Smart Contracts and Rationality

Nikolaj Ignatieff Schwartzbach

PhD Dissertation

Department of Computer Science
Aarhus University
Denmark

Smart Contracts and Rationality

A Dissertation
Presented to the Faculty of Natural Sciences
of Aarhus University
in Partial Fulfillment of the Requirements
for the PhD Degree

by
Nikolaj Ignatieff Schwartzbach
July 31, 2023

Abstract

Conventionally, transactions of goods over the internet is performed by the use of a trusted intermediary that holds the payment in escrow and settles disputes. They are assumed to behave honestly because of reputation effects; if they misbehave, its users will move to another market. This effect underlies the success of various commerce platforms that have seen a surge in profits over the past decades. However, these systems inherently constitute a privacy risk, in addition to other problems resulting from their market dominance.

In this thesis, we propose a system for fully decentralized commerce involving rational agents that interact using a blockchain. At its core, the system consists of an escrow mechanism that enables both the buyer and the seller to wager money to threaten to invoke a dispute resolution system that determines who were honest. Intuitively, an agent will wager money to settle the dispute only if they think they will win the dispute. This deters an agent from misbehaving, as they can infer that the other agent would threaten them with invoking the dispute resolution system. The dispute resolution system is in turn implemented as a decentralized jury system. Here, the main challenge is to ensure the agents exert an effort to assess the evidence and vote in favor of the honest agent. In a decentralized and anonymous setting, there is no way to hold an agent accountable to their vote, and worse yet, the true state of the dispute is unobservable to the mechanism. We analyze a class of mechanisms that use the wager from the losing agent to compensate only those jurors that made the majority decision. We show that the mechanism is likely to produce good adjudication outcomes under reasonable assumptions. While variations of these ideas are already proposed and deployed in practice, to the best of our knowledge, they lack a thorough and rigorous analysis, making it unclear under which conditions these markets can be assumed secure and/or comply with laws and regulations. By contrast, our system is secure under rather minimal assumptions. Importantly, our system can be combined with recent advances in cryptographic identity management to strike a reasonable balance between privacy and compliance with laws and regulations.

Our work can be considered a step in formalizing decentralized commerce systems, and provides a number of tools and models to analyze incentives in smart contracts — we believe that more foundational work needs to be done before these systems can be deployed and used in practice.

Resumé

Traditionelt foretages handel af varer over internettet ved hjælp af en betroet tredjepart, der holder betalingen i depot og mægler mellem køber og sælger. Tredjeparten formodes at opføre sig ærligt grundet dets omdømme: hvis den træffer dårlige beslutninger, vil dens kunder flytte til et andet market og aktierne tabe værdi. Denne effekt underbygger den enorme success af diverse handelsplatforme, der har set en eksplosiv vækst de sidste årtier. Desværre udgør disse systemer en privatlivsrisiko udover andre problemer forårsaget af deres dominante markedsposition.

I denne afhandling foreslår vi et system til fuldt decentraliseret handel mellem rationelle agenter, der interagerer ved hjælp af en blockchain. Systemet består grundlæggende af en depotmekanisme, der tillader både køber og sælger at satse penge for at true med at anvende en mægler. Intuitivt vil agenterne kun satse penge, hvis de regner med at vinde mæglingen. Dette afholder en agent fra at opføre sig uærligt, da de kan slutte, at den anden agent i givet fald vil true dem med mægling. Mekanismen til mægling er da implementeret som et decentraliseret jurysystem. Her er den største udfordring at få jurymedlemmerne til faktisk at kigge på bevismaterialet. I en decentraliseret og anonym verden kan jurymedlemmerne ikke holdes ansvarlige for deres stemmer, og værre endnu er den faktisk tilstand af disputsen ikke observerbar for mekanismen. Vi analyserer en klasse af mekanismer, der benytter de penge, der blev satset af den tabende part, til at betale de jurymedlemmer, der foretog den afgørende beslutning. Vi finder, at denne mekanisme resulterer i en god mægling under rimelige antagelser. Selvom variationer af dette system er allerede er implementeret i praksis, mangler disse efter vores overbevisning en grundig spilteoretisk analyse, hvilket gør det uklart under hvilke antagelser disse systemer kan formodes at være sikre. Derimod er vores system beviseligt sikkert under forholdsvis minimale antagelser. Ydermere, kan vores system kombineres med nye udviklinger inden for kryptografisk identitetshåndtering til at finde en fornuftig balance mellem privatliv og overholdelse af lovgivning.

Vores arbejde kan betragtes som et skridt mod at formalisere decentraliserede handelssystemer og giver en række værktøjer til at analysere incitamenter i smart contracts – vi mener, at mere grundlæggende arbejde er nødvendigt før disse systemer kan anvendes i praksis.

Acknowledgments

I thank my mom Kirsten Ann Larsen, my brother Jonatan Ignatieff Schwartzbach, the rest of my family and all of my friends for supporting me in this endeavour and taking an interest in my research. I reluctantly apologise to them for the occasional rants they have had to endure. I am especially thankful of my girlfriend, Tiziana Svenningsen, and the relationship we have built through the years. Meeting her is the best thing to happen to me.

I am grateful to my advisor Ivan Damgård for taking me under his wing and accepting me as a PhD student. Ivan introduced me to the world of research, to cryptography, and especially to secure computation — he is a great teacher and an inspiration to me. I especially thank him for entrusting me the freedom to pursue my own research interests, most of which arguably did not involve any cryptography. I would never have expected to have this level of agency over my research. Ivan always supported me and was willing to listen to my ideas and provide valuable feedback and guidance, despite my procrastination on the cryptography related projects he suggested I do.

I feel humbled to have been a part of the Aarhus Crypto Group for these past five years. I have met many wonderful and smart people in the group who have taught me about cryptography and beyond, including Damiano Abram, Diego F. Aranha, Carsten Baum, Lennart Braun, Jakob Burkhardt, Matteo Campanelli, Bernardo David, Daniel Escudero, Mathias Hall-Andersen, Adam Blatchley Hansen, Benjamin Salling Hvass, Kelsey Melissaris, Nikolas Melissaris, Jesper Buus Nielsen, Michael Nielsen, Maciej Obremski, Sabine Oechsner, Claudio Orlandi, Mahak Pancholi, Rahul Rachuri, Divya Ravi, Peter Scholl, Mark Simkin, Bas Spitters, Akira Takahashi, and Sophia Yakoubov — an exhaustive list would be quite too long to mention, I apologise for any omissions. Thank you for all the conferences, workshops, reading groups, cakes, bike rides, and beers. Special thanks to Claudio Orlandi for being helpful in answering questions concerning the PhD. Also thanks to Malene B. B. Andersen for always being helpful with administrative matters.

I thank Alon Rosen for hosting me at Bocconi University for the first six months of 2022. Alon introduced me to foundational cryptography and statistical inference — I had a lot of fun working on total search problems with him and all of his visitors. Alon is a great host and I look forward to coming back to work more with him. Thank you to STIBO fonden for

sponsoring my visit. Thanks to Alon I met many fantastic people during my visit to Milan, and I am thankful for the friends I made. This includes Romain Bourneuf, Andrea Celli, Lukáš Folwarczný, Pavel Hubáček, Daji Landis, Daniel Mitropolsky, and Prashant Nalini Vasudevan. My visit to Milan would not have been the same without all of you. Thank you to Pavel Hubáček for inviting me to Czech Republic and for making me an honorary Pavel.

I thank Ioannis Caragiannis for introducing me to the world of social choice theory and for entertaining my crude ideas for adjudication with strategic jurors. I was glad to see Ioannis take a genuine interest in my ideas, and I am pleased with the work we did together. Thank you to Matteo Campanelli, Luca Nizzardo, Irene Giacomelli, and William George for helpful discussions in various stages of the process. I also thank Ioannis for encouraging me to to submit my paper on payment schemes to EC'22 — visiting Colorado was a great experience that gave me a lot of confidence.

I thank Prashant Nalini Vasudevan for hosting me at National University of Singapore in the fall of 2022. I had a great time with him and Sagnik Saha working on the planted k-SUM problem — I learned a lot about fine-grained complexity and concentration bounds from my visit with Prashant. Thank you to Divesh Aggarwal, Eldon Chung, Maciej Obremski, Li Zeyong, and the rest of their group from Centre for Quantum Technologies for teaching me about extractors and keeping me company during my visit.

Thank you to all the amazing professors at Aarhus University who have taught me and shaped my understanding of computer science, including Aslan Askarov, Ira Assent, Lars Birkedal, Gerth Stølting Brodal, Henrik Bærbak Christensen, Olivier Danvy, Kristoffer Arnsfelt Hansen, Kurt Jensen, Kasper Green Larsen, and Anders Møller. Thanks to Kurt Jensen for letting me contribute to the teaching of Introduction to Programming, and to Anders Møller and also to Gudmund Skovbjerg Frandsen for helping me with various practical matters concerning the PhD.

A big thanks to all my co-authors: Romain Bourneuf, Ioannis Caragiannis, Ivan Damgård, Lukáš Folwarczný, Mathias Hall-Andersen, Pavel Hubáček, Daji Landis, Boyang Li, Alon Rosen, Sagnik Saha, and Prashant Vasudevan. I am proud of all the work we have done together.

Finally, I dedicate this thesis to my father,

Michael Ignatieff Schwartzbach.

His curiosity and enthusiasm for computer science was the driving force behind this undertaking.

Nikolaj Ignatieff Schwartzbach,
Aarhus, Denmark
July 31, 2023

Til min far

Contents

Abstract **i**

Resumé **iii**

Acknowledgments **v**

Contents **ix**

0 Prologue **1**

1 Introduction **3**
1.1 Dispute Resolution Systems 6
1.2 Blockchains and Smart Contracts 10
1.3 A Framework for Designing Secure Contracts 14
1.4 Smart Contracts and Commitments to Strategies 17
1.5 List of Publications 22

2 *Homo Economicus* **25**
2.1 The Nash Equilibrium 28
2.2 Representations of Games 31
2.3 Subgame Perfection 34
2.4 Multi-Lateral Deviations 37

3 Commerce **39**
3.1 The Basic Contract 42
3.2 The Generalized Contract 47
3.3 Practical Considerations 50

4 Adjudication **55**
4.1 Modeling Assumptions 58
4.2 Equilibrium Analysis 59
4.3 Selecting Payments for Correct Adjudication 63
4.4 Computational Experiments 66

5 Payments **73**

5.1 Payment Schemes 77
5.2 Computational Complexity 83
5.3 Case Study: Secure Rational MPC from PVC 88
5.4 A Lower Bound on the Size of Payments 92

6 Commitments **95**
6.1 Contracts as Stackelberg Equilibria 97
6.2 Imperfect Information, One Contract, NP-completeness 101
6.3 Imperfect Information, k Contracts, Σ_k^{P}-hardness 103
6.4 Perfect Information, Two Contracts, Upper Bound 104
6.5 Perfect Information, Unbounded Contracts, PSPACE-hardness . 106

7 Threats **109**
7.1 Stackelberg Resilience 110
7.2 Downward Transitivity 112
7.3 Decentralized Commerce 118
7.4 Auctions and Transaction Fee Mechanisms 121

Bibliography **137**

Prologue

A CAR HONKS and swerves to avoid a pedestrian. The sound of screeching tires is faintly heard through the window that faces the street. Alice has been awake for some time. Her room is dimly lit as the sun starts to peer through the shutters. She knows it is too early to get up so she lies thinking. Today is the day where she will finally get her money back. The jury had two days to reach a decision that will be finalized by noon today. Her evidence was indisputable, she reckoned, so surely they will vote in her favor. And to think what could have happened if she did not act as quickly as she did. What happened was the following: the smell prompted her to pay attention, as the toaster starting emitting a foul odor of burnt plastic. She turned her head and found that the machine was producing smoke and bright sparks. She felt a sharp rush of adrenaline and reached to unplug the device. There was no immediate sign of fire but she stood frozen and inspected the machine without blinking for twenty-two seconds. She had used the toaster only a handful of times previously, so it was clearly faulty from production. She grabbed her phone and photographed the machine from three different angles. The plastic cover had visibly melted on one of its sides and it was still producing a bit of smoke. She pondered what to do with the machine and placed it under the fume hood that she turned on at the highest setting. She went to sit at her desk and turned on her desktop computer. She opened up the contract for the toaster and clicked the button that raises a dispute. The toaster was not even that expensive – 39,95€ + 8€ shipping – but she felt compelled to seek redemption for almost being set on fire. She hastily wrote a description of what had happened and attached the photos that she took on her phone. She knew that raising a dispute requires her to submit a deposit to prove she is serious. She felt convinced her evidence was unequivocal, so she transferred the 47,95€ required to raise a dispute. The seller now has twenty-four hours to respond to the dispute. Surely, they would forfeit and Alice would get back her money. Several hours went by with Alice constantly checking her phone to see if she got her money back yet. Finally, she received a notification on her phone that the seller had responded to her dispute. “Dispute was Countered”. She was taken aback and stared at her screen in disbelief. She refreshed the page but nothing had changed; her dispute had, in fact, been countered. The seller insisted that the photos she

uploaded were not genuine and suggested that maybe they had been generated. Like her, the seller had also transferred a deposit of 47,95€. Alice rolls over on her side and reaches for her phone on the nightstand. It is barely 6am but she is wide awake and decides to start her day.

Bob peeks at his wrist watch and picks up the pace as he rushes towards the bus stop. It is 11:23am. The light turns green and bus no. 87 starts driving. It pulls over at the bus stop and opens its doors. Bob makes it inside and finds a seat near the entrance. He sits down and lets out a sigh. His heart is racing but now he can relax. He pulls out his phone and notices a notification: "Judgment Pending - *Anonymous v. WeToast*". Oh no! He had forgotten about the case and now there is only half an hour left to make a verdict. He clicks the notification and opens up the contract. A user had purchased a toaster from the company WeToast and claims it caught fire after using it only a few times. They had attached three photos of the toaster with the cover partially melted. The seller claims the photos are not real and were generated as they are not accompanied by a proof of authenticity. Bob looks at the photos. The lighting is indeed a bit strange and something seems off. The traffic light turns red and the bus comes to a halt. Bob has to get off at the next stop. He looks once more at the photos and decides to vote in favor in WeToast. He gets up from his seat, exits the bus and jogs towards the lecture hall.

As the clock turns 12:00pm, an automated series of events transpires. First, the votes of the jurors are tallied. A total of twelve votes were cast, eight of which are in favor of Alice and four of which are in favor of WeToast. As the majority decision rules in favor of Alice, she is declared the winner and is repaid the 47,95€ she had paid for the toaster, as well as the 47,95€ she had staked. WeToast is repaid nothing and lose their wager of 47,95€ as well as their marginal cost of having produced and shipped the toaster. Second, the jurors that voted in favor of Alice share a reward, while the jurors that voted in favor of WeToast share a penalty. The case had a total reward of 79,95€, so each juror had staked 79,95€ - 47,95€ = 32€. This reward is shared among the eight jurors in the majority who are each repaid 32€ + 79,95€ / 8 = 42€, yielding a profit of 10€. The remaining four jurors – including Bob – share the penalty of 32€, and are thus repaid 24€, causing each of them to lose 8€. Alice is thrilled to see that all her money was repaid. She may have spent some time dealing with the dispute, but at least she did not have to pay for the faulty toaster. Bob wishes that he had spent more time deliberating before placing his vote.

Chapter 1

Introduction

> "*However beautiful the strategy, you should occasionally look at the results.*"
>
> Winston Churchill

The story of Alice and WeToast describes a system that enables any two agents to exchange goods and services for money: Alice purchased a toaster online from the company WeToast, who for whatever reason delivered to her a faulty machine. Fortunately, she raised a dispute and was able to get her money back. The system is designed to disincentivize malicious behavior, in the sense that an agent should expect to lose money by attempting to cheat. Indeed, the behavior of WeToast seems strange: we cannot know if WeToast delivered a faulty toaster out of negligence or out of malice. It can be assumed that the seller knows the condition of the item they are selling. Thus, WeToast would have known that sending the faulty toaster would result in Alice raising a dispute. In this case, they should have been reluctant to counter the dispute, knowing that doing so requires them to stake money that they would probably lose. Knowing they would not want to counter the dispute, we can infer they would also not want to send the faulty item — they can infer that the recipient would raise a dispute and probably win that dispute. This is an unfavorable outcome for WeToast. By contrast, if the payment is large enough, they would get a net profit by instead sending the good item, as the buyer would presumably accept delivery of the item, as also they might also be faced with a dispute they would probably lose, if they were to raise a dispute. We might say that WeToast behaved *irrationally*. Our goal is to ensure that cheating is always irrational.

The system we just described is an alternative to centralized marketplaces such as Amazon, eBay, Alibaba, or Etsy. Of course, with centralized marketplaces being so well-established, this begs the question of why we should bother trying to replace them. After all, these systems work well most of the time [105]. Indeed, a decentralized alternative immediately raises a number

of practical and ethical concerns. First and foremost, we might ask if the decentralized marketplace is secure — are there any reasonable assumptions under which we can *prove* that the system is secure? Secondly, if these systems are, in fact, fully decentralized and potentially anonymous, to what extent can they be expected to comply with laws and regulations?

In this thesis, we explore designing such a decentralized marketplace and give an affirmative answer to the first of these questions. The second question can be answered by choosing an appropriate blockchain: if we use a blockchain with *revocable anonymity* [50], such as the one proposed by Damgård *et al.* [76, 79], agents are in principle anonymous but can be de-anonymized under cooperation of the appropriate authorities, say if illegal behavior is suspected. Our results hold in a rather minimal model where the agents are rational, care about money, and have shared access to a blockchain. Along the way, we develop several new models and methods for reasoning about smart contracts in these highly strategic environments. Our work can be considered a step in formalizing decentralized commerce systems, and provides some tools to analyze smart contracts from a game-theoretic perspective — we believe there is still much foundational work to be done before these systems can be deployed and used in practice.

Why Decentralization? The key feature of our proposed system is that here, all processing of data and funds can be done in a fully decentralized manner, obviating the need to trust any single agent[1]. By contrast, the security of a centralized marketplace inherently relies on users being able to trust these services: if such a marketplace is controlled by an adversary, no guarantees can be made on its security. In particular, such marketplaces have inherent privacy concerns, in that centralized marketplaces may collect data on their consumers to use e.g. for targeted advertising [139]. There are also other potential problems: such a centralized marketplace may have an incentive to engage in monopolistic behavior, such as removal of competitors' products or differential pricing based on customer demographics [134]. Since the 2010s, various markets were deployed online that allow users to transact with more anonymity, so-called *darknet markets* [21, 163]. A darknet market is a centralized market that runs on a mix network (such as Tor [86]) and uses cryptocurrency [18, 257] for payments that the market holds in escrow until the trade has been completed. This mechanism has proved to be remarkably efficient, evidenced by the enormous market caps of some of the larger markets [185]. Unfortunately, the vast majority of this volume is related to criminal activities, such as the sale of drugs, weapons, and counterfeit passports [46] — some trade-off needs to be carefully struck between user privacy and compliance with laws and regulations.

[1]Specifically, we show that both the escrow contract and the jury system can be implemented as automated programs (smart contracts) that run on a 'world computer' (a blockchain), for which we *prove* it is secure in a model where agents are rational — we make these notions more precise in Section 1.2.

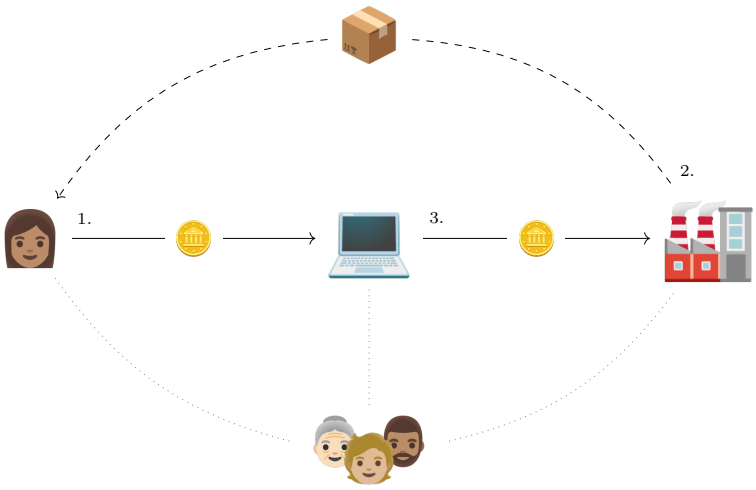

Figure 1.1: High-level depiction of the main escrow contract that allows the company WeToast to send a package (dashed lines, top) to Alice in a way that disincentivizes cheating. The workflow of the contract is as follows. 1. Alice transfers the money to the contract. 2. WeToast sends the toaster. 3. Alice can either accept or reject delivery of the item. If she accepts, the money is transferred to WeToast and the trade is finalized. The dotted lines to the jurors (bottom) indicates that they are only invoked in case either Alice or WeToast raises a dispute.

By design, darknet markets also suffer from other problems. Indeed, darknet markets are infamous for exit scamming [138], where the market is suddenly closed, with the operators stealing all the funds that were held in escrow. Such an attack may yield profits in the millions of dollars. It is often difficult, if not impossible, to find the perpetrators and hold them accountable for their actions [89].

One solution that solves both of these problems involves combining a fully decentralized marketplace with recent advances in cryptographic identity management [50] that we will detail further in Section 1.2.

