i.e. cooperate) will be selected by both players. If one player adopts the specified program and the other player deviates from that program, then the above condition does not hold any more. Since in this case the program instructs DO(1,0) the payoff for the deviator is at most d< a

, which yields the desired result.

Theorem 2:

Given a game G, any individually rational strategy profile of it is played in some program equilibrium. As a result, the set of program equilibrium payoffs coincides with the set of feasible and rational payoffs of G.

Proof:

Assume that all players use the previously prescribed program (where feasible_i

and minimax(i, j

) are replaced by appropriate constants). Notice that in this case, syntactically, all the players have the same program. On the other hand, the execution of the program may be different for different players. When all players adopt the prescribed program then player i

will instruct the computer to perform feasible_i

, and its expected payoff will be according to feasible. This is implied by the syntactic comparison of the programs' content, as prescribed by the programs. If player i

deviates from the prescribed program, while the others stick to it, then the syntactic comparison will detect this deviation, and will cause each player j ≠ i

to call for the execution of minimax(i, j

). Given that feasible is individually rational, this will imply that a deviation is irrational.

Indeed, program equilibrium provides a strong theoretical foundation for understanding intents and achieving individually rational payoffs in one-shot games. However, there exist significant challenges in translating these ideas into practical implementations. In his talk [Realizing Intents with a Resource Model

](https://youtu.be/4Nh4EOpvKMY), @cwgoes highlights four key challenges:

- Termination/fix-point finding
- Function introspection
- Nominal/structural identities
- Unclear information flow

The resource model is introduced as a potential solution to addressing these challenges. In particular, the resource model:

- Separates computation and verification, avoiding the need to care about the specific path of execution.
- Resources encode constraints rather than imperative execution logic
- It can serve as a substrate for implementing information flow control policies.

Examples like solver selection, batching, and Threshold FHE (Aggregate Statistics) demonstrate how the resource model can provide a substrate for information flow control.

The talk concludes with three main takeaways:

- Intent-centric VM design is a separate problem from traditional von Neumann architectures.
- Speculative execution and atomicity are powerful tools for intent-centric systems.

- Some form of the resource model is likely inevitable for any intent-centric architecture.

Since then, the team working on the Anoma Resource Machine has written some math down. The curious reader is encouraged to engage with this concrete research.

- @degregat & @AHart published research on Intents Machines, entities capable of processing user intents and transforming system state accordingly.
- @vveiln & @cwgoes published a versioned resource machine specification,
- @cwgoes wrote about the Resource Machine it in relation to Ethereum.
- @vveiln published a new blog post as well, [Intents from the resource model perspective

](https://anoma.net/blog/intents-rm).

In lieu of our discussion, on intents and 50-year-old game theory, this section features Program Equilibrium and related literature.

Program equilibrium

By Moshe Tennenholtz

Abstract:

In a computerized setting, players' strategies can be implemented by computer programs, to be executed on a shared computational devise. This situation becomes typical to new Internet economies, where agent technologies play a major role. This allows the definition of a program equilibrium

. Following the fundamental ideas introduced by von Neumann in the 1940s (in parallel to his seminal contribution to game theory), a computer program can be used both as a set of instructions, as well as a file that can be read and compared with other files. We show that this idea implies that in a program equilibrium of the one-shot prisoners dilemma mutual cooperation is obtained. More generally, we show that the set of program equilibrium payoffs of a game coincides with the set of feasible and individually rational payoffs of it.

A commitment folk theorem

By: Adam Tauman Kalai, Ehud Kalai, Ehud Lehrer, Dov Samet

Abstract:

Real world players often increase their payoffs by voluntarily committing to play a fixed strategy, prior to the start of a strategic game. In fact, the players may further benefit from commitments that are conditional on the commitments of others. This paper proposes a model of conditional commitments that unifies earlier models while avoiding circularities that often arise in such models.

[

](https://ibb.co/Dz1QkG1)

A commitment folk theorem shows that the potential of voluntary conditional commitments is essentially unlimited. All feasible and individually rational payoffs of a two-person strategic game can be attained at the equilibria of one (universal) commitment game that uses simple commitment devices. The commitments are voluntary in the sense that each player maintains the option of playing the game without commitment, as originally defined.