Decentralized Escrow. The first contribution of this thesis is to propose a mechanism for decentralized escrow (Chapter 3). It works as follows: a seller broadcasts an advertisement to a set of potential buyers. A buyer then enters into a contract with the seller to agree on the terms of the transaction, such as the delivery method or other details, e.g. if the item is customizable. Both agents sign the contract and the escrow game start (see Fig. 1.1 for an illustration). The buyer transfers the payment to the contract which is then held in escrow. The seller then completes their end of the transaction and notifies the contract that the work was completed. The buyer then either accepts completion of the transaction, in which case the payment is transferred

to the seller and the contract is terminated, or the buyer raises a dispute, claiming the transaction was not completed as agreed. To disincentivize frivolous disputes, raising a dispute requires the buyer to submit a stake to the contract of roughly the same amount as the payment itself. The seller can then either forfeit and accept defeat, in which case all the money is returned to the buyer and the contract is terminated, or they can counter the dispute by also submitting a stake to the contract. In this case, we invoke an external *dispute resolution system* to determine who were the honest agent (in a black-box way, we analyze such a system in Chapter 4). We then return the payment and the stake to that agent. The losing agent's stake is used to pay for resolving the dispute and the contract is terminated. We show that this contract is indeed secure in a model where the agents are rational and the dispute resolution system is biased in favor of the honest agent. In this case, the seller always completes their end of the transaction, and the buyer always accepts delivery — the dispute resolution system is never invoked and is only needed as a 'threat' to disincentivize dishonest behavior. We will return in more generality to these threats later in this thesis (Chapters 6 and 7).

It is an important feature of this escrow that the dispute resolution is invoked *optimistically*, i.e. only when necessary. An alternative implementation invokes the dispute resolution system for every trade. This is undesirable for two reasons: first of all, resolving a dispute presumably incurs some cost, and secondly, unless the dispute resolution system is perfect and has zero percent error, there will be instances in which both agents acted honestly, and the system rules in an agent's disfavor. Invoking the dispute resolution system only optimistically arguably solves both of these problems.

1.1 Dispute Resolution Systems

Security of the escrow contract inherently relies on being able to trust the dispute resolution system: it is not clear if it is possible to design a dispute resolution system that is biased in favor of the honest agent. When the content of the transaction is a digital item (such as an e-book), there are dispute resolution protocols that work under computational assumptions [96, 97, 128]. However, these protocols inherently rely on being able to encode the item in question as binary strings, and thus do not meaningfully generalize to the case where the item is physical (such as a toaster). The most natural solution is to designate a third agent to act as the arbiter. This approach is used in practice in systems such as OpenBazaar [8] and ArbStore [7]. In both cases, security is argued using a reputation system: presumably, the arbiter cares about their reputation, giving them some incentive to vote impartially (in ArbStore, the arbiters' identities are public). Such systems are supported theoretically e.g. by Dellarocas [85] who finds that under the right conditions, a long-lived arbiter has an incentive to behave honestly when faced with many disputes. The

incentive is strongest in the initial phase where the arbiter has to work hard to build up a good reputation and diminishes as their reputation increases, ultimately disappearing altogether when the arbiter nears the 'end of its life'. Several systems have been proposed that formalize the notion of reputation, notably the beta reputation system [140] for e-commerce, and the EigenTrust algorithm [148] for peer-to-peer systems. Recently, reputation systems have also been explicitly designed for blockchain-based e-commerce systems [267]. However, it is not clear that all arbiters care about their reputation when they are anonymous and cannot easily be held accountable for their judgments. In any case, relying on reputation is undesirable when the goal is to decentralize, as again, these systems inherently have a single point of failure. Ideally, we want to distribute the trust across a wider set of agents to make it more plausible that the system works as intended.

When the moderator solution does not work, the next best solution involves using a jury system where we appoint a set of jurors to determine which agent were honest. We then ask the buyer and the seller to provide evidence that we forward to the jurors, ask them for their vote, and take the majority decision. Variants of this setup are well-studied in voting theory [55, 70, 262], and social choice theory [9, 43], in what is known as truthful elicitation. Here, quadratic scoring rules [116], Bayesian truth serum [206] and peer-prediction methods [189] find applications in various different contexts [101]. However, truthful elicitation methods do not seem appropriate for our use-case, as we fundamentally do not care about whether the jurors vote truthfully. Instead, we care only about whether or not the outcome is biased towards the ground truth: ideally for us, a juror with the 'wrong' opinion would vote against their belief to have a higher probability of recovering the ground truth. The fundamental problem with a jury-based approach is that the jurors are anonymous and cannot be held accountable for how they voted, since the 'true' outcome is inherently unobservable to the blockchain. Presumably, assessing the evidence requires some effort by the jurors which they would rather not expend unless necessary, e.g. unless compelled by morality or persuaded with payments. Indeed, there are various models in the literature [111, 112, 187, 198] studying the relation between the effort exerted and the preferences of the agents. However, this line of work is not immediately applicable to our setting where the jurors might not care about the outcome. Generally, a mechanism that forwards content onto a blockchain is known as a *blockchain oracle* [49]. Many solutions are proposed in the literature, some of which are also deployed in practice, including Town Crier [263], Astraea [3], ChainLink [44], and Infochain [117]. A common feature of these systems is that they lack a thorough game-theoretic analysis or rely on rather idealized models where e.g. there is a trusted agent, the ground truth is eventually observable, or all the jurors are biased towards the ground truth (and so, correctness of the adjudication follows directly from an appropriate concentration bound [137]).

In our setting with anonymous jurors, we will explicitly assume that the

jurors are *morally agnostic* and thus do not care whether or not they collectively reach the correct outcome. For such jurors, the best-response is to simply vote randomly without assessing the case evidence, as to do so would require an effort they would rather not expend. This results in a random coin flip which does not conform to the requirement needed for the escrow contract to work. Instead, we have to use payments to incentivize the jurors to properly assess the case evidence. To avoid the same problem, these payments should be somehow conditioned on the votes of the agents. One natural proposal which is used in practice by systems such as Kleros [169, 170] and Augur [201] is to reward jurors for voting in accordance with the final verdict[2]. The hope is that if the payments are set correctly, the jurors are incentivized to exert an effort to receive the payment, in such a way that the jurors collectively reach the correct verdict. Lesaege, Ast and George [169] argue that such a mechanism results in good outcomes using focal points [219]: the jurors expect the other jurors to vote honestly, so they will do so themselves to obtain the payment. In general, the incentives of these systems are only sparsely studied: Lesaege, George, and Ast [170] propose a payment function for which they show truthfulness is a weakly dominating strategy, in a model where the jurors expect 1) the other jurors to vote independently of themselves, and that 2) the outcome is independent of the votes of the jurors. We will later show that, in general, the full strategic behavior is more complicated.

Adjudication Games. The second contribution of this thesis is to formally study a model of the adjudication game that we just described (Chapter 4). This section is adapted from the introduction of [54]. For simplicity, we restrict our attention to binary disputes that, in particular, may be used in the aforementioned escrow mechanism. We then consider majority voting and consider payment functions that reward those jurors that made the majority decision and (optionally) punish those jurors that were in the minority (see Fig. 1.2 for an illustration). We assume jurors have access to a history of similar disputes in the past that they can use to correlate the outcomes with their own understanding. The role of payments is to amplify an agent's incentive to take these correlations into account when producing their vote: if there is no correlation, the agent will vote randomly. If instead, the correlation is positive, they will cast their opinion as their vote, while if the correlation is negative, they will vote opposite to their opinion. It is important to remark that our work deviates from the traditional literature on voting theory [55, 70, 262], as we do not care about truthfulness as long as the outcome of the adjudication is mostly correct. We analyze the equilibria of the resulting strategic game under standard assumptions on the utilities of the jurors (risk-neutrality and

[2]The full system Kleros is significantly more complicated: it has measures to mitigate Sybil attacks, as well as subcourts and appeal mechanisms. In this work, we focus only on the strategic behavior in the adjudication mechanism itself.

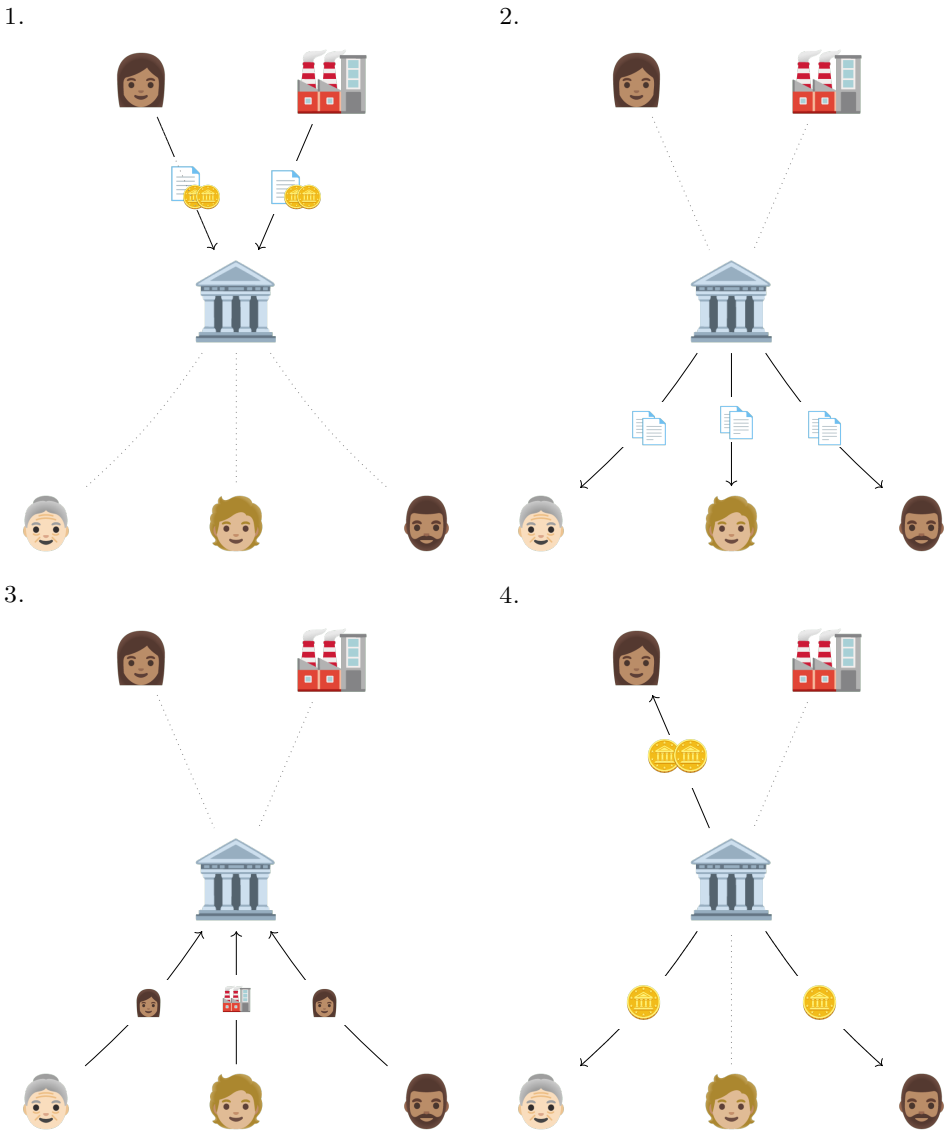

Figure 1.2: Illustration of the jury system. 1. Alice and WeToast both submit evidence of their honesty to the court system, as well as a deposit of two coins. 2. The court system forwards the evidence to the jurors. 3. All jurors report back their vote to the court contract. 4. Alice had the majority vote, so she receives back her deposit, and the leftover money is used to compensate those jurors that voted for her.

quasi-linearity). We give sufficient conditions on the payments so that the equilibria are simple, in the sense that all agents either vote randomly or are all biased towards the same outcome. For these equilibria, it is easy to bound the accuracy by application of e.g. a Hoeffding bound [137]. We show that these conditions are satisfied e.g. by the simple 'threshold payment function' that simply gives a constant reward to each juror in the majority. The conditions are also satisfied by a simplified version of the payment function used by Kleros [169, 170], the award/loss sharing function, where the minority shares a fixed cost which is then split as a reward among those jurors that voted in the majority. We show how to find minimal payments that satisfy these conditions for a simple model of the jurors using linear programming. We show that for this class of payment functions, there are three different equilibria: a 'trivial' equilibrium where all agents exert zero effort and vote randomly, and two symmetric equilibria: a 'good' equilibrium where no agent is biased towards the bad outcome, and a 'bad' equilibrium where no agent is biased towards the good outcome. Finally, we perform computational experiments to justify that jurors in practice tend to reach the good equilibrium with high probability, assuming the jurors are, on average, well-informed.

Our work suggests that it is indeed possible to construct a dispute resolution mechanism that is biased in favor of honest agents, which weakens the assumptions needed to construct the decentralized marketplace that we propose in this thesis. The model may also be applicable to other scenarios, e.g. in peer reviewing.

1.2 Blockchains and Smart Contracts

Throughout this thesis, we will consider a model in which the agents have shared access to a blockchain that allows them to deploy smart contracts. We will now give a simplified description of what we mean by this: at a high level, a blockchain is a decentralized ledger that stores data in a totally ordered manner [18]. The ledger is represented by a chain of blocks of data (see Fig. 1.3 for an illustration), hence the name. This chain is replicated across many nodes around the globe. To enter the network, a new device queries the network for the latest copy of the chain: they then download the entire blockchain and verify its authenticity on their own machine. The idea is that knowing the latest block in the chain allows for determining the balance of each account by computing backwards to the first block (the genesis block). This is used to prevent double-spending, though this verification stepmay be expensive. For Bitcoin as of 2023, the entire ledger takes up nearly 500 GB and takes up to five days to verify on an ordinary computer. While this is small compared to industrial databases, this data has to be stored on every device that wants to make transactions. There are ways to avoid having to go back to the genesis block, e.g. using a finality layer [48]: here, the network periodically runs

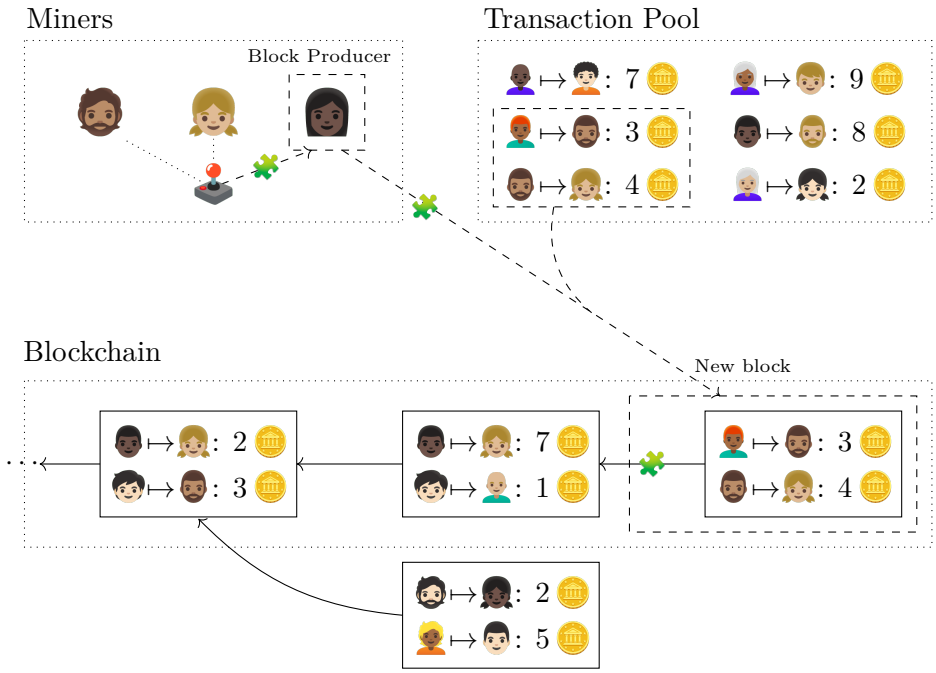

Figure 1.3: High-level depiction of the function of a blockchain. In the bottom, we see the blockchain which consists of a chain of blocks, each of which contains multiple transactions. Knowing the latest block in the chain allows for determining the balances of each account. When there are multiple competing chains, the longest chain is regarded as the current one (in this case, the top-most chain). The chain is extended by a set of miners (top-left) who attempt to solve a puzzle. When a miner solves the puzzle, they are designated as a *block producer* and are allowed to use the solution of the puzzle to extend the chain using some number of transactions from the transaction pool.

a consensus protocol on the state of the chain, to avoid having to redo the entire computation [87, 266]. Other solutions involve incremental verifiable computation [246] where the blocks propagate a succinct proof that they were generated from the genesis block.

Extending the Chain. The chain is periodically extended by designating an agent — the block producer — to choose some data to include in the next block. The block producer is chosen using a lottery for which the distribution of ballots is determined by some scarce resource that can be reasonably assumed to be majority controlled by honest agents. The prototypical scarce resource is that of *computation*, where the block producer is chosen by having agents solve a hard cryptographic puzzle, i.e. inverting a hash function [94]. This is the design used e.g. by Bitcoin [18]. When an agent successfully finds a preimage, they are allowed to propose the next block and choose which transactions to

include. They are compensated for their with a block reward well as tips from the users that want their transactions included. The block producer agent essentially has full control over the next block, subject to the transactions included being 'legal', in the sense that no account ends up having a negative balance, and that all transactions verify under the appropriate public keys. If the block producer proposes an illegal block, it will be rejected by the users and they will lose their reward. The agents that attempt to solve the puzzle are known as 'miners'. Typically, finding a block is a rare event, so miners tend to pool together their resources and share the block reward [173]. This reduces the variance of the payoff from mining [51]. A blockchain that is based on inverting a cryptographic hash function is said to use proof-of-work [94, 141]. They have since been criticized for their intense energy consumption: a single Bitcoin transaction has an energy consumption of 700 kWh as of 2023 [233], almost the monthly energy consumption of an average US household (886 kWh per month as of 2021 [4]). To combat this, other solutions have been proposed that rely on the scarcity of other resources, including proof-of-stake [154], proof-of-space [95], or proof-of-space-time [190]. These schemes typically requires the miners to stake money — for Ethereum, a miner is required to stake 32 ETH [174], more than 60.000$, in order to be allowed to produce new blocks. To incentivize agents to expend effort to extend the chain, they are given a cash reward for their work. The data that is put in blocks come from users who want to put data on the chain. Each new block is typically quite small which leads to the issue of how the block producer should allocate the limited space on the block. To incentivize efficient allocations, users typically pay fees to the block producer for their data to be included, thus implementing an auction [212]. Because of the decentralized nature of the protocol, at any given time there might be several extensions of a given chain that may conflict with each other. The rule is then that whichever chain is longest is regarded as the current state of the chain. As long as the resource in question is controlled by a majority of honest agents, the longest-chain rule is known to result in various desirable properties [19, 36, 72, 107, 228]. However, for the purposes of this thesis, we will assume the blockchain is perfect and incorruptible and set aside practical issues of actually implementing the blockchain.

Smart Contracts. Blockchains are most famously used to implement decentralized money by having a set of accounts and using the data in the blocks to keep track of transactions between these accounts. The consensus property of the data prevents double spending attacks which is what makes the technology attractive as an alternative to central banking. The first blockchain that was used in practice is Bitcoin [18] which as of 2023 remains the largest blockchain (by volume) [68]. However, there is nothing that inherently limits the data stored on the chain to being transactions of money: in principle, any data can be stored on the chain, even programs. This idea was pioneered by the

blockchain Ethereum [257] to support the automatic execution of arbitrary programs on the chain, in what is know as *smart contracts* [237]. These programs are typically written in a high-level language such as Solidity and then compiled to a low-level representation such as EVM (Ethereum Virtual Machine) [258] bytecode that can then be executed automatically on the chain. Here, rather than the transactions stating that X money was transferred between this and that account (as is the case for Bitcoin), now transactions include instructions such as 'add the values of these two registers' or 'copy this value'. Each miner then maintains the full state of the chain and executes these bytecode instructions on their own machine. They are compensated for this activity by the user executing the instructions. This is referred to as the 'gas cost' of executing a transaction. Ethereum maintains a detailed list of gas costs for each of its bytecode instructions [258]. Fundamentally, just as how knowing the latest chain reveals the balance of all accounts, so does knowing the latest chain reveal the state of all smart contracts currently deployed on the chain. Thus, smart contracts enable the automatic processing and transfer of money which makes them attractive for various applications — and also attractive for adversaries to attack [93]. In this thesis, we will give several applications of smart contracts in the context of commerce, dispute resolution, and secure computation. Throughout this thesis, we will assume an abstract *ideal blockchain* that is secure and incorruptible. We have omitted a precise definition of the functionality offered by such a mechanism and trust that it is clear from the context what we mean. There are formal models of blockchains and smart contracts [19, 24, 152, 156] in the universal composability (UC) model [52], but consider such modeling outside the scope of this thesis.