Program equilibrium—a program reasoning approach

By Wiebe van der Hoek, Cees Witteveen, and Michael Wooldridge

The concept of program equilibrium, introduced by Howard (Theory and Decision 24(3):203–213, 1988) and further formalised by Tennenholtz (Game Econ Behav 49:363–373, 2004), represents one of the most ingenious and potentially far reaching applications of ideas from computer science in game theory to date. The basic idea is that a player in a game selects a strategy by entering a program, whose behaviour may be conditioned on the programs submitted by other players. Thus, for example, in the prisoner's dilemma, a player can enter a program that says "If his program is the same as mine, then I cooperate, otherwise I defect". It can easily be shown that if such programs are permitted, then rational cooperation is possible even in the one-shot prisoner's dilemma. In the original proposal of Tennenholtz, comparison between programs was limited to syntactic comparison of program texts.

[

](https://ibb.co/YfPXLFq)

While this approach has some considerable advantages (not the least being computational and semantic simplicity), it also has some important limitations. In this paper, we investigate an approach to program equilibrium in which richer conditions are allowed, based on model checking—one of the most successful approaches to reasoning about programs. We introduce a decision-tree model of strategies, which may be conditioned on strategies of others. We then formulate and investigate a notion of "outcome" for our setting, and investigate the complexity of reasoning about outcomes. We focus on coherent outcomes: outcomes in which every decision by every player is justified by the conditions in his program. We identify a condition under which there exist a unique coherent outcome. We also compare our notion of (coherent) outcome with that of (supported) semantics known from logic programming. We illustrate our approach with many examples.

Cooperation in the Prisoner's Dilemma

By: J.V. Howard

Discussion and Conclusion:

Biologists have suggested that individuals may cooperate in situations of potential conflict in two cases:

- if they recognise that the other player is a close relative, or
- if they recognise the other player as someone with whom they are playing a sequence of games.

[

](https://ibb.co/ng72g0v)

Both situations depend on recognising something about the other player. And a central assumption of much of Game Theory is that each player knows that the other players are rational (and they know that he knows, etc.). This assumption is not clearly specified unless "rational" has previously been defined. So we can regard Game Theory as attempting to provide a satisfactory definition of rational behaviour in game playing. The attempt is successful in the case of zero-sum games, but not for non-zero-sum games. The original problem must therefore be specified more clearly. We have tried to explore the use of Binmore's idea, replacing the assumption that players know each other to be rational by the assumption that they know each other's game algorithms. So here is a third way in which individuals may recognise each other. (We have shown that this does not necessarily mean that they can forecast what pure or mixed strategies the other players will use.)

In general a good algorithm (program) to use will depend on the opponents' programs. As in the Axelrod tournaments there may not be a best program for all populations of opponents. However, two points should be noted. Firstly, we are closer to the pure Game Theory problem of the single-shot game: in our tournaments there need be no memory of previous encounters. Secondly, we can define an ideal population for playing the Prisoner's Dilemma. If all players use the self-recognition program listed in the Appendix, and play cooperatively only if they recognise their opponents as their twins, then every game will be played cooperatively. Moreover no small group of invading programs could exploit the existing population. In fact, the invaders would do badly and die off after a few generations of competitive interaction. In this sense then, in the new formulation the Prisoner's Dilemma has a solution, and that solution is very reminiscent of the argument given by Rapoport and others.

P2P

[

](https://ibb.co/qDF33ng)

In this section, we begin by reviewing a recent whiteboard session from@cwgoes at the Modular summit where he breaks down Anoma, starting with the p2p Layer. Next, we feature a recent paper from the Science of Blockchain Conference 2024 Kadabra, a decentralized protocol for computing the routing table entries in Kademlia to accelerate lookups. Kadabra is motivated by the multi-armed bandit problem, and can automatically adapt to heterogeneity and dynamism in the network. The section concludes with a new work on blockchain economic denial of sustainability attacks.

Wtf is Anoma? - Modular Summit white board

By: @cwgoes

Transcript:

One thing Anoma tries to standardize areservice commitments. In Anoma, nodes are heterogeneous. Byheterogeneous we mean that these nodes have different preferences everything. They have different preferences about what messages they want to send to whom, and the types of messages they are interested in receiving. These nodes have different preferences about who they trust to provide different services to. Currently, Anoma conceptualizes services into four groups; ordering services, storage services, compute services, network relay/bandwidth services.

[

](https://ibb.co/2Z58v4F)

Services are things that one node does for another node. Think of how blockchains are used today. When sending a transaction to a blockchain, typically you get at least ordering and storage if you're using it as a DA layer, and you also often get compute if there is execution. By virtue of the P2P network, you are reliant on other nodes to relay your messages. You are using at least implicitly some network relay. All of these nodes have heterogenous preferences on what kinds of services they want and who they want them from.

Kadabra: Adapting Kademlia for the Decentralized Web

By: Yunqi Zhang and Shaileshh Bojja Venkatakrishnan

Abstract:

Blockchains have become the catalyst for a growing movement to create a more decentralized Internet. A fundamental operation of applications in a decentralized Internet is data storage and retrieval. As today's blockchains are limited in their storage functionalities, in recent years a number of peer-to-peer data storage networks have emerged based on the Kademlia distributed hash table protocol. However, existing Kademlia implementations are not efficient enough to support fast data storage and retrieval operations necessary for (decentralized) Web applications.

[

](https://ibb.co/6sC6jWm)

In this paper, we present Kadabra, a decentralized protocol for computing the routing table entries in Kademlia to accelerate lookups. Kadabra is motivated by the multi-armed bandit problem, and can automatically adapt to heterogeneity and dynamism in the network. Experimental results show Kadabra achieving between 15-50% lower lookup latencies compared to state-of-the-art baselines.

Swarm: Cost-Efficient Video Content Distribution with a Peer-to-Peer System

By: Dehui Wei, Jiao Zhang, Haozhe Li, Zhichen Xue, Yajie Peng,

Xiaofei Pang, Rui Han, Yan Ma, and Jialin Li

Abstract:

As ByteDance's business expands, the substantial infrastructure expenses associated with centralized Content Delivery Network (CDN) networks have rendered content distribution costs prohibitively high. In response, we embarked on exploring a peer-to-peer (P2P) network as a promising solution to alleviate the escalating costs of content distribution. However, the decentralized nature of P2P often introduces performance challenges, given the diversity and dispersion of peer devices. This study introduces Swarm, ByteDance's innovative hybrid system for video streaming. Swarm seamlessly integrates the robustness of a conventional CDN with the cost-efficiency of a decentralized P2P network. Its primary aim is to provide users with reliable streaming quality while minimizing traffic expenses. To achieve this, Swarm employs a centralized control plane comprised of a tracker cluster, overseeing a data plane with numerous edge residual resources. The tracker also takes on the responsibility of mapping clients to servers. Addressing the performance disparities among individual peer servers, Swarm utilizes our proprietary multipath parallel transmission method for communication between clients and peer servers. Operating stably for six years, Swarm now manages over a hundred thousand peer servers, serving nearly a hundred million users daily and saving the company hundreds of millions of RMB annually. Experimental results affirm that, while significantly cutting costs, Swarm performs on par with traditional CDNs.

Blockchain Economic Denial of Sustainability Attack: Exploiting Latency Optimization in Ethereum Transaction Forwarding

By: Taro Tsuchiya, Liyi Zhou, Kaihua Qin, Arthur Gervais, Nicolas Christin

Abstract:

Strategies related to the blockchain concept of Extractable Value (MEV/BEV), such as arbitrage, front- or back-running create an economic incentive for network nodes to reduce latency, including minimizing transaction validation time – a core feature to secure blockchain networks. A modified node, that neglects to filter invalid transactions in the Ethereum P2P network, introduces novel attack vectors. In this work, we formalize and evaluate a Blockchain Economic Denial of Sustainability (EDoS) attack, which can cause financial losses in traffic costs for operators of modified nodes.

[

](https://ibb.co/3m8ZqjS)

We 1) mathematically define the attack model, 2) identify thousands of empirical instances of this similar attack in the wild, 3) empirically measure the model parameters from our two monitoring nodes, and 4) conduct attack simulations on the local network to compare its performance with existing Denial-of-Service attacks. We show that an attacker can amplify network traffic at modified nodes by a factor of 3,600, and cause economic damages 13,800 times greater than the amount needed to carry out the attack. Despite these risks, aggressive latency reduction may still be profitable enough to justify the existence of modified nodes. To assess this trade-off, we 1) simulate the transaction validation process in the local network and 2) empirically measure the latency reduction by deploying our modified node in the Ethereum testnet. We conclude with a cost-benefit analysis of skipping validation and provide mitigation strategies against this attack.

Distributed Systems

[

](https://ibb.co/GnNWc0V)

The following section on distributed systems features five new papers (some of which were featured at SBC 2024), and one research workshop presentation. In particular, we feature Mysticeti and BRAID two protocols recently under consideration for implementation of multiple concurrent block proposers (MCP) on Ethereum - the motivation of which is to achieve censorship resistance.