Choice of Blockchain

We did not consider any specific blockchain in the previous sections: in fact, our work is mostly transparent to the choice of blockchain, as long as the blockchain is capable of executing smart contracts. As a result, our contracts inherit many properties of the underlying blockchain, which means they can be instantiated in a variety of ways. We now consider some instantiations of the contract in different types of blockchains. Instantiating the contract on a public ledger such as Ethereum is the most straightforward solution. Here, users are implicitly anonymous, while the flow of money is globally visible [185]. This means that accounts are pseudonymous and that all transactions between accounts are public. These curious properties makes it possible to use data mining algorithms to somewhat deanonymize its users [32, 132, 184, 209], leading some scholars to suggest that Bitcoin offers essentially no anonymity at all [149]. However, for some applications (notably in commerce), pseudonymity can be considered a feature: having access to the transaction history of a seller indicates how likely they are to cheat and holds the agents somewhat responsible for their actions [100, 238].

Privacy-Preserving Blockchains In response to the perceived lack of anonymity in traditional blockchains such as Bitcoin and Ethereum, alternative blockchains were developed that make use of cryptography to ensure anonymity, notably Monero [247] and Zerocash [26]. These systems hide the identities and the values of all transactions in the network by including the transactions in 'anonymity sets' from which it is hard to determine the origin and destination. Correctness of the blocks is ensured with the use of zero-knowledge proofs [38, 84, 118] (so-called *zk-SNARKs* [33, 126] are often used in practice). Variations of these systems are known to be provably secure under computational assumptions [103, 202]. The main drawback of using these blockchains is that they inherently make it impossible to enforce regulation on the goods being transacted, in particular, anti-money laundering (AML) and 'know your customer' (KYC) regulations [204]. A market on such a blockchain would likely be used primarily for criminal activity [46]. Note that Zerocash [26] does have a solution for managing KYC and AML, although this solution relies on being able to trust a single agent.

Accountability and Revocable Anonymity We now elaborate on how the proposed marketplace can be made to comply with laws and regulations, using the identity management system proposed by Damgård *et al.* [76] (a system which is used in practice by Concordium [79]). To register in such a blockchain, an agent needs to identify itself with an identity provider using some formal document. They can then create new anonymous user accounts to be used on the blockchain. Using a designated verifier zero-knowledge protocol, a user can prove to satisfy some predicate on their real identity, such as verifying that their age is ≥ 18. The users are, by default, anonymous, but can be *deanonymized* under suitable conditions, say if illegal behavior is suspected. This requires an agreement between several qualified authorities, the so-called *anonymity revokers*. For example, the local police, or the local courts may be able to deanonymize users in their relevant jurisdictions. This serves as a "best of both worlds" in that regular users retain their anonymity, while criminal users are subject to legal repercussions. This would allow for a kind of certification or blue-print of marketplaces based on smart contracts even if they are essentially anonymous, so long as the underlying blockchain uses revocable anonymity.

1.3 A Framework for Designing Secure Contracts

The contracts we propose for decentralized commerce and adjudication are quite simple and seemingly follow a generic pattern of using payments to incentivize certain behavior and change the equilibrium from dishonesty to honesty. It is quite natural to ask if we can adapt our solution to also work in other settings. Ideally, we would want a generic framework where the designer

inputs an arbitrary game and an arbitrary intended behavior, for which the framework outputs deposits that incentivize the intended behavior. Of course, this is not hard to do if we assume all actions taken are public record — we may then simply fine every agent that deviates from their intended behavior. If the fines are set sufficiently large, this always incentivizes them to do what we want (assuming the agents care about money). Unfortunately, the actions taken by the agents are not always externally observable: in the case of decentralized commerce, both the buyer and the seller know whether or not the correct item was indeed received – however, this information is not externally observable which is precisely why we needed the dispute resolution system. In order for this problem to be interesting we need some analogue of the dispute resolution system for arbitrary games.

Incentivizing Arbitrary Behavior. The third contribution of this thesis is a framework for computing payments to incentivize an intended behavior in an arbitrary game (Chapter 5). We give a model for specifying how actions can be externally observed through an *information structure*. Here, we consider some fixed alphabet of possible outcomes and associate with each leaf of the game a pdf on this alphabet that we may collect into an emissions matrix. Being able to observe all actions thus corresponds to having an alphabet with a size that equals the number of leaves, and having the emissions matrix equal a permutation matrix. An example of a non-trivial information structure is for the decentralized commerce game where the alphabet consists of three symbols: a success symbol denoting that everything went well, and two other symbols which are emitted by the dispute resolution system that correspond to the buyer, respectively the seller, being ruled dishonest by the jurors. A *payment scheme* can then be modeled as a mapping that takes as input a symbol from this alphabet and outputs a payment for each agent. Such a payment scheme can readily be implemented using a smart contract, assuming the contract has access to the appropriate information structure: in this case, the outcome of the commerce contract (see Fig. 1.4). Our model generalizes the model of 'adversarial level agreements' by George and Kamara [109] that can recovered as a special case of our model where the emissions matrix is a diagonal matrix.

We show several results on the relation between the payments and the information structures. Our first result is that payments can be used to implement any intended behavior if and only if essentially all actions can be observed, i.e. if the emissions matrix is full rank. We then show how to restate our commerce contract in this model and obtain a similar set of payments as we did by analyzing the contract explicitly. Next, we analyze the computational complexity of finding optimal payment schemes and find that this problem is equivalent to linear programming under logspace reductions, and hence PTIME-complete [121]. We then demonstrate the power of the

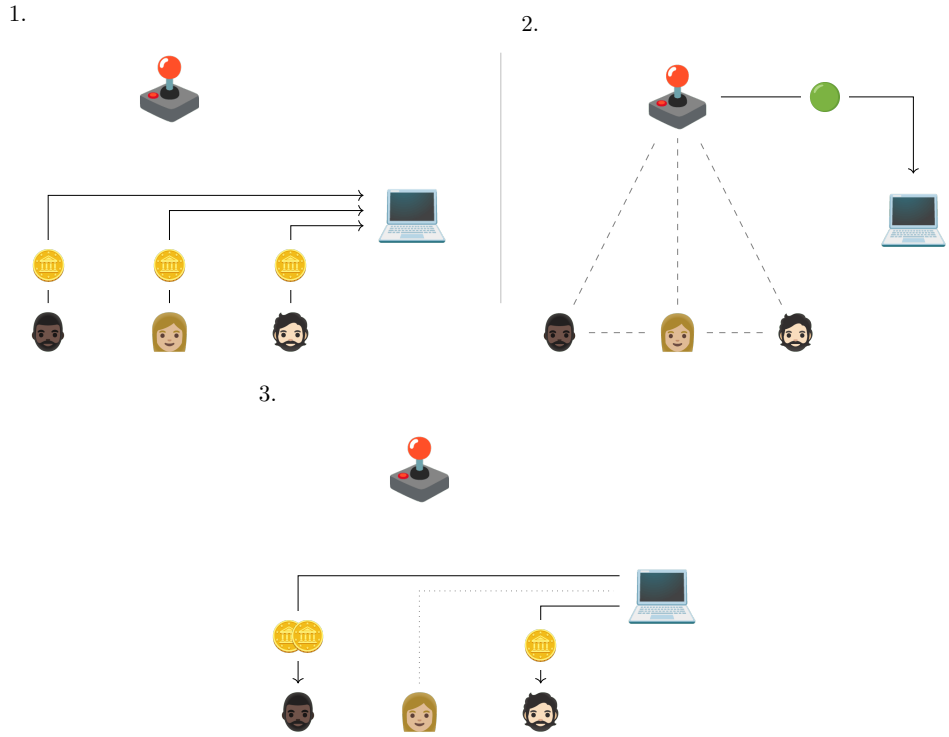

Figure 1.4: Workflow of the generic mechanism to incentivize a specified behavior in arbitrary games. 1. The agents transfer a deposit to the smart contract. 2. The agents play the (unchanged) game which emits a symbol to the smart contract. 3. The smart contract repays the agents based on the symbol it received. If these payments are instantiated correctly, the agents will have an incentive to behave as we would like them to.

model by analyzing a more complicated scenario involving secure multi-agent computation (MPC) [25, 60, 260, 261]. In MPC, a group of n agents wishes to compute some public function f on private inputs $x_1, x_2, \ldots, x_n$ using an interactive protocol, in such a way that the interaction leaks nothing about the inputs of a given agent — other than that which can be gathered from the function output $f(x_1, x_2, \ldots, x_n)$ itself. Such protocols are said to be secure against an honest-but-curious adversary, and the protocol is said to have *passive security*. If in addition, security holds even if some of the agents are allowed to deviate arbitrarily, we say the protocol enjoys *active security* [74]. In this thesis, we will consider an intermediate notion of security, known as *(publicly verifiable) covert security* where agents are allowed to deviate but are caught with some fixed (constant) probability [12, 16]. We show how to model a covert secure protocol using an information structure. We then derive an expression for the optimal size of the payments, such that the equilibrium of the resulting game is for all agents to act honestly. The result is a compiler

that takes as input a covert secure MPC protocol and outputs a protocol that is secure against rational agents. Famously, Halpern and Teague [130] showed that MPC is impossible for rational agents that 1) want to learn the function output, and 2) prefer that as few other agents also learn the function output. We circumvent this impossibility using quasi-linearity of the agents' utility functions. Namely, property 2) does not hold if we can pay an agent to accept that other agents also learn the output. Finally, we use properties of matrix norms to derive a lower bound on the size of the payments needed to ensure honesty. We find that this lower bound matches asymptotically the size of the payments needed for the MPC.

1.4 Smart Contracts and Commitments to Strategies

All our work so far is set in a model where the agents have shared access to a blockchain that allows them to deploy smart contracts. We have seen that this is a powerful model that, among other things, enables fully decentralized commerce, dispute resolution systems, and rational secure multi-party computation. However, there is an important aspect of this model that we have ignored so far: the agents may *themselves* deploy smart contracts to act on their behalf. This turns out to change the structure of the equilibria in non-trivial ways, as an agent may use the contract to restrict their set of moves, effectively committing to acting irrationally in certain situations. This in turn may change the best response for the other agents, thus inducing a meta-game of determining which is the best contract to deploy.

Stackelberg Equilibria. A simple model with commitments to strategies was introduced by von Stackelberg in 1934 [251] to model competing firms where a *leader* company has a market advantage and is allowed to choose their strategy first. The leader's strategy is then revealed to a *follower* company who then adaptively chooses their strategy. The resulting equilibrium is known as a *Stackelberg equilibrium*, and can also be extended to the setting of multiple leaders [227] and/or multiple followers [177]. Because of first-mover advantage [176], the leader is never worse off in the Stackelberg equilibrium, as the leader can simply commit to doing nothing. Stackelberg equilibria are quite well-studied and are important e.g. in control theory [23, 35, 88, 211] and security games [150, 155, 230]. It is well-known that computing Stackelberg equilibria on finite extensive-form games of imperfect information is NP-hard in the general case [172], and remains NP-hard even to approximate for some classes of games, see e.g. [42, 171] for an overview of some results in this direction. More sophisticated models of commitments have since been proposed, including reverse Stackelberg equilibria [17, 232] where the leader commits to a strategy conditioned on the strategy chosen by the follower: the leader commits to a mapping ϕ that for every strategy σ_F the follower may choose, defines what

strategy $\phi(\sigma_F)$ the leader will play, and gives this mapping to the follower. This is strictly advantageous for the leader since they can punish the followers for choosing the 'wrong' contract [17, 123, 135]. Such equilibria are even more advantageous for the leader since the leader can punish the follower for choosing the wrong strategy. A related line of work studies equilibria in the context of arbitrary Turing machines, in what is known as program equilibria [193, 241]. Note that while Rice's theorem implies that no computer program can verify a non-trivial property of another program [210], the first contract does not actually need to verify that the other contract satisfies an arbitrary predicate: instead, the first contract can provide the other agents with contracts that they must deploy (say, by publishing the source code on the blockchain), and if not, it executes the threat. This means the other agents are faced with the choice of deploying the contract given to them or accept the threat. A rational agent will then deploy the contract given unless they receive an even worse outcome by doing so. Reverse Stackelberg equilibria primarily find applications in routing [124, 125] and in control theory [125, 175, 236, 240].

Smart Contract Moves

The fourth contribution of this thesis is to give a formal model of games with smart contracts (Chapters 6 and 7). In this model, deploying a smart contract corresponds to making a 'cut' in the move set for that agent which induces a new game of exponential size, containing as subgames all the cuts that this given agent can make. Each subgame corresponds to a contract where the agent has cut away the given set of moves. We find that the Stackelberg equilibrium is retained as a special case with one smart contract. The model supports multiple such contracts that are allowed to reason about each other, by making cuts that are conditioned on the cuts chosen by the subsequent agents. We thus find that the reverse Stackelberg equilibrium can be recast as a special case, containing two subsequent smart contracts. This means that our model gives a unifying view of Stackelberg equilibria and reverse Stackelberg equilibria and establishes a hierarchy of generalizations hereof. We study the computational complexity of finding the subgame perfect equilibria (SPE) in these games and show several lower bounds. In general, we find that computing the SPE in these games is PSPACE-hard. More precisely, we show that computing the SPE in a game of imperfect information with k contracts is Σ_k^{P}-hard. The reduction establishes as special cases hardness results that were already known in the literature [41, 239]. Next, we show that computing the SPE remains PSPACE-hard in games of perfect information when the number of contracts is allowed to be unbounded (i.e. linear in the size of the game tree). We also give an upper bound and show that two-contract games of perfect information can be computed in time quadratic in the size of the description of the game.

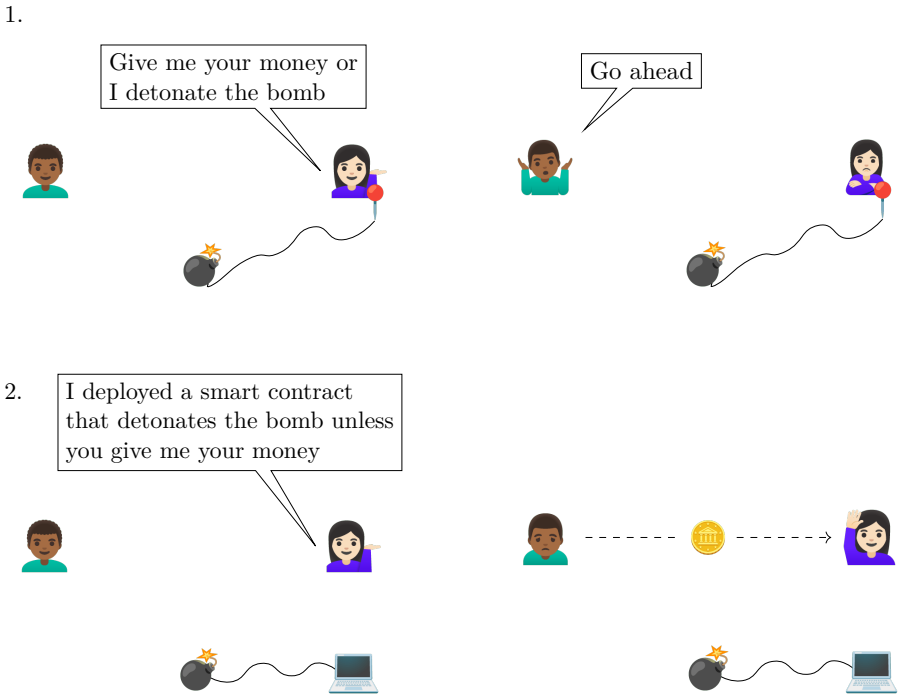

Figure 1.5: Illustration of the power of commitments to strategies. 1. Alice asks Bob to hand over all his money, with the threat of detonating a bomb that would kill both of them. Bob refuses to cooperate because he can infer that Alice would not actually detonate the bomb — it is but an empty threat. 2. Alice commits to detonating the bomb unless Bob hands over all his money which forces Bob to cooperate.

Stackelberg Resilience. These results, disappointingly, imply that reasoning about games deployed on a blockchain is a hard computational task. This is a potential problem for the design of smart contracts (see Fig. 1.5): typically, such contracts are analyzed *in vitro* under the assumption that agents cannot arbitrarily commit to strategies. The contracts are then deployed *in vivo* in a different context where the agents are, in fact, allowed to deploy smart contracts to act on their behalf. This means that we should not expect whatever analysis was conducted on the game without commitments to hold when the contract is actually deployed. We say that a game is *Stackelberg k-resilient* if it retains its equilibrium when k agents are allowed to arbitrarily commit to strategies. We show that Stackelberg resilience is downward transitive, in the sense that Stackelberg k-resilience implies Stackelberg $(k-1)$-resilience. Note that this is a non-trivial result because, conceivably, the removal of a contract could potentially thwart an attack that relied on some agent being forced to deploy a specific contract. The complexity results on smart contract moves imply that Stackelberg resilience is hard to compute in general, but that it

is efficiently computable for two-agent games of perfect information. We use this approach to analyze the escrow contract introduced in the beginning of this thesis and find that the contract is, in fact, Stackelberg resilient. This means that the addition of contracts does not change the security analysis. We also analyze another related escrow contract [11] and find that it is not Stackelberg resilient. These results establish that Stackelberg resilience is a non-trivial and hard-to-compute property. Finally, we consider an auction with multiple identical items and demonstrate a Stackelberg attack. This models the transaction fee mechanisms used by most blockchains. Here, a user commits to a strategy that ensures they receive one of the items for free, while forcing all other users to enter into a lottery for the remaining items. The attack works under reasonable assumptions on the valuations of the users, as long as the auction is not too congested. We find that the attack is detrimental to the auctioneer who loses most of their revenue. This implies that the transaction fee mechanisms of most major blockchains are vulnerable to these commitment attacks and may be cause for re-evaluation of the use of auctions in transaction fee mechanisms. It also suggests that other contracts deployed on major blockchains may be vulnerable to these attacks.

13:42

Hi there, I ordered a lamp and I haven't received it yet.

14:28

Hi, Can you show a proof that you really didn't receive the lamp?

14:29

1.5 List of Publications

During my PhD, I have published seven papers that are listed here.

[40] Romain Bourneuf, Lukáš Folwarczný, Pavel Hubáček, Alon Rosen, and Nikolaj Ignatieff Schwartzbach. PPP-Completeness and Extremal Combinatorics. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-263-1. doi: 10.4230/LIPIcs.ITCS.2023.22

[54] Ioannis Caragiannis and Nikolaj Ignatieff Schwartzbach. Outsourcing Adjudication to Strategic Jurors. To appear in *32nd International Joint Conference on Artificial Intelligence (IJCAI 2023).*, 2023

[78] Ivan Bjerre Damgård, Boyang Li, and Nikolaj Ignatieff Schwartzbach. More Communication Lower Bounds for Information-Theoretic MPC. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-197-9. doi: 10.4230/LIPIcs.ITC.2021.2

[129] Mathias Hall-Andersen and Nikolaj Ignatieff Schwartzbach. Game Theory on the Blockchain: A Model for Games with Smart Contracts. In Ioannis Caragiannis and Kristoffer Arnsfelt Hansen, editors, *Algorithmic Game Theory*, pages 156–170, Cham, 2021. Springer International Publishing. ISBN 978-3-030-85947-3

[165] Daji Landis and Nikolaj I. Schwartzbach. Stackelberg Attacks on Auctions and Blockchain Transaction Fee Mechanisms, 2023. To appear in *26th European Conference on Artificial Intelligence (ECAI 2023)*

[222] Nikolaj Ignatieff Schwartzbach. An Incentive-Compatible Smart Contract for Decentralized Commerce. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2021. doi: 10.1109/ICBC51069.2021.9461077

[223] Nikolaj Ignatieff Schwartzbach. Payment Schemes from Limited Information with Applications in Distributed Computing. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 129–149, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391504. doi: 10.1145/3490486.3538342

In addition, I have two manuscripts currently in submission.

[164] Daji Landis and Nikolaj I. Schwartzbach. Stackelberg Attacks or: How I Learned to Stop Worrying and Trust the Blockchain, 2023

[213] Sagnik Saha, Nikolaj Ignatieff Schwartzbach, and Prashant Nalini Vasudevan. The Planted k-SUM Problem: Algorithms, Lower Bounds, Hardness Amplification, and Cryptography, 2023

Funding. This work was funded by European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO); and VILLUM FONDEN under the Villum Kann Rasmussen Annual Award in Science and Technology under grant agreement no 17911.

Thesis Organization

This thesis is based on the papers [54, 129, 164, 165, 222, 223] and is organized as follows. In accordance with GSNS rules, parts of this thesis were also used in the progress report for the qualifying examination.

Chapter 2. We give a formal model of game theory and introduce the tools and notation that we will use for the remainder of this thesis. We also discuss the applicability of game theory and its limitations. This chapter is based on standard game theory literature.

Chapter 3. We propose a smart contract that enables any two agents to exchange physical goods and services for money with the use of a blockchain. The contract makes optimistic use of an adjudicator in a black-box manner. We analyze the contract as an extensive-form game and prove that the unique best-response for both buyer and seller is to behave honestly, assuming the adjudicator is biased in favor of honest agents. Finally, we consider various aspects of deploying the contract in practice, including transaction fees and choice of blockchain. This chapter is based on the paper [222].

Chapter 4. We analyze a simple adjudication game involving binary disputes and majority voting. We consider a set of rational and strategic jurors that are indifferent to the outcome of the case and consider different payment rules to incentivize them to exert effort in such a way that they collectively produce a correct adjudication outcome. We characterize the equilibria of the resulting game and find that for an appropriate choice of payments, there are there three classes of equilibria: a trivial one, a good one, and a bad one. We perform simulations to argue that, in practice, the jurors tend to reach the good equilibrium. This chapter is based on the paper [54].