The section rounds out with an excellent paper on Programmable Privacy, a systemization of knowledge. The paper surveys contemporary distributed blockchain protocols, with the aim of identifying cryptographic and design techniques which practically enable both expressive programmability and user data confidentiality. This section includes the following:

MYSTICETI: Reaching the Latency Limits with Uncertified DAGs

By: Kushal Babel, Andrey Chursin, George Danezis, Anastasios Kichidis, Lefteris Kokoris-Kogias,Arun Koshy, Alberto Sonnino, and Mingwei Tian

Abstract:

We introduce MYSTICETI-C, the first DAG-based Byzantine consensus protocol to achieve the lower bounds of latency of 3 message rounds. Since MYSTICETI-C is built over DAGs it also achieves high resource efficiency and censorship resistance. MYSTICETI-C achieves this latency improvement by avoiding explicit certification of the DAG blocks and by proposing a novel commit rule such that every block can be committed without delays, resulting in optimal latency in the steady state and under crash failures.

[

](https://ibb.co/3WH2xjx)

We further extend MYSTICETI-C to MYSTICETI-FPC, which incorporates a fast commit path that achieves even lower latency for transferring assets. Unlike prior fast commit path protocols, MYSTICETI-FPC minimizes the number of signatures and messages by weaving the fast path transactions into the DAG. This frees up resources, which subsequently result in better performance. We prove the safety and liveness in a Byzantine context. We evaluate both MYSTICETI protocols and compare them with state-of-the-art consensus and fast path protocols to demonstrate their low latency and resource efficiency, as well as their more graceful degradation under crash failures. MYSTICETI-C is the first Byzantine consensus protocol to achieve WAN latency of 0.5s for consensus commit while simultaneously maintaining state-of-the-art throughput of over 200k TPS. Finally, we report on integrating MYSTICETI-C as the consensus protocol into the Sui blockchain [67], resulting in over 4x latency reduction.

HotStuff-1: Linear Consensus with One-Phase Speculation

Abstract:

This paper introduces HotStuff-1, a BFT consensus protocol that improves the latency of HotStuff-2 by two network-hops while maintaining linear communication complexity against faults. Additionally, HotStuff-1 incorporates an incentive-compatible leader rotation regime that motivates leaders to commit consensus decisions promptly. HotStuff-1 achieves a reduction by two network hops by sending clients early finality confirmations speculatively, after one phase of the protocol. Unlike previous speculation regimes, the early finality confirmation path of HotStuff-1 is fault-tolerant and the latency improvement does not rely on optimism.

[

](https://ibb.co/sbnwSFJ)

An important consideration for speculation regimes in general, which is referred to as the prefix speculation dilemma, is exposed and resolved. HotStuff-1 embodies an additional mechanism, slotting, that thwarts real-world delays caused by rationally-incentivized leaders. Leaders may also be inclined to sabotage each other’s progress. The slotting mechanism allows leaders to drive multiple decisions, thus mitigating both threats, while dynamically adapting the number of allowed decisions per leader to network transmission delays.

Braid - Implementing Multiple Concurrent Block Proposers

By: Max Resnick

Basic ingredients:

1. Run k separate parallel chains using the same validator set.
2. All chains are synchronized (i.e. slot i is common to all chains)
3. The "block" in slot i

is the union of all transactions in slot i

of all k

chains.

1. After consensus on the union, execution/ state transition occurs using agreed upon deterministic rule, e.g.
2. Decreasing priority fees (Robinson and White)
3. Knapsack (maximize set of priority fees you can execute)
4. Execute As needed (to be released)

[

](https://ibb.co/q1cG4cv)

Some straightforward properties of BRAID:

1. Liveness of BRAID is inherited from the liveness of individual chains:
2. As long as each individual chain (or even some subset of the individual chains) makes progress, so does the union.
3. Eventual Consistency of BRAID follows from eventual Consistency of component chains:
4. Sketch: if each chain is consistent in t

periods, then so is union. Take $t \to \infty$

for eventual consistency.