Chapter 5. We propose a framework for reasoning about payments in the context of blockchains. Our crucial insight is the notion of an *information structure* that specifies which information is observable by the blockchain. We study the problem of finding minimal payments that incentivize a specified behavior, and find that this problem is equivalent to linear programming under logspace reductions and thus P-complete. We give various applications of the framework in the context of decentralized commerce and secure multi-party computation. This chapter is based on the paper [223].

Chapter 6. We identify a subtle issue in deploying smart contracts, caused by the fact that agents may themselves publish smart contracts. This induces a 'meta game' of determining the optimal contract to commit to. We propose a model that captures these types of commitments by introducing 'smart contract moves' that allow an agent to make cuts in their move set. We show that our model captures Stackelberg equilibria (respectively, reverse Stackelberg equilibria) as special cases, as games with one contract (respectively, two consecutive contracts). We establish several bounds on the computational complexity of these games and find that determining the SPE is PSPACE-hard, even when the games are restricted to having perfect information. However, we give an efficient algorithm that works for two-contract games of perfect information. This chapter is based on the paper [129].

Chapter 7 We study how smart contract capability changes the equilibria of games. A game that does not change is said to be *Stackelberg resilient*. We find that the smart contract from Chapter 3 is indeed Stackelberg resilient. We analyze a class of transaction fee mechanisms that we find to not be Stackelberg resilient: we demonstrate an attack whereby the users will spontaneously organize to conspire against the miner. The attack allows a user to have their transaction included for free, while coercing the remaining users into entering a lottery for the rest of the space on the block. We find that the attack works under natural conditions for both first-price auctions, second-price auctions and EIP-1559 (the transaction fee mechanism used by Ethereum). This chapter is based on the papers [164, 165].

Chapter 2

Homo Economicus

> "*We call a man irrational when he acts in a passion, when he cuts off his nose to spite his face. He is irrational because he forgets that, by indulging the desire which he happens to feel most strongly at the moment, he will thwart other desires which in the long run are more important to him. If men were rational, they would take a more correct view of their own interest than they do at present; and if all men acted from enlightened self-interest the world would be a paradise in comparison with what it is.* "
>
> Bertrand Russell

GAME THEORY is the study of rational agents and their interactions. It seeks to model interactions that involve agents with potentially mutually incompatible preferences and tries to predict how the agents will behave. Each agent is taken to be rational, in the sense that they take actions that maximize a certain numerical quantity known as their *utility*. This leads the agents to strategize on which actions to choose, based also on their knowledge of the preferences of the other agents. In this chapter, we will give a brief introduction to game theory, focusing on the tools we will be using. This chapter is based on standard game theory literature unless otherwise stated; we refer to [194] for more details. We assume familiarity with set theory, algebra, and basic computational complexity theory.

Utility as a Numerical Quantity. The idea that utility can be measured as a numerical quantity has its roots in utilitarianism, pioneered by philosophers such as Bentham [27], Mill [188], and Edgeworth [98]. This reasoning has since been a pillar of modern economics, though it has often been criticized since for being impossible to measure or incomplete: a popular theory by Kahnemann, Wakker, and Sarin [145] suggests distinguishing between 'decision utility' and 'experienced utility', the former of which relates to the decisions an

agent makes, while the latter corresponds to the experienced valence. Stigler [234, 235] proposed the theory of marginal utility that, motivated by wanting to predict actions, only seeks to model the change in utility that certain actions would give. This solves the problem of having to measure the full depth of the 'sea' of utility and instead focus on the 'waves' of the relative change in utility, in reference to Georgescu-Roegen [110] using the ocean as a metaphor for utility. Scholars such as Pareto [196] reject altogether the idea that utility is cardinal, and instead suggests making do with ordinal utility, i.e. comparing the preferences of agents. However, Von Neumann and Morgenstern [250] show that under certain axioms on the rational behavior of an agent, the agent acts *as though* they are maximizing a utility function. These axioms have been criticized both by theoreticians and empiricists. A related theory by Savage [217] gives a different set of seven axioms that are consistent with maximizing subjective expected utility. In recent years, support for quantifying utility has emerged also in consciousness research, e.g. by Johnson [143] who propose a 'symmetry theory of valence' which posits that experienced utility relates to the symmetry of the internal representation of an experience (that could potentially be measured). Throughout the years, numerous alternative theories to cardinal utility have been proposed [114, 220, 244]. Throughout this thesis, we will take for granted that utility can be measured as a numerical quantity and consider these alternative theories outside the scope of this thesis.

Definition 2.1 (Game)**.** *A* game *on n agents consists of n sets $S_1, S_2, \ldots, S_n$ that comprise the set of strategies available to each agent. Elements of S_i correspond to deterministic (pure) strategies that agent i may choose. A (pure) strategy profile $s \in S := S_1 \times S_2 \times \cdots \times S_n$ specifies a strategy for each agent. We assume the existence of a utility function,*

$$u : S \to \mathbb{R}^n,$$

that for each set of pure strategy profile $s \in S$ and for every $i \in [n]$ gives the (expected) utility $u_i(s)$ that agent i receives when playing the pure strategy profile s.

In general, the strategies can be randomized (in which case they are usually called *mixed* strategies) so we will, in fact, instead consider distributions on S, i.e. a distribution on the pure strategies of each agent. We overload notation and denote also by s_i a mixed strategy for agent i and trust it is clear that we actually mean a distribution on S_i. Also, unless otherwise stated, we will assume that the strategies for each agent are independent. Correlated strategies can be used to model cases where agents privately receive public information from a correlation device [14, 15]. Such strategies can result in more efficient outcomes than their uncorrelated counterparts: the canonical example of a correlated equilibrium is that of traffic lights, where the suggested strategies of 'stop' or 'go' prevent vehicles from colliding.

Remark 2.2. *We will consistently use the terminology* agent *to refer to the participants of the game. In certain contexts, it might have been more appropriate to use the term* player *or* party*: we have instead opted for homogenizing the language and use the term agent regardless of the context.*

Properties of Utility Functions.

It will often be convenient to make assumptions on the structure of the utility function u. These assumptions serve mainly to simplify the analysis and in most cases can be removed, at the cost of making the analysis less tractable and the resulting theorems more complicated. As such, throughout most of this thesis we will be assuming that all utility functions satisfy *quasi-linearity* and *risk neutrality*, that we will elaborate on now.

Quasi-Linearity. The first property intuitively states that agents care about money. More precisely, suppose an agent gets x utility from a certain outcome (measured in terms of some base currency, the *numéraire*). Quasi-linearity then means that if this outcome occurs and we pay the agent y money in said currency, the utility of the agent is $x+y$. More formally, if an outcome consists of a set of goods $x_1, x_2, \ldots, x_n$, the utility function u is said to be quasi-linear in x_1 if for every i, there is a function u_i' such that,

$$u_i(x_1, x_2, \ldots, x_n) = x_1 + u_i'(x_2, x_3, \ldots, x_n).$$

When we say we assume the agents have quasi-linear utilities, we implicitly mean *quasi-linearity in money*, such that x_1 is the net change in funds for agent i. We stress that quasi-linearity of money is frequently contested by economists, who might opt for other measurements such as log-utility [29], or exponential utility [10]. Note that using different measurements does not fundamentally change the analysis we will conduct as the utility can then be considered quasi-linear in 'log money'. However, it may change the payments used in various theorems. We consider such modeling outside the scope of this thesis.

Risk Neutrality. The second property states that an agent is indifferent between obtaining two outcomes with the same *expected* utility. That is, if a strategy profile S results in a probability distribution p on a set of outcomes Ω, each of which $x \in \Omega$ gives a utility of $u_{i,x}$ to agent i, then we have that,

$$u_i(S) = \sum_{x \in \Omega} p(x)\, u_{i,x}.$$

Intuitively, such agents are indifferent to risk and would be equally happy to receive 50€ or flip a coin for the chance to win 100€. It was argued by

Bernoulli[1] that humans are not risk neutral: consider a pool of money starting at 2€ and an agent playing that repeatedly flips a fair coin. If the agent gets heads, they are given whatever money is left in the pool, and whenever they get tails, the pool doubles in size. In this case, the agent would receive 2€ with probability $\frac{1}{2}$, 4€ with probability $\frac{1}{4}$, and so on. In this case, the expected value of the game is,

$$\frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 4 + \frac{1}{8} \cdot 8 + \cdots = 1 + 1 + 1 + \cdots = \infty,$$

which diverges. This means that, under the expected utility theory, the agent would be happy to pay any finite amount of money (say 1.000.000€) to play this game. This does not seem realistic, and Bernoulli[2] later writes that he would not even pay 20€[3] to play this game [29]. This game has since been known as the St. Petersburg paradox. The usual way to resolve this alleged paradox involves invoking the law of diminishing marginal utility, as suggested by Bernoulli [30]. The idea is that the value of 1€ diminishes as you get more money, i.e. there is some sublinear function $f(\cdot)$ such that the utility of having ω is $f(\omega)$. A common choice is $f(\omega) = \log(\omega)$, known as log-utility. In this case, the paradox disappears as now the expected utility of the game is a finite number that depends on the precise choice of $f(\cdot)$. However, as pointed out by Menger [186], the paradox can always be reintroduced by changing the game such that the pool of money grows even faster. In any case, such solutions explicitly reject our first assumption of quasi-linearity and are thus undesirable. Other solutions involve taking the finite budget of the casino into account [104], or discounting probabilities that are sufficiently small [82]. There are other ways to solve the alleged paradox that involve rejecting the expected utility theory altogether, most notably using prospect theory [144, 244], or more recently, using ergodicity economics [199, 200]. However, we consider such modeling outside the scope of this thesis.

2.1 The Nash Equilibrium

Ultimately, the reason to model games in formal mathematical language is to use mathematics to predict how the agents will behave. Being rational, agents will take actions that maximize their (expected) utility, subject also to the preferences of the other agents. The key observation is that once we fix a strategies of all agents, the situation is only 'stable' if none of the agents may deviate to obtain a better outcome (or else they would). This leads us to the most important definition in game theory.

[1] Nicolaus Bernoulli.

[2] Daniel Bernoulli, the cousin of Nicolaus Bernoulli.

[3] Bernoulli, of course, had no knowledge of euros; in his version of the game, the currency was ducats, a coin made of 3.5g gold, worth about 200€ as of May 31, 2023.

Definition 2.3 (Equilibrium, [191])**.** *A strategy profile $s^* \in S$ is an* equilibrium *for a game G, if for any i and any s_i, it holds that,*

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*). \tag{2.1}$$

If the inequality only holds with a 'slack', i.e. if for some constant $\varepsilon > 0$, it holds that $u_i(s_i^*, s_{-i}^*) + \varepsilon \geq u_i(s_i, s_{-i}^*)$, we say that s^* is an ε-equilibrium*. If instead, for every $s_i \neq s_i^*$, the above inequality is strict (i.e. $>$ instead of $\geq$), we say that s^* is a* strong equilibrium*.*

The equilibrium identifies those strategy profiles that are stable with respect to unilateral deviations by the agents. Note that while the equilibrium is named after Nash [191], it was first used by Cournot [71] to study the strategic behavior of competing firms. The equilibrium is often considered the most basic property a strategy profile must satisfy to be played by rational agents. However, there are various relaxations of the notion of equilibrium that are in some sense consistent with being rational, e.g. rationalizable strategies [28, 197]. In this thesis, we are only interested in studying equilibria and various refinements hereof.

Bounded Rationality. An implicit assumption of the equilibrium is that there is full knowledge on the strategies chosen by other agents: an agent best-responds to the full strategy profile chosen. In many cases, this is a limiting assumption. To model uncertainty about the strategies of other agents, one may instead use other notions such as (perfect) Bayesian Nash equilibria [63, 146], or sequential equilibria [158], both of which assign a prior distribution on each information set in the game, modeling the beliefs that the agent has. A related and problematic aspect of the equilibrium is that rationality might be a tall order: agents may not possess the required knowledge or computational resources to realize full rationality. This phenomenon is well-documented in behavioral game theory [6, 99] and can be modeled using e.g. trembling-hand equilibria [225] or quantal response models [181, 182], both of which allow agents to make irrational choices with non-zero probability.

Rationality and Cryptography. Rationality is also seemingly incompatible with cryptographic assumptions. Suppose that a renegade computer scientist has acquired a nuke that they have hid in an undisclosed location. In 72 hours, the nuke detonates unless a proof of the Riemann hypothesis is uploaded to the Ethereum blockchain. The scientist is credible and there is no way to find the bomb in the allotted time. Clearly, the rational thing for the agent to do is to simply upload a proof of the Riemann hypothesis. However, this is likely hard to do unless the agent has infinite computing power (in which case they could brute-force a proof). That is, being rational seems fundamentally at odds with being computational bounded. While this example is a bit conceived, this

is a real concern when modeling cryptographic primitives in the context of rational agents: a fully rational agent would be able to trivially break any non-information-theoretic cryptography. This is problem for our blockchain setting where we crucially rely on e.g. the agents not being able to break the underlying hash function (for proof-of-work). There are ways to model such agents using computational extensive-form games proposed by Halpern, Pass, and Seeman [131]: here, we have an ideal representation of a game as an extensive-form game G, and consider a (non-uniform) infinite sequence $\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots$ of real games, one for each choice of security parameter $\lambda \in \mathbb{N}$. A strategy profile is a (non-uniform) sequence of strategy profiles $S^{(0)}, S^{(1)}, S^{(2)}, \ldots$, that we say is a *computational equilibrium* if there is a negligible[4] function $\mathsf{negl}(\cdot)$ such that $S^{(\lambda)}$ is a $\mathsf{negl}(\lambda)$-equilibrium. This means we can make the incentive the cheat small by setting the security parameter to a reasonable value. Halpern, Pass, and Seeman then give conditions under which an equilibrium in G is also a computational equilibrium in $\mathcal{G}^{(0)}, \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots$ (under an appropriate translation of the strategy profile). For the purposes of this thesis, we will conveniently sweep such issues under the rug and assume our underlying cryptographic primitives are secure and incorruptible.

Existence and Computation of Equilibria. Nash famously showed that every game admits an equilibrium when the agents are allowed to randomize their strategies [191]. His proof used Kakutani's fixed-point theorem [147] and, as a result, is non-constructive; a plethora of work in computer science has since been devoted to studying the computational complexity of finding equilibria in various models of games [42, 61, 69, 81, 115, 172, 203]. The consensus is that finding equilibria is, in general, hard to do. Famously, Daskalakis, Goldberg, and Papadimitriou [81] showed that finding an equilibrium in games with at least three agents is complete for the complexity class PPAD[195]. Informally, this means that finding an equilibrium is as hard as finding the end of an exponentially long line, for which the best-known algorithm is to essentially take every step [168]. This is clearly not efficient, and hence it is commonly believed that finding equilibria is hard. This was reduced to two agents by Chen, Deng, and Teng [61]. The same authors later showed that even approximating an equilibrium is hard [62]. Note that the equilibrium of a game is, in general, not unique [183], and in fact, it is known that finding a second equilibrium is NP-hard. Determining if a game has a pure equilibrium is NP-hard [120], and determining whether a game has a strong Nash equilibrium is Σ_2^{P}-complete [120]. Finding an equilibrium can also be shown to be hard under various cryptographic assumptions [34, 64]. For a survey of more hardness results on finding equilibria, we refer to the book by Nisan, Roughgarden, Tardos, and

[4]Here, negligible is a standard cryptographic definition that means that the function grows slower than any polynomial, i.e. $f(\cdot)$ is *negligible* if for any $c \in \mathbb{R}$, it holds that $f(x) = o(x^c)$. The prototypical example is the exponential function 2^{-x}.

Vazirani [192]. As a caveat, note that the word 'efficient' is used in the usual computer science sense, meaning 'computable in polynomial time in the size of the instance'. In practice, chess and go are far too large to exhaust fully, even if backward induction in principle takes linear time in the size of the games. That is, efficiency describes the *asymptotic* relationship between the size of the instance and the time it takes to solve it, as the size of the instance tends to infinity — it says nothing about the time it takes to solve any concrete instance. As a curiosity, chess has been solved when there are at most seven pieces on the board [208], and as of 2023, work is ongoing to increase this number to eight.

2.2 Representations of Games

Normal-Form Games

The classic formulation of a game by Von Neumann and Morgenstein [249] is as a real-valued matrix. The game involves two agents that act simultaneously, one of whom chooses a row of the matrix and the other of whom chooses a column. This determines an entry of the matrix, the value of which defines the utility given to the row agent, and also the utility taken from the column agent. That is, the game is *zero-sum*, in the sense that the sum of all utilities for every outcome is zero. Any two-agent zero-sum game be modeled in this way, and includes most competitive interactions where one agent can 'win'. This includes chess, go, or rock-paper-scissors. In the case of the former two, the resulting matrix is far too larger to write down explicitly, but we have done so for rock-paper-scissors for the purpose of illustration, see Fig. 2.1.

	🪨	📄	✂️
🪨	0	-1	1
📄	1	0	-1
✂️	-1	1	0

Figure 2.1: The game of rock-paper-scissors modeled as a matrix. The agents act simultaneously, one of whom choose a row and the other chooses a column. Here, 0 is a draw, -1 is a win for the column agent, and 1 is a win for the row agent.

Von Neumann [245] famously showed that any finite, zero-sum, two-person game has a mixed equilibrium that can be found efficiently using linear programming. We define the *value* of the game as the expected utility for the row agent at this equilibrium. In the case of rock-paper-scissors, we find that

the value is zero and that the optimal strategy for each agent is to play each hand with probability $\frac{1}{3}$. These zero-sum matrix games are a special case of a more general formalism of games, known as *normal-form games* [250]. In the general case, we would have multiple arrays of matrices (tensors) of utilities, one for each agent, that gives their payoff in each combination of moves by the other agents, e.g. for general-sum two-agent games we would have two payoff matrices. Two-agent normal-form games can be solved using the Lemke-Howson algorithm [168] that pivots around the corners of a polytope comprising the set of feasible solutions, in order to eventually arrive at the equilibrium. The algorithm runs in exponential time in the worst case [218], though it is quite efficient in practice [203]. At a high level, it works in much the same manner as the simplex algorithm for solving linear programs [80]. In this thesis, we shall not be using normal-form games and have thus omitted their precise definition.

Extensive-Form Games

In this section, we present a different model of games, extensive-form games, that will serve as our model of choice throughout the rest of this thesis. Extensive-form games were first introduced by Kuhn [160] This model is equivalent in some sense to the normal-form representation[5]. Namely, rather than thinking of the game as a matrix, we imagine all games as *trees* (in the computer science sense). The leaves correspond to outcomes and branches are decisions that an agent has to make. All leaves of the tree are labeled with a utility for each of the participants, and each branch of the tree is owned by exactly one of the agents. The game is played by, starting at the root, letting the agent who owns the current node choose one of its children to recurse into. This process continues until we reach a leaf which terminates the game. Each agent is then given the utility associated with them with the corresponding leaf. To see rock-paper-scissors as an extensive-form game, see Fig. 2.2.

Definition 2.4 (Extensive-Form Game of Perfect Information)**.** *An* extensive-form game of perfect information *consists of:*

- *A rooted tree T, the leaves of which are labeled with a utility for each agent. We denote by $L \subseteq T$ the set of leaves in T, and suppose some arbitrary but fixed order on its elements, $\ell_1, \ell_2, \ldots \ell_m$.*
- *An $n \times m$ matrix $\mathbf{U} = (u_{ij}) \in \mathbb{R}^{n \times m}$, called the utility matrix of G, that for each agent P_i specifies how much utility u_{ij} they receive when the game terminates in the leaf $\ell_j \in L$.*

[5]Any normal-form game can be converted to an extensive-form game of the same size. The same holds in the opposite direction but incurs an exponential blow-up on the size of the representation.

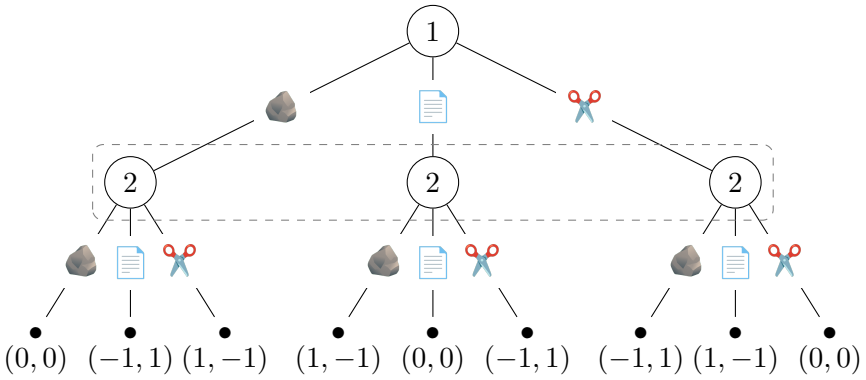

Figure 2.2: Rock-paper-scissors as an extensive-form game. The game starts with a move for agent 1 who chooses which hand to throw. For each choice of hand, agent 2 has a subgame where they also choose their move. The leaves encode which agent wins the game, with the first coordinate being the payoff to agent 1 and the second coordinate the payoff to agent 2. The dashed rectangle around the moves of agent 2 is an information set. It enforces that agent 2 cannot condition their move on which move agent 1 chose. This game is equivalent to its normal-form representation in Fig. 2.1.

- *A partition of size n on the nodes $T \setminus L$, with each set corresponding to the nodes owned by a given agent.*

Extensive-form games may be depicted as upside down trees whose branches are labeled with the index of an agent, and correspond to the moves in the game. The leafs are then labeled with a vector that assigns to each agent their corresponding utility. The game is played, starting at the root, by recursively letting the agent who owns the current node choose a child to descend into. We stop when a leaf ℓ_j is reached, after which agent P_i is given u_{ij} utility. A mapping s_i that dictates the moves an agent P_i makes is called a strategy for that agent and is said to be pure if it is deterministic, and mixed otherwise. A set of strategies $s = (s_1, s_2, \ldots, s_n)$, one for each agent, is called a strategy profile and defines a distribution on the set of leaves in the game. We overload notation and let $u_i(s)$ denote the expected utility for agent P_i when playing the strategy profile s. If $C \subseteq \{1, 2, \ldots n\}$ is a set of indices of agents, a coalition, we denote by $-C$ its complement so that we may write a strategy profile s as $s = (s_C, s_{-C})$.

Moves by Nature. Occasionally in the literature, extensive-form games have an additional component to their structure that models random selection: rather than partitioning the set of branches into n sets, we partition it into $n + 1$ sets, with the new set corresponding to 'moves by nature' with nature playing a fixed strategy of common knowledge (i.e. there is a fixed distribution

on the children of each move by nature). Moves by nature are sometimes called chance nodes. We have omitted this component from our definition, as our assumption of risk-neutrality allows us to assume w.log. that the trees contain no such chance nodes.

Bifurcating Trees. It will occasionally be convenient to assume that T is a bifurcating tree, i.e. each branch has exactly two children. Note that this is without loss of generality, as any generic T may be made bifurcating by collapsing branches with only one child, and replacing branches with more than two children by small binary trees, with all branches belonging to the same agent. Doing so increases the size of the tree by at most a factor $O(\log m)$.

2.3 Subgame Perfection

A *subgame of* G is a subtree $G' \subseteq G$ that is transitively closed under the 'is child of'-relation, i.e. whenever $v \in G'$ and $w \in G$ is a child of v, then $w \in G'$. Our definitions suffice for games of perfect information, where at each step, an agent knows the actions taken by previous agents, though, more generally, we may consider partitioning each set of nodes belonging to an agent into *information sets*, the elements of which are sets of nodes that the agent cannot tell apart. More formally, any agent must assign the same strategy to all branches belonging to the same information set[6]. Also subgames cannot cut through information sets, making it possible for a non-trivial game to contain only itself as subgame. A game of perfect information is a special case where all information sets are singletons. A game of perfect recall is a game in which no two distinct nodes belonging to the same information set are related transitively under 'is child of'. In games of perfect recall, agents 'remember' all actions they have previously taken in the game. In this thesis, we will mostly be working with games of perfect information.

Empty Threats. While an equilibrium is some sense natural, sometimes strange equilibria appear in games. Consider the following game, known as the ultimatum game (Fig. 2.3). Here, two agents have to split some resource, say a bundle of cash, in two; the first agent (the proposer) proposes a split, e.g. a fair split or an unfair split, where say the second agent (the responder) only receives 10% of the total value. The responder can then accept the offer or reject it, in which case both agents receive nothing (maybe they burn the money in rage). If the proposer makes a fair split, the responder is forced

[6]Technically speaking, we have eluded the precise definition of what it means for an agent not to 'tell apart' two node. This is of no concern for defining the games used in this thesis, though it technically makes some of the wording less precise. This can be fixed using epistemic modal logic [159, 229], though we have omitted such a treatment in this thesis, as we believe the less formal version suffices for our purposes.

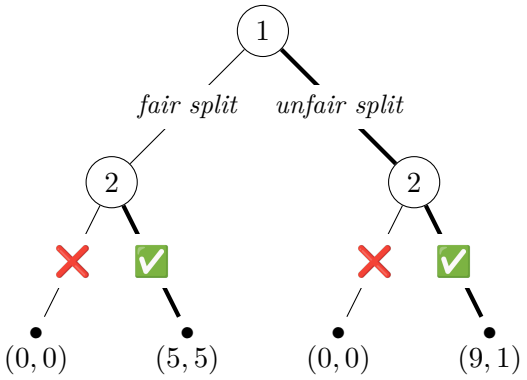

Figure 2.3: The ultimatum game as an extensive-form game. agent 1 either proposes a fair or an unfair split to agent 2, who subsequently chooses whether to accept or reject the offer. This game has three equilibria, though only one of them does not involve empty threats, namely the unique subgame perfect equilibrium which is denoted in bold edges.

to accept only the fair split (i.e. in the unfair split branch of the game, the responder has to reject the unfair split), as otherwise the proposer would deviate to propose an unfair split. This is the most equitable outcome for the two parties.

However, there are other equilibria of the game: if instead, the proposer makes an unfair split, the responder has to accept the unfair split, as otherwise they receive nothing. However, in case they do so, they can arbitrarily accept or reject the offer in the other branch without changing their utility. This means there are three equilibria of the game. However, we will now argue that one of these equilibria is more 'natural' in some sense. That is, the strategy profile where the proposer suggests an unfair split and the responder accepts *only* the unfair split, although it is an equilibrium, seems unnatural; clearly, the responder would also accept the fair split on the off-chance the proposer makes a fair split. In some sense, also the fair split is also unnatural, as the proposer, having the first move of the game, obtains a strictly larger utility by proposing an unfair split. The reason these games seem strange is that we can identify sequences of moves that seem irrational. In other words, the two strange equilibria are not equilibria in every subgame of the original game. If an equilibrium remains an equilibrium in each subgame of the original game, we say it is a subgame perfection equilibrium. We have thus identified a subclass of equilibria with a stronger property, which is known in the literature as a *solution concept*, specifically a *refinement* of the standard equilibrium. A subgame perfect equilibrium (or simply, SPE) is known to exist for every game with perfect recall [160].

Definition 2.5 (Subgame Perfect Equilibrium)**.** *A strategy profile s^* is a* subgame perfect equilibrium (SPE) *for a game G if it is an equilibrium for every subgame*

of G.

Backward Induction. There is a nice and simple algorithm to compute subgame perfect equilibria for games of perfect information known as *backward induction*. In chess, the algorithm is also known as retrograde analysis [242]. The algorithm was first used by Cayley [57] to solve the secretary's problem. The idea is to reason 'backwards' from the end of the game to infer the actions of the agents. Consider some subgame belonging to agent i. In order to determine what agent i will do, we may consider each choice available to them, and rank them according to the utility offered to agent i in each of them. To compute these values, agent i has to determine their payoff in each of the corresponding subgame, and so runs the algorithm recursively. The algorithm stops when it reaches a leaf, in which case it is output as the SPE. The algorithm thus runs in time $O(m)$ where m is the size of the tree. It will often be convenient to assume each agent has a strict order on the outcomes of the tree, i.e. their utilities are distinct. In this case, we say the game is in *generic form*. A pseudo-code of this algorithm is depicted in Algorithm 1.

Data: Extensive-form game G.
Result: Dominating leaf $\mathbf{u}^*$.
function BackwardInduction(G)**:**
  switch G **:**
    case Leaf($\mathbf{u}$) **:**
      return $\mathbf{u}$

    case Branch($i, G^{(1)}, G^{(2)}, \ldots, G^{(k)}$) **:**
      for $j = 1 \ldots k$ **:**
        $\mathbf{u}^{(j)} \leftarrow$ BackwardInduction($G^{(j)}$)
      $j^* \leftarrow \arg\max_{j=1\ldots k} \mathbf{u}_i^{(j)}$
      return $\mathbf{u}^{(j^*)}$

Algorithm 1: Pseudo-code of backward induction. The algorithm computes the subgame perfect equilibrium (specifically, it computes the utility vector of the *dominating leaf*) in an extensive-form game with perfect information. For simplicity, we are assuming that the game is in generic form, so that the choice of $\mathbf{u}^*$ is unique; we can modify the algorithm to compute *all* SPEs at the cost of an $O(m)$ overhead in the computation.

Unexpected Hanging Paradox. Philosophers have since discuss the merits of backward induction, in what is known as the *unexpected hanging paradox* [65].

> A prisoner is told that he will be hung by the neck at noon some weekday the following week. He is not told what day the execution

> will occur, but he is told it will come to him as a surprise. The prisoner then reasons that it cannot happen on Friday, since this will not come to him as a surprise. But if it does not occur on Friday, then it also cannot be Thursday, since by Wednesday evening, he would know his fate. He continues and concludes it also cannot be on Tuesday, nor on Monday. The prisoner happily concludes that he will not be executed and retires to his cell. Monday at noon, the executioner knocks on his door – that was quite surprising.

While the story suggests an apparent limitation of backward induction, the paradox results from the self-referential nature of the setup.

2.4 Multi-Lateral Deviations

In this section, we generalize the concept of equilibrium to account for deviations by multiple agents. That is, the standard (Nash) equilibrium only takes into deviations by a single agent. This is often inadequate for modeling cryptographic scenarios where an adversary is usually assumed to control a constant fraction of the agents. In such cases, the equilibrium is too weak to guarantee that the adversary does not have an incentive to deviate. The definition is by Abraham, Dolev, Gonen, and Halpern [1].

Definition 2.6 (t-Robust Equilibrium)**.** *Let G be an n-agent game with strategy space $S_1 \times S_2 \times \cdots \times S_n$. A strategy profile $s^* = (s^*_C, s^*_{-C})$ is said to be a* t-robust equilibrium *if for every coalition $C \subseteq [n]$ of size $|C| \leq t$, and every joint strategy $s_C \in \prod_{i \in C} S_i$, and every $i \in C$, it holds that,*

$$u_i(s^*_C, s^*_{-C}) \geq u_i(s_C, s^*_{-C}).$$

This definition is essentially just the 'natural' extension of the (Nash) equilibrium to multilateral deviations. It has a parameter t that controls the threshold on the size of the coalition. The definition was motivated by *secret sharing* with rational agents. Here, n agents each have a share of a secret value. If $t + 1$ agents pool their shares, they can learn the secret, while any set of t agents has no information on the shares. Such schemes are known to exist from various algebraic objects. When the agents are rational, it is not clear, however, that an agent would want to contribute their share. Indeed, Halpern and Teague [130] show that secret sharing is impossible in a model where 1) agents strictly prefer learning the output, and 2) agents prefer that as few other agents learn the secret. Halpern and Teague also show when removing the second assumption, 1-robust secret sharing is possible using a randomized mechanism. Later, Abraham, Dolev, Gonen, and Halpern [1] show that secret sharing can be achieved with $(t - 1)$-robustness using also a randomized mechanism.

Game-Theoretic Security

We now define what we mean when we say a game is secure in a game-theoretic sense. The following definition originates in [222], though the following is taken (almost) verbatim from [223], with only minor changes to the prose. At the least, security should mean the honest strategy profile is an equilibrium, though this is likely not sufficient for some applications. The fact that the honest strategy profile is an equilibrium does not mean it is the only equilibrium. Namely, there might be several dishonest strategy profiles with the same properties, and there is no compelling reason for agents to be honest when given the choice not to. In fact, there might be reasons to be dishonest that are not captured by the utilities of the game, say for revenge or out of spite. To remedy this, we want to quantify how much utility agents lose by deviating from the honest strategy profile, in effect measuring the cost of dishonesty. We introduce a parameter $\varepsilon \geq 0$ such that being dishonest results in the deviating agents losing at least ε utility. A game with this property is considered secure against ε-deviating rational agents. We give a definition that generalizes t-robust subgame perfect equilibria for finite games of perfect information. Let G be a fixed finite game with n agents. Let s^* be the honest strategy profile, and $\mathbf{u}^* \in \mathbb{R}^n$ the corresponding utility vector. We say a utility vector $\mathbf{u} \in \mathbb{R}^n$ is *C-inducible in G* for a coalition C if there is a strategy s_C such that playing $s = (s_C, s^*_{-C})$ terminates in a leaf ℓ labeled by $\mathbf{u}$ with non-zero probability.

Definition 2.7 (Game-Theoretic Security)**.** *Let G be a game, and s^* an intended strategy profile. We say G has* ε-strong t-robust game-theoretic security *if for every subgame of G, and every C-inducible vector $\mathbf{u} \neq \mathbf{u}^*$ in that subgame with $|C| \leq t$, and every $i \in C$ with $s_i \neq s^*_i$, it holds that:*

$$\mathbf{u}^*_i \geq \mathbf{u}_i + \varepsilon \tag{2.2}$$

In other words, every coalition of $\leq t$ agents that deviates from s^* at any point in the game should lose at least ε utility for each member of the deviating agent. We note that for finite games of perfect information, t-robust subgame perfect equilibria are retained as a special case of this definition by letting $\varepsilon = 0$.

Throughout parts of this thesis, we will also refer to a strategy profile as simply having *ε-strong game-theoretic security* (omitting the t-robustness), in which case we implicitly let $t = 1$.

Chapter 3

Commerce

"*When goods do not cross borders, soldiers will.*"

Frederic Bastiat

A FUNDAMENTAL PROBLEM of electronic commerce is ensuring both ends of the trade are upheld: an honest seller should always receive payment, and an honest buyer should only pay if the seller was honest. Traditionally, this is ensured by introducing a trusted intermediary who holds the payment in escrow until the trade has completed, after which it releases the funds to the seller. It typically requires the agents not to be anonymous, to enable either agent to hold the other agent accountable in case of fraudulent behavior, and potentially subject to legal repercussions. This, in conjunction with reputation systems, has proved to be an effective means to honest and efficient trading, as evidenced by the enormous market cap of online marketplaces such as Amazon or Alibaba. However, this relies on being able to trust the intermediary to behave honestly: while the intermediary has a strong incentive to maintain a good reputation, this does not address the fundamental issue from a cryptographic point of view. Besides obvious privacy concerns, a central marketplace also has an incentive to engage in monopolistic behavior, such as removal of competitors' products or differential pricing based on customer demographics to the extent that it remains undetected [133].

Recent years have seen the creation of darknet markets that take advantage of cryptocurrency and mix networks to provide decentralized and somewhat anonymous trade of goods and services. They arguably solve some issues with central marketplaces, but in doing so, also enable black market/criminal activity to remain relatively unchecked. The most infamous darknet market was "Silk Road", known for selling illicit goods such as drugs, weapons, and fake passports. It operated from February 2011 until the authorities seized it in October 2013, and the developer, Ross Ulbricht, sentenced to double life imprisonment. But this is a rarity: because of the anonymous nature of the markets, it is often difficult to prosecute individuals, and many convictions of

buyers are based on circumstantial metadata such as credit card transactions purchasing cryptocurrency of similar size. However, most darknet markets remain inherently centralized, in that all data and escrowed funds are processed directly by the market itself, essentially at its mercy. It requires buyers and sellers to trust both the benevolence and competence of the market, a trust which is at best misplaced and at worst disastrous in consequence. There are many examples of prominent darknet markets being hacked, and all funds held in escrow stolen, or of the operators of the market themselves performing an *exit scam*, i.e. suddenly stealing the funds in escrow and subsequently closing the market. It is often difficult, if not impossible, to recover the stolen funds and hold anyone accountable [90].

In this chapter, we describe a system for facilitating electronic commerce in a decentralized manner. We consider a seller who wants to sell an item to a buyer using the blockchain as a trusted third agent. Specifically, we assume both agents are rational and have shared access to a blockchain that allows them to deploy smart contracts that can exchange and process cryptocurrency. Our goal is to replace the trusted third agent with a smart contract, such that agents can be trusted to complete their end of the trade. As the agents are rational, we want to prove they maximize their utility by behaving as we intend them to.

Attribution. The chapter is based on the paper [222], though most of the text is taken from the full version [221] (including the introduction) with only minor modifications (proof-reading and formatting). Contracts 3.1 and 3.3 and Fig. 3.1 were redrawn. In addition, few of the proofs have been rephrased and rewritten.

Our Results

We propose a smart contract for the escrow of funds that enables any two agents to engage in the trade of a physical good or service for cryptocurrency. The contract relies on a *dispute resolution system* that is invoked only in the case of a dispute. The purpose of the dispute resolution system is to distinguish the honest agent from the dishonest agent. Either agent may issue a dispute by making a "wager" of size λ that they will win the adjudication: the winner is repaid their deposit as well as the funds held in escrow. We prove that both buyer and seller are incentivized to behave honestly if and only if the dispute resolution system is biased in favor of honest agents. Specifically, let γ be the "error rate" of the dispute resolution system: then we show there is a value of λ such that the contract has strong game-theoretic security if and only if $\gamma < \frac{1}{2}$. This is not a particularity of our contract: we show this is inherent to any contract that achieves game-theoretic security for interesting trades. By instead considering a weaker notion of security where agents do not have strict incentives to behave honestly, we can use a random coin flip protocol as

the dispute resolution system that can be implemented under computational assumptions with the use of Blum's coin toss protocol.

The contract can be run on any blockchain that supports smart contracts (such as Ethereum). As a result, many properties (anonymity, efficiency, etc.) of the contract are inherited from the corresponding blockchain. We feature a discussion of different ways to instantiate the smart contract. In particular, the contract can be used in a manner that complies with current laws and regulations by using a blockchain with revocable anonymity: an agent who takes part in distributing illicit goods can be deanonymized by the courts, while all other agents remain anonymous. This would allow for a kind of certification or blueprint of marketplaces based on smart contracts even if they are essentially anonymous, so long as the underlying blockchain uses revocable anonymity.

Related Work

A variety of solutions have been proposed for replacing the trusted third agent with a smart contract in so-called atomic swaps. Most academic work has focused on digital goods, the delivery of which can be deterministically determined under computational assumptions on the participating agents.

Dziembowski, Eckey, and Faust [96] propose a protocol, called *FairSwap* , with essentially optimal security: the goods are delivered to the buyer if and only if the seller receives the money. Their solution relies on cryptography and assumes the goods can be represented as a finite field element. As a result, their protocol does not apply in any meaningful way to physical goods. It seems unlikely we can achieve this notion of security for non-digital goods due to a fundamental difference between the physical and the digital world. Asgaonkar and Krishnamachari [11] propose a smart contract for the trade of digital goods: both agents deposit funds *a priori* (a *dual-deposit*) which is only refunded if the trade was successful. They prove that the honest strategy is the unique subgame perfect equilibrium for sufficiently large deposits. Like FairSwap, their solution only works for digital goods as it requires a hash function to verify the delivery of the item. Witkowski, Seuken, and Parkes [255] consider the setting of escrow in online auctions. Their idea is to pay some of the buyers a rebate to offset their expected loss from engaging in a transaction with the seller. Whether a buyer is paid a rebate depends on the reports of other buyers. They prove that the seller has a strict incentive to be honest, while the buyers are only weakly incentivized to do so. They show that strict incentives for the buyers are possible if the escrow has distributional knowledge about the variations in seller abilities, based on a *peer prediction* method. Unfortunately, their solutions rely on a somewhat idealized setting in which there are many buyers concurrently transacting with the same seller, as otherwise buyers and/or sellers may have an incentive to collude, thus breaking

security. Besides, it is not obvious how to apply their work to a non-auction setting.

Outside academic circles, there are several proposed solutions, of which the most promising are Kleros [169, 170] and OpenBazaar [8]. They are both blockchain-based and as such provide some level of decentralization. Unfortunately, the dispute resolution of OpenBazaar remains centralized in a sense, since all moderation is done by an agreed-upon moderator, requiring both buyer and seller to trust the moderator. From a cryptographic point-of-view, this only serves to move the problem of having to trust the seller to having to trust the moderator. The dispute resolution of Kleros is more sophisticated, in that adjudication is done by a decentralized court where jurors can opt-in on a case-by-case basis. Jurors who vote in accordance with the majority decision are rewarded with money, while jurors who vote differently are penalized. We will study this mechanism in more detail in Chapter 4. Lesaege, George, and Ast [170] argue security by the use of *focal points* [219], defined as the strategy people choose in the absence of communication: jurors will act honestly because they expect other jurors to do so. Unfortunately, no empirical study of Kleros has been published, so whether the focal point of Kleros is "truth" remains conjecture at this point. Besides, neither system has any formal analysis of correctness or security and thus fall short in rigorously solving the buyer and seller's dilemma. To the best knowledge of the author, there is no "truly decentralized" market with game-theoretic security at the time of writing.

3.1 The Basic Contract

In this section, we describe our contract for the trade of non-digital goods and services. We consider a buyer B who wants to purchase an item *it* from a seller S. The item can be a physical good or a service. The item is sold for a price of x, and has a "perceived value" to the buyer of $y > x$, while the seller perceives the value at $x' < x$. From a game-theoretic point of view, we have to assume $y > x > x'$, as otherwise neither buyer nor seller has incentive to engage in the transaction. Before describing our contract, we first describe the related contract by Asgaongkar and Krishnamachari. The contract assumes *it* is a digital item, in the sense that *it* can be passed as input to a hash function $H(\cdot)$.