The Economic Limits of Permissionless Consensus

By: Eric Budish, Andrew Lewis-Pye, Tim Roughgarden

Abstract:

The purpose of a consensus protocol is to keep a distributed network of nodes "in sync," even in the presence of an unpredictable communication network and adversarial behavior by some of the participating nodes. In the permissionless setting, these nodes may be operated by unknown players, with each player free to use multiple identifiers and to start or stop running the protocol at any time. Establishing that a permissionless consensus protocol is "secure" thus requires both a distributed computing argument (that the protocol guarantees consistency and liveness unless the fraction of adversarial participation is sufficiently large) and an economic argument (that carrying out an attack would be prohibitively expensive for an attacker). There is a mature toolbox for assembling arguments of the former type; the goal of this paper is to lay the foundations for arguments of the latter type.

[

](https://ibb.co/9NL9MNZ)

An ideal permissionless consensus protocol would, in addition to satisfying standard consistency and liveness guarantees, render consistency violations prohibitively expensive for the attacker without collateral damage to honest participants. We make this idea precise with our notion of the EAAC (expensive to attack in the absence of collapse) property, and prove the following results:

1. In the synchronous and dynamically available setting, with an adversary that controls at least one-half of the overall resources, no protocol can be EAAC.
2. In the partially synchronous and quasi-permissionless setting, with an adversary that controls at least one-third of the

overall resources, no protocol can be EAAC.

3. In the synchronous and quasi-permissionless setting, there is a proof-of-stake protocol that, provided the adversary controls less than two-thirds of the overall stake, satisfies the EAAC property.

All three results are optimal with respect to the size of the adversary.

Atomic and Fair Data Exchange via Blockchain

By: Ertem Nusret Tas, István András Seres, Yinuo Zhang, Márk Melczer, Mahimna Kelkar, Joseph Bonneau, Valeria Nikolaenko

Abstract:

We introduce a blockchain Fair Data Exchange (FDE) protocol, enabling a storage server to transfer a data file to a client atomically: the client receives the file if and only if the server receives an agreed-upon payment. We put forth a new definition for a cryptographic scheme that we name verifiable encryption under committed key (VECK), and we propose two instantiations for this scheme. Our protocol relies on a blockchain to enforce the atomicity of the exchange and uses VECK to ensure that the client receives the correct data (matching an agreed-upon commitment) before releasing the payment for the decrypting key.

[

](https://ibb.co/02Rxy8S)

Our protocol is trust-minimized and requires only constant-sized on-chain communication, concretely 3 signatures, 1 verification key, and 1 secret key, with most of the data stored and communicated off-chain. It also supports exchanging only a subset of the data, can amortize the server's work across multiple clients, and offers a general framework to design alternative FDE protocols using different commitment schemes. A prominent application of our protocol is the Danksharding data availability scheme on Ethereum, which commits to data via KZG polynomial commitments. We also provide an open-source implementation for our protocol with both instantiations for VECK, demonstrating our protocol's efficiency and practicality on Ethereum.

Programmable Privacy in Distributed Systems

By: Daniel Benarroch, Bryan Gillespie, Ying Tong Lai, and Andrew Miller

Abstract:

This Systematization of Knowledge conducts a survey of contemporary distributed blockchain protocols, with the aim of identifying cryptographic and design techniques which practically enable both expressive programmability and user data confidentiality. To facilitate a framing which supports the comparison of concretely very different protocols, we define an epoch-based computational model in the form of a flexible UC-style ideal functionality which divides the operation of privacy preserving networks into three phases: Independent, Mediated, and Global computation.

[

](https://ibb.co/6HVZ5mB)

Our analysis of protocols focuses in particular on features of the Mediated computation phase, which provides the facility to execute non-trivial program logic on private inputs from multiple users. Specifically, we compare implementations in different protocols for private limit order auctions, which we find to be a representative application which is common and relatively simple, but which exhibits adversarial dynamics which demonstrate the capabilities of a non-trivial Mediated computation mechanism. In our analysis, we identify four protocols representative of different high-level approaches used to implement Mediated computations. We compare protocols according to the degree and flexibility of programmability, the privacy properties achieved, and the security assumptions required for correct operation. We conclude by offering recommendations and best practices for future programmable privacy designs.

Restaking and Cryptoeconomic Safety

[

](https://ibb.co/HDx3P8m)

In our final section we feature a new work, How much should you pay for Restaking Security?

by Tarun Chitra and Melsh Pai.

In particular the paper's main resultsshow that:

- LRTs need to allocate intelligently for network-wide security
- AVSs need to choose a minimum yiled reward that avoids cascade risk
- Restaking risk combines
- PoS risk (e.g. combinatorial nature, rewards, attackers)
- DeFi risk (e.g. cascades)
- PoS risk (e.g. combinatorial nature, rewards, attackers)
- DeFi risk (e.g. cascades)

The second featured paper STAKESURE: Proof of Stake Mechanisms with Strong Cryptoeconomic Safety

, attempts to formalize a model for analyzing the cryptoeconomic safety of PoS blockchains, which separately analyzes the cost-of-corruption - the cost incurred by an attacker, and the profit-from-corruption - the profit gained by an attacker.