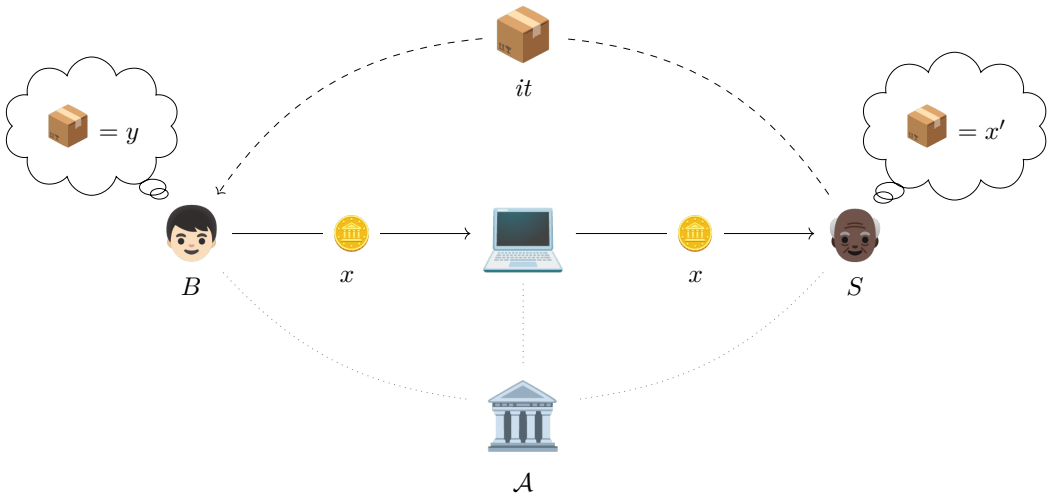

Figure 3.1: Workflow of the basic contract. The item *it* being sold by the seller S to the buyer B for a price of x. The buyer has a perceived value of *it* of y, while the seller has a perceived value of x'. Here, $y > x > x' > 0$. The arbiter $\mathcal{A}$ is only invoked in case of disputes.

Contract 3.1.

1. *S makes public $h \leftarrow H(it)$ where H is a hash function, and deposits λ money to the contract.*
2. *B deposits $x + \lambda$ money to the contract.*
3. *S either submits it to the contract, or submits $it' \neq it$[a].*
4. *B either **accepts** or **rejects** delivery of the item.*
   - 4.1. *If B accepted, λ money is given to B and $x + \lambda$ money given to S, and the contract terminates.*
   - 4.2. *If B rejected, the contract recomputes $H(it)$ and compares with h.*
     - 4.2.1. *If equal, it forwards $x + \lambda$ money to S.*
     - 4.2.2. *If unequal, it forwards $x + \lambda$ money to B.*

[a] An alternative implementation involves encrypting *it* to keep *it* hidden from the smart contract.

Theorem 3.2 (Asgaonkar, Krishnamachari [11])**.** *For any $\varepsilon \geq 0$, and any sufficiently large $\lambda > 0$, Contract 3.1 has ε-strong game-theoretic security.*

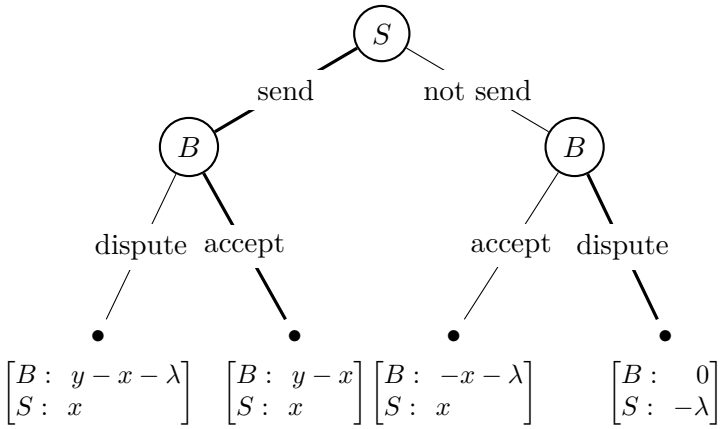

Figure 3.2: The commerce contract by Asgaongkar and Krishnamachari represented as an extensive-form game. First, the seller chooses whether or not to send the item to the buyer. This action is then observed by the buyer who chooses whether or not to accept delivery. If they reject delivery, they retain their deposit only if the seller did not send the item. Otherwise, the seller loses their deposit. The heavy edges denote the subgame perfect equilibrium whenever $\lambda > 0$. In [11], the agents also have the option of submitting garbage to the contract, however this is inconsequential to the analysis and has been removed for brevity.

Non-Digital Commerce.

In the following, let us instead assume the item *it* is *non-digital* which means it has to be shipped through a physical channel "off-chain" and thus eludes Contract 3.1. See Fig. 3.1 for an illustration. By definition, no computer program can rigorously determine whether or not *it* was physically delivered to the buyer. This is a fundamental difference between the digital and the physical world. We assume both agents have access to a blockchain, which for our purposes is a shared data structure that allows both agents to deploy a smart contract π that can maintain state, respond to queries, and transfer funds. Unlike a human third agent, the smart contract can be guaranteed to behave honestly due to the security of the underlying blockchain. For simplicity, we assume the blockchain is secure and incorruptible, and consider only attacks on the contract itself. For now, we assume transaction fees are negligible compared to the items being transacted, such that they can be disregarded entirely. We will dispense with this assumption later.

The contract is parameterized by a *dispute resolution system* $\mathcal{A}$ which is a protocol invoked in case of disputes: its purpose is to distinguish the honest agent from the dishonest agent. We denote by γ the error rate of the dispute resolution system. In the case of digital goods, with computational assumptions on the agents involved, cryptography allows us to get $\gamma = 2^{-\kappa}$ for

any κ which has been exploited in previous work [11, 96]. In Chapter 4, we show how to implement a dispute resolution system using rational agents that interact with a blockchain.

The naïve solution is to invoke the dispute resolution system at every transaction to determine whether or not the seller should be paid. However, this is impractical because invoking the dispute resolution system is potentially expensive; we desire a solution that only invokes the dispute resolution system when necessary, a so-called 'optimistic' contract. We parameterize the contract by a *wager constant* $\lambda > 0$. The contract proceeds as follows: both agents sign a contract committing to making the trade, and B places x money in escrow. S then delivers *it* to B "off-chain" who then notifies the smart contract to transfer the funds in escrow to S, thus terminating the contract. If S does not deliver *it* to B, then B can trigger a dispute by placing a "wager" of size λ that they can convince the dispute resolution system that they were the honest agent. If S does not respond (or forfeits), it is assumed *it* was not delivered to B and the contract refunds $x + \lambda$ funds to B. However, a dishonest buyer may trigger the dispute phase even when they received *it*. In this case, the honest S may counter the wager by also placing a wager of size λ that they will win the adjudication. Of course, a dishonest S may also counter the wager. If both agents counter, the dispute resolution system is invoked and chooses a winner among them. The winner is repaid $x + \lambda$, while the loser receives nothing. We can use the leftover λ to compensate the dispute resolution system for their time. We handle crashing by having timeouts in the contract in a way that favors the agent that did not crash; a buyer that crashes is assumed to have received *it*. Likewise, a seller who fails to respond to a dispute is assumed to forfeit. A full description of the contract is given in Contract 3.3.

Contract 3.3.

1. *B submits x money to the smart contract.*
2. *S sends* it *to B (off-chain).*
3. *B either* ***accepts*** *or* ***rejects*** *delivery, in which case they deposit λ money.*
   - 3.1. *If B accepts, x money is forwarded to S and the contract terminates.*
   - 3.2. *If B rejects, S can either* ***forfeit*** *or* ***counter*** *the dispute, in which case they also deposit λ money.*
     - 3.2.1. *If S forfeits, $x + \lambda$ money is returned to B.*
     - 3.2.2. *If S disputs, the oracle is invoked. Whomever is deemed honest by the oracle receives back $x + \lambda$ money.*

Analysis of Contract

To analyze Contract 3.3 from a game-theoretic perspective, we consider it as an extensive-form game and draw the corresponding game tree (seen in Fig. 3.3). The payoff for each agent is defined as their expected change in funds, where for simplicity we have explicitly omitted transaction fees. As an example, consider a dispute between a dishonest buyer and an honest seller. The buyer has earned y value since the seller was honest. The buyer may lose the adjudication with probability $1 - \gamma$, in which case they lose $x + \lambda$, for an expected payoff of $y - (x + \lambda)(1 - \gamma)$. Likewise, the seller receives their payment of x with probability $1 - \gamma$ and loses λ with probability γ, for an expected payoff of $x\,(1 - \gamma) - \lambda\gamma - x'$. The other cases are similar and are summarized in Fig. 3.3.

Theorem 3.4. *There is a value of λ such that Contract 3.3 has ε-game-theoretic security for some $\varepsilon > 0$ if and only if $\gamma < \frac{1}{2}$ and $\varepsilon \leq x\,(1 - 2\gamma)$.*

Proof. We proceed using backwards induction in the game tree. We see that the honest actions an ε larger payoff if and only if the following inequalities are satisfied:

$$x\,(1 - \gamma) - \lambda\gamma - x' - \varepsilon \geq -x' \tag{3.1}$$

$$-\varepsilon \geq x\gamma - \lambda\,(1 - \gamma) \tag{3.2}$$

$$y - x - \varepsilon \geq y - (x + \lambda)(1 - \gamma) \tag{3.3}$$

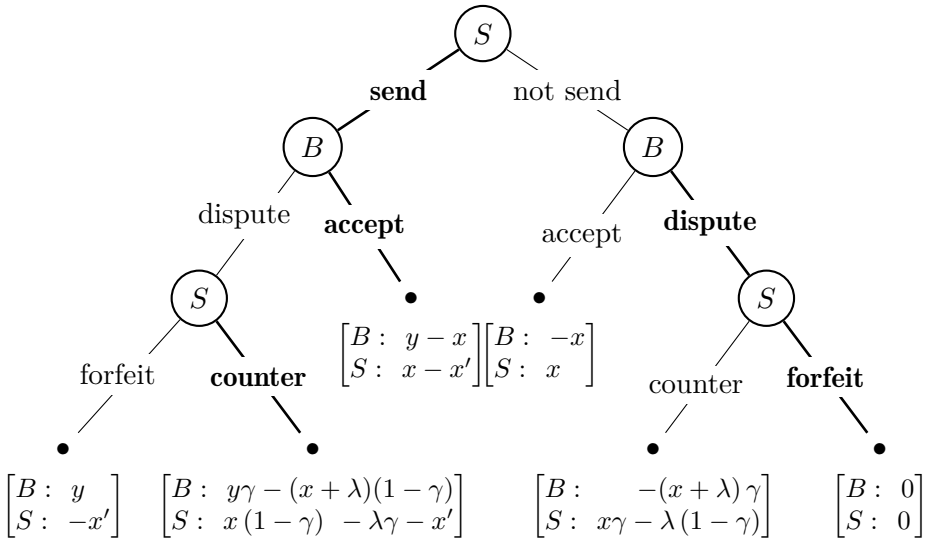

Figure 3.3: Game tree of the smart contract after both agents have accepted the transaction. The seller first chooses whether or not to send the item or not which is observed by the buyer. The buyer can then accept delivery of the item or stake money to raise a dispute. In this case, the seller is given the option to forfeit, in which case all money held in escrow is repaid to the buyer, or they can counter the dispute by also putting money at stake. In this case, we invoke a dispute resolution system to determine who is the honest agent. The honest strategy profile is denoted using bold edges.

Solving for λ, Eq. (3.1) gives $\lambda \leq \frac{x(1-\gamma)-\varepsilon}{\gamma}$. By Eq. (3.2) we get $\lambda \geq \frac{x\gamma+\varepsilon}{1-\gamma}$, while Eq. (3.3) is equivalent to Eq. (3.2). In summary, ε must satisfy:

$$\frac{x\gamma+\varepsilon}{1-\gamma} \leq \lambda \leq \frac{x(1-\gamma)-\varepsilon}{\gamma}$$

We must have $\gamma < \frac{1}{2}$ since $\varepsilon > 0$, while the latter condition can be established by solving for ε. □

Corollary 3.5. *Contract 3.3 has $x(1-2\gamma)$-strong game-theoretic security whenever $\gamma < \frac{1}{2}$ and $\lambda = x$.*

3.2 The Generalized Contract

In this section, we consider a generalization of the previous contract that allows us to obtain various tradeoffs between security and wager size. We show that the bound of $\gamma < \frac{1}{2}$ is inherent to any contract that achieves game-theoretic security for 'interesting trades'. We define an *interesting trade* as a trade where agents have a net increase in utility if they are successful in cheating, compared

to being honest. If a trade is not interesting, security is trivial. Intuitively, this necessitates the use of some mechanism $\mathcal{A}$ that determines if an agent was dishonest. A rational dishonest agent will never reveal themselves if they lose utility by doing so, so $\mathcal{A}$ needs to be external to the agents in the protocol. We call such a mechanism an *dispute resolution system*. We can assume the dispute resolution system is only invoked if one of the agents were dishonest, and we will assume $\mathcal{A}$ outputs a single bit determining whether a fixed agent (say the seller) were the dishonest agent. If an agent is deemed dishonest by the dispute resolution system, we say they are the 'winner'; otherwise they are the 'loser'. We let γ be the error rate of $\mathcal{A}$, i.e. the probability that the loser were honest. Let ω, ℓ be functions such that the winner is paid ω money and the loser is paid ℓ money. We have to assume that $\omega > \ell$ such that winning is preferred over losing. Now, consider a seller who has to decide whether or not to counter a dispute from the buyer. Regardless of whether the seller is honest or not, they have to decide whether to forfeit or counter. If the seller is honest we want them to counter the dispute, i.e. $\omega(1-\gamma) + \ell\gamma > 0$. If the seller is dishonest we want them to forfeit, i.e. $\omega\gamma + \ell(1-\gamma) < 0$. That is,

$$\omega\gamma + \ell(1-\gamma) < \omega(1-\gamma) + \ell\gamma.$$

Since we have $\omega > \ell$ this can only be true for $\gamma < \frac{1}{2}$.

Affine Rebate Functions

In the following we let α be a constant such that the winner is paid back $\alpha\lambda$ money. Naturally, we must have that $\alpha \geq 0$ as the winner cannot lose more than they have wagered. Also, we must have $\alpha \leq 2$ to prevent the contract from minting money. Note that the original contract is a special case where $\alpha = 1$.

Theorem 3.6. *Contract 3.3 has (maximal) ε-strong game-theoretic security if and only if $\gamma < \frac{1}{2}$ and one of the following conditions are established:*

1. $\alpha = 2$*; and* $\lambda \geq \frac{x\gamma+\varepsilon}{1-2\gamma}$.
2. $\frac{1}{1-\gamma} < \alpha < 2$*; and* $\varepsilon \geq x\left(\frac{1-2\gamma}{2-\alpha}\right)$*; and* $\lambda \geq \frac{x\gamma+\varepsilon}{1-\alpha\gamma}$.
3. $\alpha = \frac{1}{1-\gamma}$*; and* $\varepsilon = x(1-\gamma)$*; and* $\lambda = x\left(\frac{1-\gamma}{1-2\gamma}\right)$.
4. $\alpha < \frac{1}{1-\gamma}$*; and* $\varepsilon = x\left(\frac{1-2\gamma}{2-\alpha}\right)$*; and* $\lambda = x\left(\frac{1}{2-\alpha}\right)$.

Proof. As in the proof of Theorem 3.4, we only need to consider a seller faced with a dispute, as this implies the other cases. This leads to the following two

inequalities:

$$(x + \alpha\lambda)(1 - \gamma) - \lambda - \varepsilon \geq 0 \tag{3.4}$$

$$-\varepsilon \geq (x + \alpha\lambda)\gamma - \lambda \tag{3.5}$$

We now consider different values of α, choose the maximum permitted value of ε to maximize security, and solve for λ. Note that we are intentionally leaving out some cases where ε is small.

1. When $\alpha = 2$, Eq. (3.5) gives a lower bound of $\lambda \geq \frac{x\gamma+\varepsilon}{1-2\gamma}$, while as in the proof for completeness Eq. (3.4) gives a trivial lower bound.

2. When $\frac{1}{1-\gamma} < \alpha < 2$, equations Eq. (3.4) and Eq. (3.5) give the following lower bounds.

$$\lambda \geq \frac{x\gamma + \varepsilon}{1 - \alpha\gamma}, \qquad \lambda \geq \frac{x(1-\gamma) - \varepsilon}{1 - \alpha + \alpha\gamma}.$$

   When $\varepsilon \geq x\left(\frac{1-2\gamma}{2-\alpha}\right)$ the lower bound from Eq. (3.4) is strongest. Since we choose the *maximal* value of ε, i.e. not $\varepsilon < x\left(\frac{1-2\gamma}{2-\alpha}\right)$, this gives the desired bound.

3. When $\alpha = \frac{1}{1-\gamma}$, Eq. (3.4) gives an upper bound on the security parameter $\varepsilon \leq x(1-\gamma)$. Similarly, Eq. (3.5) gives a lower bound of $\lambda \geq \frac{(1-\gamma)(x\gamma+\varepsilon)}{1-2\gamma}$. Choosing the maximum $\varepsilon = x(1-\gamma)$ and substituting gives the desired result.

4. When $\alpha < \frac{1}{1-\gamma}$, Eq. (3.4) gives an upper bound of $\lambda \leq \frac{x(1-\gamma)-\varepsilon}{1-\alpha+\alpha\gamma}$, while Eq. (3.5) gives a lower bound of $\lambda \geq \frac{x\gamma+\varepsilon}{1-\alpha\gamma}$. This means there is a value of λ such that ε-soundness is satisfied if and only if,

$$\frac{x\gamma + \varepsilon}{1 - \alpha\gamma} \leq \frac{x(1-\gamma) - \varepsilon}{1 - \alpha + \alpha\gamma}.$$

   The maximal value of ε satisfying this equation is $\varepsilon = x\left(\frac{1-2\gamma}{2-\alpha}\right)$ which solves to $\lambda = \frac{x}{2-\alpha}$, showing the desired. □

Tradeoffs

We now have a characterization of different ways of instantiating Contract 3.3, allowing us to reason about the pros and cons of different choices of parameters. We consider a few special cases: Suppose that, in addition to receiving their own wager back, the winner also receives the loser's wager, i.e. the generalized contract with $\alpha = 2$. By Theorem 3.6, we have no upper bound on ε, allowing us to get arbitrarily high security by making the wager sufficiently large. The

downside to this is that it results in larger wagers: suppose we let $\varepsilon = x\,(1-2\gamma)$, the maximum value in the old contract, then the new wager is:

$$\lambda = \frac{x\gamma + x\,(1-2\gamma)}{1-2\gamma} = x + \frac{x\gamma}{1-2\gamma} > x$$

which is always larger than the old wager. This is natural in a sense: since we expect to win back the wager by disputing, the wager needs to be larger to offset the increased incentive to issue a false dispute.

Corollary 3.7. *With a winner rebate of size λ, Contract 3.3 has ε-strong security (for any $\varepsilon > 0$) whenever $\gamma < \frac{1}{2}$ and $\lambda = \frac{x\gamma+\varepsilon}{1-2\gamma}$.*

Instead, consider what happens if the wager is withheld even for the winning agent, i.e. letting $\alpha = 0$. This naturally requires $\lambda < x$ since otherwise there would be no incentive to dispute. We again refer to Theorem 3.6 which gives the following result:

Corollary 3.8. *When Contract 3.3 withholds all wagers, it has $\frac{1}{2}x\,(1-2\gamma)$-strong game-theoretic security when $\gamma < \frac{1}{2}$, and $\lambda = \frac{1}{2}x$.*

It is not hard to see (referring to Theorem 3.6) that this is the minimal value of λ we can use if we want to maximize the security of the protocol. It seems our construction requires $\lambda = \Omega(x)$. This is natural in a sense: if λ were a constant, by increasing x, at some point, the expected utility from attempting to cheat would outweigh the cost of losing the wager.

Invoking the dispute resolution system is not free. This is the very motivation behind our contract: if the dispute resolution system were free and better than random, we could trivially invoke it in every purchase to determine who should receive the money. However, this unfairly punishes honest agents by requiring them to pay for an expensive and unnecessary adjudication. Worse yet, invoking the arbiter at every interaction would result in false convictions even when both agents are satisfied with the transaction. Still, we have to compensate the dispute resolution system somehow. We can accomplish this by varying α such that the left over funds equal the price of the adjudication. If P is the price of the adjudication, we can accomplish this by letting $\alpha = 2 - P/\lambda$. This works whenever $P > 2\lambda$ such that the total wager exceeds the price of paying for the adjudication. We may instantiate this in various ways by referring to Theorem 3.6.

3.3 Practical Considerations

Transaction fees

Our analysis assumes transaction fees are negligible which is not the case in practice. In this section, we consider adding transaction fees to our model

and show that this incurs a loss of security which is additive in the size of the transaction fee. Doing so in general is tricky business and is specific to the implementation and the blockchain of choice. Instead, we adopt a simplified approach where playing a move in the game tree has a unit cost of τ for some $\tau > 0$, the only exception being the default action in case of timeouts: an agent can always time out to choose the default action at zero cost. For simplicity, we let $\alpha = 1$.

Lemma 3.9. *With transaction fees of size τ, Contract 3.3 has ε-game-theoretic security if and only if,*

1. *the adjudication is biased in favor of honest agents; and,*
2. *the transaction fee is bounded $\tau < x\,(1-\gamma) - \lambda\gamma$; and,*
3. *the item is of sufficient value, $x - x' > \tau$.*

Proof. We proceed using backward induction in the game tree. It is not hard to see we still need $\gamma < \frac{1}{2}$. Consider a seller faced with a dispute. When they are dishonest their incentive to be honest is increased by τ, while the converse is true when they are honest. This yields $\tau < x\,(1-\gamma) - \lambda\gamma - \varepsilon$. Now consider a buyer. If they did receive the item, their incentive to accept has only increased by τ. If they did not receive the item, their added cost of τ for issuing a dispute must outweigh the size of the payment. This means we must have $x > \tau$. Finally, consider a seller deciding whether to send or not. If they do not send they incur a cost of 0, while accepting gives $x - x' - \tau > 0$. □

Corollary 3.10. *With transaction fees of size τ, Contract 3.3 has $[x\,(1-2\gamma)-\tau]$-strong game-theoretic security when $\lambda = x$ and $x - x' > \tau$.*

Denial-of-Service Attacks

We briefly consider malicious sellers that waste the buyer's time and resources by accepting a contract only for it to time out. When there are no transaction fees, this attack is free to deploy and clearly imposes negative utility on the buyer, since their funds are locked until the timeout passes. With transaction fees, the attack is no longer free, though it is still fairly cheap for large purchases. To circumvent this, we can force the seller to deposit to accept the contract. The deposit is paid back when the contract is completed. In this way, the seller can only mount the attack by suffering a similar utility loss as the buyer which a rational seller would not do.

Coin Toss Adjudication

In this subsection, we consider the special case in which the output of the dispute resolution system is independent of the evidence being submitted,

i.e. $\gamma = \frac{1}{2}$. The advantage of this is that we can implement such a dispute resolution system using a cryptographic protocol. However, we showed that strong game-theoretic security is only possible when $\gamma < \frac{1}{2}$ so we need to relax our security definition. For this reason, we say that a protocol enjoys *weak game-theoretic security* if the intended strategy profile is a subgame perfect equilibrium. Note that while this guarantees that being honest is an equilibrium strategy it does not provide a strict incentive to do so. However, it remains secure in a strong sense against risk-averse agents which also means it is strictly insecure against risk-seeking agents.

Theorem 3.11. *Using a coin toss dispute resolution system, Contract 3.3 has weak game-theoretic security for $\gamma = \frac{1}{2}$ and $\lambda = x$.*

Proof. For s^* to be a subgame perfect equilibrium, there must be no $s \neq s^*$ that achieves a strictly larger payoff. As before, this can only be achieved when:

$$x\,(1-\gamma) - \lambda\gamma \geq x\gamma - \lambda\,(1-\gamma) \tag{3.6}$$

which solves to $\lambda = x$ for $\gamma = \frac{1}{2}$. □

We can implement the coin toss dispute resolution system using a variant of Blum's coin-flipping protocol [37, 77]. Suppose we have a commitment scheme, and let commit be the commitment function. Then the adjudication proceeds as follows:

1. S samples a random bit $b_S \in_R \{0,1\}$, and a random string $r \in_R \{0,1\}^\kappa$.
2. S computes $C \leftarrow \mathsf{commit}(b, r)$ and submits C to the smart contract.
3. B samples a random bit $b_B \in_R \{0,1\}$ and submits it to the smart contract.
4. S submits b_S and r to the blockchain.
5. The smart contract verifies that b_S, r is a valid opening of C: if not, let $b := 0$. Otherwise let $b := b_S \oplus b_B$.
6. The smart contract transfers $x + \lambda$ to S if and only if $b = 1$, and transfers $x + \lambda$ to B otherwise.

If at some point either agent times out, it is assumed they forfeited, and the funds held in escrow are released to the other agent. Analysis of the protocol is straightforward: the output is uniform, and security reduces to that of the commitment scheme. From a cryptographic perspective, this protocol is unsatisfactory because it does not satisfy fairness: S can choose not to complete step 4 and simply abort the protocol without revealing the output to B if they are dissatisfied with the result. However, this is not an issue for

our application, since S loses the dispute by doing so. In general, it is hard to achieve a fair coin flip on a blockchain: the best known protocol to date samples $\Theta(n^2)$ fair random values using an amortized $O(\log n)$ exponentiations per value [56].

Conclusion

Chapter 4

Adjudication

> "*The opinion of 10,000 men is of no value if none of them know anything about the subject.*"
>
> Marcus Aurelius (allegedly)

Juries are traditionally used in English common law to assess evidence and provide a neutral judgment on the true state of the world [253]. Historically, juries were composed of entrusted members of society whose alleged virtuousness compelled them to vote impartially in accordance with community norms [214]. The idea is that even if one juror occasionally makes errors, collectively the jury tends to produce more correct outcomes than either juror member would on their own [83]. In recent years, there has been interest in developing these jury systems for use in distributed computing to establish consensus on data whose validity inherently cannot be verified by a computer [169, 170, 201]. The main complication is that Web3 typically assumes no trusted authorities and adjudication must therefore be delegated to ordinary users (or agents), who are appointed as jurors and get compensated for this activity. Such agents are largely anonymous and cannot easily be held accountable for their actions [93]. They are largely indifferent to the outcome of the adjudication case and typically strategize to maximize their utility. As such, paying a fixed reward to the agents for their participation is insufficient; they can then just vote randomly, without putting in any effort to assess the case evidence, producing a useless adjudication outcome. Instead, to produce a non-trivial adjudication, payments to/from the agents should be in some way conditioned on their vote. Hopefully, if the agents are satisfied with their payments, they will make a reasonable effort to assess the case evidence and collectively come up with a correct adjudication.

In this chapter, we consider binary (yes/no) adjudication tasks and the following simple mechanism. Each agent submits a vote with their opinion and the adjudication outcome is decided using majority. Agents are rewarded for voting in accordance with the final verdict and less so for voting otherwise (this

approach is used in practice, e.g. in Kleros [169, 170] – it is a dispute resolution system which is deployed on Ethereum and, at the time of writing, it has allegedly settled more than one thousand disputes. We consider the problem of incentivizing jurors to properly assess case evidence so that the resulting adjudication is better than random. The problem is motivated by dispute resolution in Web3 systems, where a reliable solution would find numerous applications in, e.g., supply chain management, banking, and commerce, such as the system we proposed in Chapter 3.

Attribution. This chapter (including introduction) is taken (almost) verbatim from [54], with only minor modifications to the formatting and the prose. Fig. 4.2 and the accompanying text is taken with no changes from the full version [53] of the paper.

Our Contributions

Our main conceptual contribution is a new model for the behavior of strategic agents. The model aims to capture the two important components of strategic behavior while participating in an adjudication task. The first one is to decide the effort the agent needs to exert to get sufficient understanding of the task and form their opinion. The second one is whether they will cast this opinion as their vote or they will vote for the opposite alternative. We assume that, when dealing with an adjudication task, agents do not communicate with each other. Instead, each of them has access to the outcome of similar tasks from the past. An agent can compare these outcomes to their own reasoning for them, which allows them to conclude whether their background knowledge is positively correlated, negatively correlated, or uncorrelated to the votes cast by the other agents. Payments can be used to amplify the agent's incentive to take such correlation into account. A strategic agent then acts as follows. If there is a positive correlation, their opinion for the new adjudication task will be cast as their vote. If the correlation is negative, they will cast the opposite vote. If there is no correlation, the agent will vote randomly. We assume that each adjudication task has a ground truth alternative that we wish to recover. Agents are distinguished into well-informed and misinformed ones. Well-informed (respectively, misinformed) agents are those whose opinions get closer to (respectively, further away from) the ground truth with increased effort. The ground truth is unobservable and, thus, the agents are not aware of the category to which they belong.

After presenting the strategic agent model, we characterize the strategies of the agents at equilibria of the induced game. We use this characterization to identify a sufficient condition for payments so that equilibria are simple, in the sense that the agents either vote randomly or they are all biased towards the same alternative. Next, we focus on a simple scenario with a population of well-informed and misinformed agents with complementary effort functions and

show how to efficiently find payments that result in adjudication that recovers the ground truth with a given probability. Finally, we conduct experiments to justify that strategic play of a population with a majority of well-informed agents results in correct adjudication when payments are set appropriately.

Related Work

Voting, the main tool we use for adjudication, has received enormous attention in the social choice theory literature — originating with the seminal work of Arrow [9] — and its recent computational treatment [43]. However, the main assumption here is that agents have preferences about the alternatives and thus an interest in the voting outcome, in contrast to our case where agents' interest for the final outcome depends only on whether this gives them compensation or not. Strategic voter behavior is well-known to alter the intended outcome of all voting rules besides two-alternative majority voting and dictatorships [113, 216]. Positive results are possible with the less popular approach of introducing payments to the voting process; e.g., see [205].

The assumption for a ground truth alternative has been also inspired by voting theory [55, 70, 262]. In a quite popular approach, votes are considered as noisy estimates of an underlying ground truth; typically, agents tend to inherit the preferences in the ground truth more often than the opposite ones. Our assumption for a majority of well-informed agents is in accordance with this. However, an important feature here is that the ground truth is unobservable. This is a typical assumption in the area of peer prediction mechanisms for unverifiable information (see [101], Chapter 3), where a set of agents are used to decide the quality of data. However, that line of work has a mechanism design flavor and assumes compensations to the agents so that their evaluation of the available data is truthful (e.g., see [256]). This is significantly different from our modeling assumptions here. In particular, any evaluation of the quality of the agents — a task that is usually part of crowdsourcing systems; e.g., see [226] — is in our case infeasible. Still, our payment optimization is similar in spirit to automated mechanism design [215] but, instead of aiming for truthful agent behavior, we have a particular equilibrium as a target.

Recent work by Michelini, Haret, and Grossi [187] also considers a model where jurors can exert varying effort to obtain better or worse signals. They relate the effort exerted by an agent to their interest in producing a correct outcome and find that only when the agents care sufficiently about the outcome, the equilibrium produces a correct outcome with high probability. This is significantly different from our model where the agents are assumed to be indifferent to the outcome and are instead motivated by the payments they receive.

4.1 Modeling Assumptions

We assume that adjudication tasks with two alternatives are outsourced to n agents. We use the integers in $[n] = \{1, 2, ..., n\}$ to identify the agents. For an adjudication task, each agent casts a vote for one of the alternatives and the majority of votes defines the adjudication outcome. In the case of a tie, an outcome is sampled uniformly at random. To motivate voting, payments are used. A *payment function* $p : [0, 1] \to \mathbb{R}$ indicates that agent i gets a payment of $p(x)$ when the total fraction of agents casting the same vote as i is x. Payments can be positive or negative (corresponding to monetary transfers to and from the agents, respectively).

The objective of an adjudication task is to recover the underlying *ground truth*. We denote by T the ground truth and by F the other alternative. We use the terms T-vote and F-vote to refer to a vote for alternative T and F, respectively. To decide which vote to cast, agents put an effort to understand the adjudication case and get a *signal* of whether the correct adjudication outcome is T or F. We partition the agents into two categories, depending on whether their background knowledge is sufficient so that the quality of the signal they receive increases with extra effort (*well-informed* agents) or worsens (*misinformed* agents). Each agent i is associated with an *effort function* $f_i : \mathbb{R}_{\geq 0} \to [0, 1]$ which relates the quality of the signal received by an agent with the effort they exert as follows: the signal agent i gets when they exert an effort $x \geq 0$ is for the ground truth alternative T with probability $f_i(x)$ and for alternative F with probability $1 - f_i(x)$. We assume that effort functions are continuously differentiable and have $f_i(0) = 1/2$. The effort function for a well-informed agent i is strictly increasing and strictly concave. The effort function for a misinformed agent is strictly decreasing and strictly convex. The functions $f_i(x) = 1 - \frac{e^{-x}}{2}$ and $f_i(x) = \frac{e^{-x}}{2}$ are typical examples of effort functions for a well-informed and a misinformed agent, respectively.

Agents are rational. They are involved in a *strategic game* where they aim to maximize their utility, consisting only of the payment they receive minus the effort they exert. In particular, we assume the agents are entirely indifferent to the outcome. This may lead to voting differently than what their signal indicates. We denote by (λ_i, β_i) the *strategy* of agent i, where λ_i is the effort put and β_i is the probability of casting a vote that is identical to the signal received (and, thus, the agent casts a vote for the opposite alternative with probability $1 - \beta_i$). The utility of an agent is *quasi-linear*, i.e., equal to the amount of payments received minus the effort exerted. We assume that agents are *risk neutral* and thus aim to maximize the expectation of their utility. Denote by m_i the random variable indicating the number of agents different than i who cast a T-vote. Clearly, m_i depends on the strategies of all agents besides i but, for simplicity, we have removed this dependency from our

notation. Now, the expected utility of agent i when using strategy (λ_i, β_i) is

$$\begin{aligned}
&\mathbb{E}[u_i(\lambda_i, \beta_i, m_i)] \\
&= -\lambda_i + f_i(\lambda_i)\beta_i \cdot \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] + (1 - f_i(\lambda_i))\beta_i \cdot \mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right] \\
&\quad + f_i(\lambda_i)(1-\beta_i) \cdot \mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right] + (1 - f_i(\lambda_i))(1-\beta_i) \cdot \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] \\
&= -\lambda_i + \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] + (\beta_i(2f_i(\lambda_i) - 1) - f_i(\lambda_i)) \cdot Q(m_i). \qquad (4.1)
\end{aligned}$$

The quantities $p\left(\frac{1+m_i}{n}\right)$ and $p\left(\frac{n-m_i}{n}\right)$ are the payments agent i receives when they votes for alternatives T and F, respectively. The four positive terms in the RHS of the first equality above are the expected payments for the four cases defined depending on the signal received and whether it is cast as a vote or not. In the second equality, we have used the abbreviation

$$Q(m_i) = \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right) - p\left(\frac{n-m_i}{n}\right)\right],$$

which we also use extensively in the following. Intuitively, given the strategies of the other agents, $Q(m_i)$ is the additional expected payment agent i gets when casting a T-vote compared to an F-vote.

We say that a set of strategies, in which agent $i \in [n]$ uses strategy (λ_i, β_i), is an *equilibrium* in the strategic game induced, if no agent can increase their utility by unilaterally changing their strategy. In other words, the quantity $\mathbb{E}[u_i(x, y, m_i)]$ is maximized with respect to x and y by setting $x = \lambda_i$ and $y = \beta_i$ for $i \in [n]$.

4.2 Equilibrium Analysis

We are now ready to characterize equilibria. We remark that cases (a), (b), and (c) of Lemma 4.1 correspond to the informal terms no correlation, positive correlation, and negative correlation used in the introductory section.

Lemma 4.1 (equilibrium conditions)**.** *The strategy of agent i at equilibrium is as follows:*

(a) If $|f_i'(0) \cdot Q(m_i)| \leq 1$, then $\lambda_i = 0$ and β_i can have any value in $[0, 1]$.

(b) If $f_i'(0) \cdot Q(m_i) > 1$, then λ_i is positive and such that $f_i'(\lambda_i) \cdot Q(m_i) = 1$ and $\beta_i = 1$.

(c) If $f_i'(0) \cdot Q(m_i) < -1$, then λ_i is positive and such that $f_i'(\lambda_i) \cdot Q(m_i) = -1$ and $\beta_i = 0$.

Proof. First, observe that when agent i selects $\lambda_i = 0$, their expected utility is

$$\mathbb{E}(u_i(0, \beta_i, m_i)] = \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] - \frac{1}{2}Q(m_i),$$

i.e., it is independent of β_i. So, β_i can take any value in $[0, 1]$ when $\lambda_i = 0$.

In case (b), we have $f_i'(0) \cdot Q(m_i) > 0$ which, by the definition of the effort function f_i, implies that $(2f_i(\lambda_i) - 1) \cdot Q(m_i) > 0$ for $\lambda_i > 0$. By inspecting the dependence of expected utility on β_i at the RHS of Eq. (4.1), we get that if agent i selects $\lambda_i > 0$, they must also select $\beta_i = 1$ to maximize their expected utility in this case. Similarly, in case (c), we have $f_i'(0) \cdot Q(m_i) < 0$ which implies that $(2f_i(\lambda_i) - 1) \cdot Q(m_i) < 0$ for $\lambda_i > 0$. In this case, if agent i selects $\lambda_i > 0$, they will also select $\beta_i = 0$ to maximize their expected utility.

So, in the following, it suffices to reason only about the value of λ_i. Let

$$\begin{aligned}\Delta_i(\lambda_i) &= \frac{\partial \mathbb{E}[u_i(\lambda_i, \beta_i, m_i)]}{\partial \lambda_i} \\ &= -1 + (2\beta_i - 1)f'(\lambda_i) \cdot Q(m_i) \qquad (4.2)\end{aligned}$$

denote the derivative of the expected utility of agent i with respect to λ_i. In case (a), by the strict concavity/convexity of the effort function f_i we have $|f_i'(\lambda_i) \cdot Q(m_i)| < 1$ for $\lambda_i > 0$ and

$$\begin{aligned}\Delta_i(\lambda_i) &= -1 + (2\beta_i - 1)f_i'(\lambda_i) \cdot Q(m_i) \\ &\leq -1 + |2\beta_i - 1| \cdot |f_i'(\lambda_i) \cdot Q(m_i)| < 0.\end{aligned}$$

Hence, the expected utility of agent i strictly decreases with $\lambda_i > 0$ and the best strategy for agent i is to set $\lambda_i = 0$.

Otherwise, in cases (b) and (c), the derivative $\Delta_i(\lambda_i)$ has strictly positive values for λ_i arbitrarily close to 0 (this follows by the facts that f is strictly convex/concave and continuously differentiable), while it is clearly negative as λ_i approaches infinity (where the derivative of f approaches 0). Hence, the value of λ_i selected by agent i at equilibrium is one that nullifies the RHS of Eq. (4.2), i.e., such that $f_i'(\lambda_i) \cdot Q(m_i) = 1$ in case (b) and $f_i'(\lambda_i) \cdot Q(m_i) = -1$ in case (c). Recall that β_i is equal to 1 and 0 in these two cases, respectively. □

Using Lemma 4.1, we can now identify some properties about the structure of equilibria.

Lemma 4.2. *For any payment function, no effort by all agents (i.e., $\lambda_i = 0$ for $i \in [n]$) is an equilibrium.*

Proof. Notice that, when no agent puts any effort, each vote selects one of the two alternatives equiprobably. Then, the probability that m_i takes a value $t \in \{0, 1, ..., n-1\}$ is equal to the probability that it takes value $n - 1 - t$. Hence, $\mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] = \mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right]$ and $Q(m_i) = 0$. Hence, all agents' strategies satisfy the condition of case (a) of Lemma 4.1 and, thus, $\lambda_i = 0$ is the best response for each agent i. □

We will use the term *non-trivial* for equilibria having at least one agent putting some effort.

The next lemma reveals the challenge of adjudication in our strategic environment. It essentially states that for every equilibrium that yields probably correct adjudication, there is an equilibrium that yields probably incorrect adjudication with the same probability.

Lemma 4.3. *For any payment function, if the set of strategies* $(\lambda_i, \beta_i)_{i\in[n]}$ *is an equilibrium, so is the set of strategies* $(\lambda_i, 1-\beta_i)_{i\in[n]}$.

Proof. With a slight abuse of notation, we reserve the notation m_i for the initial equilibrium where agent i follows strategy $(\lambda_i, \beta_i)_{i\in[n]}$ and denote by m'_i the random variable indicating the number of agents different than i who cast a T-vote in the state where agent i follows strategy $(\lambda_i, 1-\beta_i)_{i\in[n]}$. Notice that, due to symmetry, the probability that m_i gets a given value t is equal to the probability that m'_i gets the value $n-1-t$. Hence,

$$\mathbb{E}\left[p\left(\frac{1+m'_i}{n}\right)\right] = \mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right]$$
$$\text{and } \mathbb{E}\left[p\left(\frac{n-m'_i}{n}\right)\right] = \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right].$$

Thus, $Q(m_i) = -Q(m'_i)$, hence $f'_i(0)\cdot Q(m_i) = -f'_i(0)\cdot Q(m'_i)$. By Lemma 4.1, we have that the strategies of all agents in the new state are consistent with the equilibrium conditions of Lemma 4.1, provided that the initial state is an equilibrium (and thus satisfies the conditions). □

We say that an equilibrium is *simple* if there exists an alternative $a \in \{T, F\}$ such that all agents cast a vote for alternative a with probability at least $1/2$. Intuitively, this makes prediction of the agents' behavior at equilibrium easy. Together with Lemma 4.1, this definition implies that, in a simple equilibrium, an agent putting no effort (i.e., $\lambda_i = 0$) can use any strategy β_i. For agents putting some effort, a well-informed agent uses $\beta_i = 1$ if $a = T$ and $\beta_i = 0$ if $a = F$ and a misinformed agent uses $\beta_i = 0$ if $a = T$ and $\beta_i = 1$ if $a = T$.

Lemma 4.4 (simple equilibrium condition)**.** *When the payment function* p *satisfies,*

$$p\left(\frac{2+m}{n}\right) - p\left(\frac{1+m}{n}\right) + p\left(\frac{n-m}{n}\right) - p\left(\frac{n-m-1}{n}\right) \geq 0, \tag{4.3}$$

for every $m \in \{0, 1, ..., n-2\}$*, all equilibria are simple.*

Proof. For the sake of contradiction, let us assume that the payment function p satisfies the condition of the lemma but, at some equilibrium, agents 1 and 2 cast a T-vote with probability higher than $1/2$ and lower than $1/2$, respectively. Clearly, the equilibrium strategies of agents 1 and 2 cannot belong to case (a) of Lemma 4.1 as the probability of casting a T-vote would be exactly $1/2$ in that case.