StakeSure transforms the cryptoeconomic landscape completely.

- Exactly measure how much cryptoeconomic security is sufficient, rather than have a random curve define it
- Self-scale the amount of security. If more more redistributable security is needed, the protocol users become willing to pay more
- Isolation of safety. If a certain user holds a certain amount of redistributable stake, then they do not need to model any other users who they may be sharing the platform with
- Compensation: Users of the platform will be compensated if something goes wrong, completing the system of “karma”.
- Universality: Any application can self-specify the amount of harm-from-corruption, so that many applications can share a common staking system.

How much should you pay for restaking security?

By: Tarun Chitra and Mellesh Pai

Abstract:

Restaking protocols have aggregated billions of dollars of security by utilizing token incentives and payments. A natural question to ask is: How much security do restaked services really need to purchase? To answer this question, we expand a model of Durvasula and Roughgarden [DR24] that includes incentives and an expanded threat model consisting of strategic attackers and users.

[

](https://ibb.co/zZtHfQh)

Our model shows that an adversary with a strictly submodular profit combined with strategic node operators who respond to incentives can avoid the large-scale cascading failures of [DR24]. We utilize our model to construct an approximation algorithm for choosing token-based incentives that achieve a given security level against adversaries who are bounded in the number of services they can simultaneously attack. Our results suggest that incentivized restaking protocols can be secure with proper incentive management.

Twitter Thread

STAKESURE: Proof of Stake Mechanisms with Strong Cryptoeconomic Safety

By: Soubhik Deb, Robert Raynor, Sreeram Kannan

Abstract:

As of July 15, 2023, Ethererum, which is a Proof-of-Stake (PoS) blockchain [1] has around 410 Billion USD in total assets on chain (popularly referred to as total-value-locked, TVL) but has only 33 Billion USD worth of ETH staked in securing the underlying consensus of the chain [2]. A preliminary analysis might suggest that as the amount staked is far less (11x less) than the value secured, the Ethereum blockchain is insecure and “over-leveraged” in a purely cryptoeconomic sense. In this work, we investigate how Ethereum, or, more generally, any PoS blockchain can be made secure despite this apparent imbalance. Towards that end, we attempt to formalize a model for analyzing the cryptoeconomic safety of PoS blockchain, which separately analyzes the cost-of-corruption, the cost incurred by an attacker, and the profit-from-corruption, the profit gained by an attacker.

[

](https://ibb.co/xSqXDMd)

We derive sharper bounds on profit-from-corruption, as well as new confirmation rules that significantly decrease this upper-bound. We evaluate cost-of-corruption and profit-from-corruption only from the perspective of attacking safety. Finally, we present a new “insurance” mechanism, STAKESURE, for allocating the slashed funds in a PoS system, that has several highly desirable properties: solving common information problem in existing blockchains, creating a mechanism for provably safe bridging, and providing the first sharp solution for automatically adjusting how much economic security is sufficient in a PoS system. Finally, we show that the system satisfies a notion of strong cryptoeconomic safety, which guarantees that no honest transactor ever loses money, and creates a closed system of Karma, which not only ensures that the attacker suffers a loss of funds but also that the harmed parties are sufficiently compensated.

Feedback

Thanks for reading. If you found some of the material interesting and want to chat with us about it, connect with us at:

- research.anoma.net

If you are interested in collaborating on an Anoma Research Topic (ART), like the one’s featured in this newsletter, we’d be more than happy to collaborate. As a refresher, what is an ART?

- ART is an open-access, lightweight, peer-reviewed index of reports, primarily written and reviewed by Anoma’s researchers and engineers, but open to anyone who wishes to participate

.

- The ART collection covers a diverse set of topics including distributed systems, cryptography, compilers and blockchains. These reports help map the theoretical foundations upon which Anoma is built.

Visit art.anoma.net for more details.

Content Submissions for future newsletters

Given that we intend to publish the newsletter on a monthly cadence, we are open to and encourage readers to submit content. Please submit content for consideration for future issues to intentnewsletter@proton.me

E-mail Signup

If you’d like this monthly newsletter delivered to your inboxsign-up here!