We first focus on agent 1 and distinguish between two cases. If their strategy is $\beta_1 = 1$, then it belongs to case (b) of Lemma 4.1 and, thus, $f_1'(0) \cdot Q(m_1) > 1$. Furthermore, the probability of casting a T-vote is $f_1(\lambda)$. Hence, $f_1(\lambda) > 1/2$, implying that agent 1 is well-informed with $f_1'(0) > 0$. By the inequality above, we conclude that $Q(m_1) > 0$. If instead, agent 1's strategy is $\beta_1 = 0$, then it belongs to case (c) of Lemma 4.1 and, thus, $f_1'(0) \cdot Q(m_1) < 1$. The probability of casting a T-vote is now $1 - f_1(\lambda)$. Hence, $f_1(\lambda) < 1/2$, implying that agent 1 is misinformed with $f_1'(0) < 0$. By the inequality involving $Q(m_1)$, we conclude that $Q(m_1) > 0$ again.

Applying the same reasoning for agent 2, we can show that $Q(m_2) < 0$. Hence,

$$Q(m_1) - Q(m_2) > 0. \tag{4.4}$$

Denote by X_1 and X_2 the random variables indicating that agents 1 and 2 cast a T-vote and by m the number of T-votes by agents different than 1 and 2. Let $\delta_i = \Pr[X_i = 1]$. For $i \in \{1, 2\}$, we have

$$\begin{aligned} Q(m_{3-i}) &= \mathbb{E}\left[p\left(\frac{1+m+X_i}{n}\right) - p\left(\frac{n-m-X_i}{n}\right)\right] \\ &= \delta_i \cdot \mathbb{E}\left[p\left(\frac{2+m}{n}\right) - p\left(\frac{n-m-1}{n}\right)\right] \\ &\quad + (1-\delta_i) \cdot \mathbb{E}\left[p\left(\frac{1+m}{n}\right) - p\left(\frac{n-m}{n}\right)\right] \\ &= Q(m) + \delta_i \left(Q(m+1) - Q(m)\right). \end{aligned} \tag{4.5}$$

Hence, from Eqs. (4.4) and (4.5) we obtain that

$$(Q(m+1) - Q(m)) \cdot (\delta_2 - \delta_1) > 0. \tag{4.6}$$

Notice that the assumption on p implies that

$$\begin{aligned} &Q(m+1) - Q(m) \\ &= \mathbb{E}\left[p\left(\frac{2+m}{n}\right) - p\left(\frac{n-m-1}{n}\right)\right] \\ &\quad - \mathbb{E}\left[p\left(\frac{1+m}{n}\right) - p\left(\frac{n-m}{n}\right)\right] \geq 0, \end{aligned}$$

while our assumption on the probability of casting a T-vote implies $\delta_1 > 1/2 > \delta_2$. These last two inequalities contradict Eq. (4.6) and the proof is complete. □

It can be verified that the payment function

$$p(x) = \begin{cases} \frac{\omega}{xn}, & x \geq 1/2 \\ -\frac{\ell}{xn}, & x < 1/2 \end{cases}$$

with $\omega \leq \ell$ satisfies the condition of Lemma 4.4. We refer to this function as the award/loss sharing payment function. Essentially, the agents with the majority vote share an award of ω while the ones in minority share a loss of ℓ. Note that for $\omega = \ell$, the payment function is strictly budget balanced unless all votes are unanimous. This is similar to the payment function used in Kleros. A sufficient condition for simple equilibria which is quite broad but does not include Kleros' payments is the following.

Corollary 4.5. *When the payment functions are monotone non-decreasing, all equilibria are simple.*

4.3 Selecting Payments for Correct Adjudication

We now focus on the simple scenario in which some of the n agents are well-informed and have the same effort function f and the rest are misinformed and have the effort function $1 - f$. Can we motivate an expected x-fraction of them vote for the ground truth?

Of course, we are interested in values of x that are higher than $1/2$. This goal is directly related to asking for a high probability of correct adjudication. Indeed, as the agents cast their votes independently, the realized number of T-votes is sharply concentrated around their expectation and thus the probability of incorrect adjudication is exponentially small in terms of the number of agents n and the quantity $(x - 1/2)^2$. This can be proved formally by a simple application of well-known concentration bounds, e.g., Hoeffding's inequality [136].

So, our aim here is to define appropriate payment functions so that a set of strategies leading to an expected x-fraction of T-votes is an equilibrium. We will restrict our attention to payments satisfying the condition of Lemma 4.4; then, we know that all equilibria are simple. We will furthermore show that all equilibria are *symmetric*, in the sense that all agents cast a T-vote with the same probability. This means that there are $\lambda > 0$ and $\beta \in \{0, 1\}$ so that all well-informed agents use strategy (λ, β) and all misinformed agents use the strategy $(\lambda, 1 - \beta)$.

Lemma 4.6. *Consider the scenario with n agents, among which the well-informed agents use the same effort function f and the misinformed agents use the effort function $1 - f$. If the payment function p satisfies the condition of Lemma 4.4, then all equilibria are symmetric.*

Proof. For the sake of contradiction, assume that non-symmetric equilibria exist. Then, by Lemma 4.3, there exists an equilibrium, in which the agent i putting the highest effort $\lambda_i > 0$ is either well-informed and follows the strategy $(\lambda_i, 1)$ or misinformed and follows the strategy $(\lambda_i, 0)$, casting a T-vote with probability $f(\lambda_i) > 1/2$. Let j be another agent using strategy (λ_j, β_j) with $\lambda_j < \lambda_i$. Since agent i casts a T-vote with probability higher than $1/2$, agent j is either well-informed and uses $\beta_j = 1$ or misinformed and uses $\beta_j = 0$; in any other case, they would cast a T-vote with probability less than $1/2$, contradicting the simplicity of equilibria from Lemma 4.4. In both cases, the probability of casting a T-vote is,

$$f(\lambda_j) < f(\lambda_i). \tag{4.7}$$

Now, denote by m the random variable indicating the number of agents different than i and j who cast a T-vote. Then, it is $m_i = m + 1$ with probability $f(\lambda_j)$ and $m_i = m$ with probability $1 - f(\lambda_j)$. Thus, by the definition of Q, we get,

$$\begin{aligned} Q(m_i) = {} & \mathbb{E}\left[p\left(\frac{2+m}{n}\right)\right] \cdot f(\lambda_j) + \mathbb{E}\left[p\left(\frac{1+m}{n}\right)\right] \cdot (1 - f(\lambda_j)) \\ & - \mathbb{E}\left[p\left(\frac{n-m-1}{n}\right)\right] \cdot f(\lambda_j) - \mathbb{E}\left[p\left(\frac{n-m}{n}\right)\right] \cdot (1 - f(\lambda_j)) \\ = {} & Q(m) \\ & + f(\lambda_j) \cdot \mathbb{E}\left[p\left(\frac{2+m}{n}\right) - p\left(\frac{1+m}{n}\right) + p\left(\frac{n-m}{n}\right) - p\left(\frac{n-m-1}{n}\right)\right], \end{aligned} \tag{4.8}$$

and an analogous equality for $Q(m_j)$. Since, by Lemma 4.4, the expectation is non-negative, Eq. (4.7) implies that,

$$Q(m_i) \leq Q(m_j). \tag{4.9}$$

Now, by the equilibrium condition for agent i, we have $f'(\lambda_i) \cdot Q(m_i) = 1$ (notice that this condition holds, no matter whether agent i is well-informed or misinformed) and, hence,

$$Q(m_i) > 0. \tag{4.10}$$

By the strict concavity of the effort function f and since $\lambda_j < \lambda_i$, we also have that

$$f'(\lambda_i) < f'(\lambda_j). \tag{4.11}$$

Using the equilibrium condition for agent j (again, this holds no matter whether agent j is well-informed or misinformed) and Eqs. (4.9) to (4.11), we obtain,

$$f'(\lambda_i) \cdot Q(m_i) < f'(\lambda_j) \cdot Q(m_j) \leq f'(\lambda_j) \cdot Q(m_j) = 1,$$

which contradicts the equilibrium condition for agent i. □

Lemma 4.6 implies that, for $x > 1/2$, an equilibrium with an expected x-fraction of T-votes has each agent casting a T-vote with probability $f(\lambda) = x$; the well-informed agents use the strategy $(\lambda, 1)$ and the misinformed agents use the strategy $(\lambda, 0)$. As agents vote independently, the random variables m_i follow the same binomial distribution $\text{Bin}(n-1, x)$ with $n-1$ trials, each having success probability x. Also, notice that the fact that the effort function is strictly monotone implies that λ is uniquely defined from x as $\lambda = f^{-1}(x)$.

We now aim to solve the optimization task of selecting a payment function p which satisfies the conditions of Lemma 4.4, induces as equilibrium the strategy $(\lambda, 1)$ for well-informed agents and the strategy $(\lambda, 0)$ for misinformed agents, ensures non-negative expected utility for all agents (individual rationality), and minimizes the expected amount given to the agents as payment. As all agents cast a T-vote with the same probability and the quantities m_i are identically distributed for different is, it suffices to minimize the expected payment

$$x \cdot \mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] + (1-x) \cdot \mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right] \tag{4.12}$$

of a single agent. By the definition of expected utility in Eq. (4.1), restricting this quantity to values at least as high as $f^{-1}(x)$ gives the individual rationality constraints for all agents. Furthermore, by Lemma 4.1, the equation,

$$f'(f^{-1}(x)) \cdot Q(m_i) = 1, \tag{4.13}$$

gives the equilibrium condition for both well-informed and misinformed agents.

We can solve the optimization task above using linear programming. Our LP has the payment parameters $p(1/n)$, $p(2/n)$, ..., $p(1)$ as variables. The linear inequalities (Eq. (4.3)) for $m \in \{0, 1, ..., n-2\}$ form the first set of constraints, restricting the search to payment functions satisfying the conditions of Lemma 4.4. Crucially, observe that the quantities $\mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right]$ and $\mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right]$ and, subsequently, $Q(m_i)$, can be expressed as linear functions of the payment parameters. Indeed, for $t = 0, 1, ..., n-1$, let $z(t) = \Pr[m_i = t]$ be the known probabilities of the binomial distribution $\text{Bin}(n-1, x)$. Clearly,

$$\mathbb{E}\left[p\left(\frac{1+m_i}{n}\right)\right] = \sum_{t=0}^{n-1} z(t) \cdot p\left(\frac{1+t}{n}\right),$$

and,

$$\mathbb{E}\left[p\left(\frac{n-m_i}{n}\right)\right] = \sum_{t=0}^{n-1} z(t) \cdot p\left(\frac{n-t}{n}\right).$$

Thus, the objective function (Eq. (4.12)), the individual rationality constraint, and the equilibrium condition constraint can be expressed as linear functions of the LP variables. Overall, the LP has n variables and $n+1$ constraints (n inequalities and one equality). The next statement summarizes the above discussion.

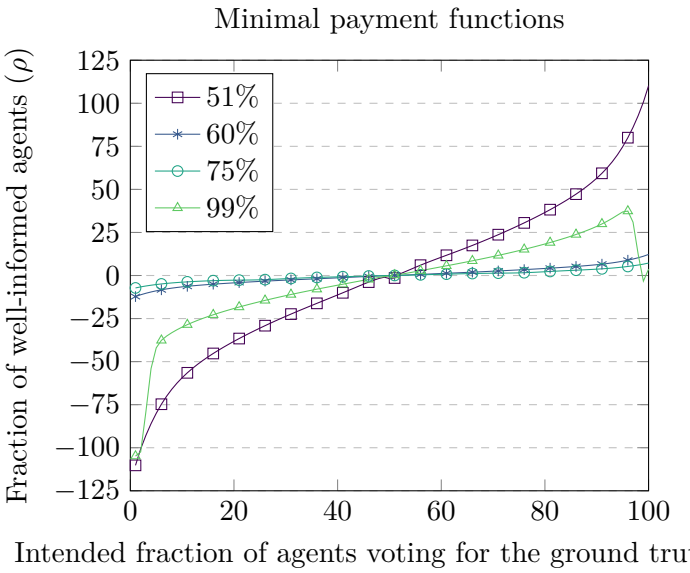

Figure 4.1: Minimal payment functions that ensure the existence of a simple equilibrium with an x-fraction of the agents casting a T-vote on average, so that all agents have non-negative expected utility. The scenario uses $n = 100$ and the effort function $f(x) = 1 - \frac{e^{-x}}{2}$. The payment functions obtained by solving the linear program from Theorem 4.7 for $x \in \{0.51, 0.6, 0.75, 0.99\}$ are shown. Each point (x, y) on a curve means that an agent will receive a payment of y if an x-fraction of the agents voted in the same way as they did. There is a marker for every fifth data point.

Theorem 4.7. *Consider the scenario with n agents, among which the well-informed ones have the same effort function f and the misinformed ones have the same effort function $1 - f$. Given $x \in (1/2, 1)$, selecting the payment function that satisfies the conditions of Lemma 4.4, induces an equilibrium in which all agents have non-negative expected utility and an expected x-fraction of agents casts a T-vote so that the expected amount given to the agents as payment is minimized, can be done in time polynomial in n using linear programming.*

Our approach can be extended to include additional constraints (e.g., non-negativity or monotonicity of payments), provided they can be expressed as linear constraints of the payment parameters. Fig. 4.1 depicts four payment solutions obtained by solving the above LP for $n = 100$ and the effort function $f(x) = 1 - \frac{e^{-x}}{2}$, and values of x ranging from 51% to 99%.

4.4 Computational Experiments

Our goal in this section is to justify that appropriate selection of the payment parameters can lead to correct adjudication in practice, even though Lemma 4.3

shows the co-existence of both good and bad equilibria. The key property that favors good equilibria more often is that, in practice, jurors are on average closer to being well-informed than misinformed. Formally, this means that $\frac{1}{n} \cdot \sum_{i \in [n]} f_i(x) > 1/2$ for every $x > 0$.

Due to the lack of initial feedback, it is natural to assume that agents start their interaction by putting in some small effort and convert their signal to a vote. We claim that this, together with their tendency to be well-informed, is enough to lead to probably correct adjudication despite strategic behavior. We provide evidence for this claim through the following experiment implementing the scenario we considered in Section 4.3.

We have n agents, a ρ-fraction of whom are well-informed and the rest are misinformed. Agent i's effort function is $f_i(x) = 1 - \frac{e^{-x}}{2}$ if they are well-informed and $f_i(x) = \frac{e^{-x}}{2}$ if they are misinformed. We consider the minimal payment functions, defined as the solution of the linear program detailed in the last section, parameterized by the fraction x of agents intended to vote for the ground truth. A small subset of these payment functions can be seen in Fig. 4.1. In addition, we consider two different payment functions, both defined using a parameter $\omega > 0$:

- $p(x) = \omega$ if $x \geq 1/2$ and $p(x) = 0$, otherwise.
- $p(x) = \frac{\omega}{xn}$ if $x \geq 1/2$ and $p(x) = -\frac{\omega}{xn}$, otherwise.

With the first payment function, each agent gets a payment of ω if their vote is in the majority, while they get no payment otherwise. With the second payment, the agents in the majority share an award of ω, while the agents in the minority share a loss of ω. Notice that both payment functions satisfy the conditions of Lemma 4.4. We will refer to them as *threshold* and *award/loss sharing* payment functions, respectively.

In our experiments, we simulate the following dynamics of strategic play. Initially, all agents put an effort of $\epsilon > 0$ and cast the signal they receive as their vote. In subsequent rounds, each agent best-responds. In particular, the structure of the dynamics is as follows:

Round 0**:** Agent i puts an effort of ϵ and casts their signal as their vote.

Round j**, for** $j = 1, 2, ..., R$**:** Agent i gets m_i as feedback. They decide their strategy $\beta_i \in \{0, 1\}$ and effort level $\lambda_i \geq 0$. They draw their signal, which is alternative T with probability $f_i(\lambda_i)$ and alternative F with probability $1 - f_i(\lambda_i)$. If $\beta_i = 1$, they casts their signal as their vote; otherwise, they casts the opposite of their signal as their vote.

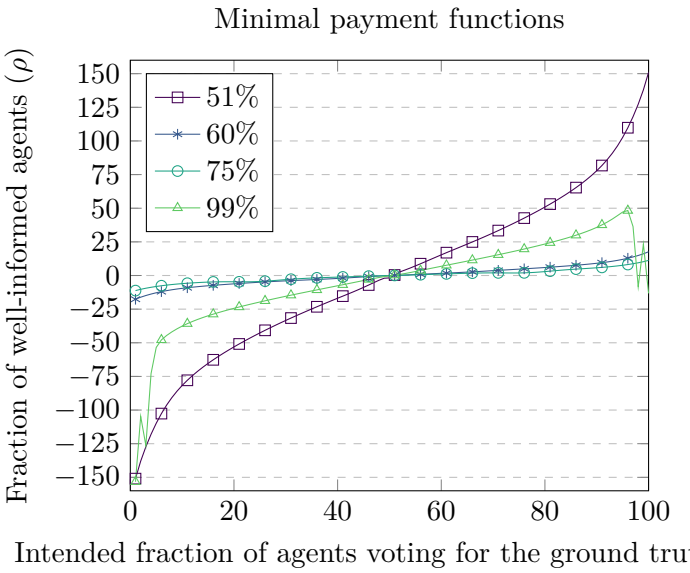

Figure 4.2: Minimal payment functions computed using the approach of Section 4.3, after relaxing Equation (4.13) to a lower bound inequality. The scenario uses $n = 100$ and the effort function $f(x) = 1 - \frac{e^{-x}}{2}$. The payment functions obtained by solving the linear program from Theorem 4.7 for $x \in \{0.51, 0.6, 0.75, 0.99\}$ are shown. Each point (x, y) on a curve means that an agent will receive a payment of y if an x-fraction of the agents voted in the same way as they did. There is a marker for every fifth data point.

In each round after round 0, agents get the exact value of m_i as feedback (as opposed to its distribution)[1] but maximize their expected utility with respect to the components λ_i and β_i of their strategy. Hence, the only difference with what we have seen in earlier sections is that the calculation of expected utility considers the actual value of payments and not their expectation, i.e.,

$$\mathbb{E}[u_i(\lambda_i, \beta_i, m_i)] = -\lambda_i + p\left(\frac{1+m_i}{n}\right) + (\beta_i(2f_i(\lambda_i) - 1) - f_i(\lambda_i)) \cdot Q(m_i),$$

where

$$Q(m_i) = p\left(\frac{1+m_i}{n}\right) - p\left(\frac{n-m_i}{n}\right).$$

By applying Lemma 4.1, we get the following characterization of the best-response of agent i in round $j > 0$.

Corollary 4.8. *The best response of agent i receiving feedback m_i is as follows:*

[1] An alternative implementation would assume that m_i takes the number of T-votes in a randomly chosen previous round. The results obtained in this way are qualitatively similar to those we present here.

(a) If $|Q(m_i)| \leq 2$*, then* $\lambda_i = 0$ *and* β_i *can take any value in* $[0, 1]$*.*

(b) Otherwise, $\lambda_i = \ln \frac{|Q(m_i)|}{2}$*.*

(b.1) If agent i *is well-informed and* $Q(m_i) > 2$ *or agent* i *is misinformed and* $Q(m_i) < -2$*, then* $\beta_i = 1$*.*

(b.2) If agent i *is misinformed and* $Q(m_i) > 2$ *or agent* i *is well-informed and* $Q(m_i) < -2$*, then* $\beta_i = 0$*.*

In our experiments, we consider an agent population of fixed size $n = 100$, with the fraction of well-informed agents ranging from 0 to 1. We simulate the dynamics described above for $R = 50$ rounds and repeat each simulation 20 times. For each experiment, we measure the frequency with which the majority of votes after the R-th round is for the ground truth alternative T. We do so for both the threshold and award/loss sharing payment functions, with parameter ω taking values ranging between 0 and 5 for the threshold payment functions and between 0 and 100 for the award/loss sharing one. We also consider the payment functions that arise as solutions to the linear programs considered in the previous section. In each experiment, we play with the values of two parameters simultaneously. We consider 100 values on each axis and plot the resulting data using a heatmap, with each data point corresponding to the average correctness observed during the experiment. We represent the correctness using the viridis color scale, with yellow points corresponding to a good recovery of the ground truth, and dark blue points corresponding to poor recovery. Random values are represented by turquoise points.

In the first experiment (Fig. 4.3.a), we consider the threshold payment function and vary the size of the reward ω and the fraction ρ of well-informed agents. We consider a reasonably high starting effort of $\epsilon = 1$, corresponding to a probability of 0.816 of receiving the ground truth as signal. We observe two distinct regions as we vary the size of the payment. Initially, when the payment is too small (i.e. $\omega \leq 2.5$), the outcome of the adjudication is mostly random. When the payment increases above the threshold, we observe a *sharp phase transition* independent of ρ, where the correctness is extremified by the payment in the following sense: when ρ is sufficiently large (respectively, small), the mechanism recovers the ground truth with high (respectively, low) probability. When $\rho \approx 0.5$, we see that the outcome of the adjudication is mostly random.

In the second experiment (Fig. 4.3.b), we consider the award/loss sharing payment function. The range of ω is changed from $[0, 5]$ to $[0, 100]$, as the latter constitutes the total award, while the former is the award per agent. All other parameters are kept the same. We obtain similar results as for the threshold payment function, i.e. the outcome is mostly random below a threshold above which we observe a sharp phase transition where the outcome of the mechanism is extremified. Here, the phase transitions happens when the total award is $\omega \approx 60$.

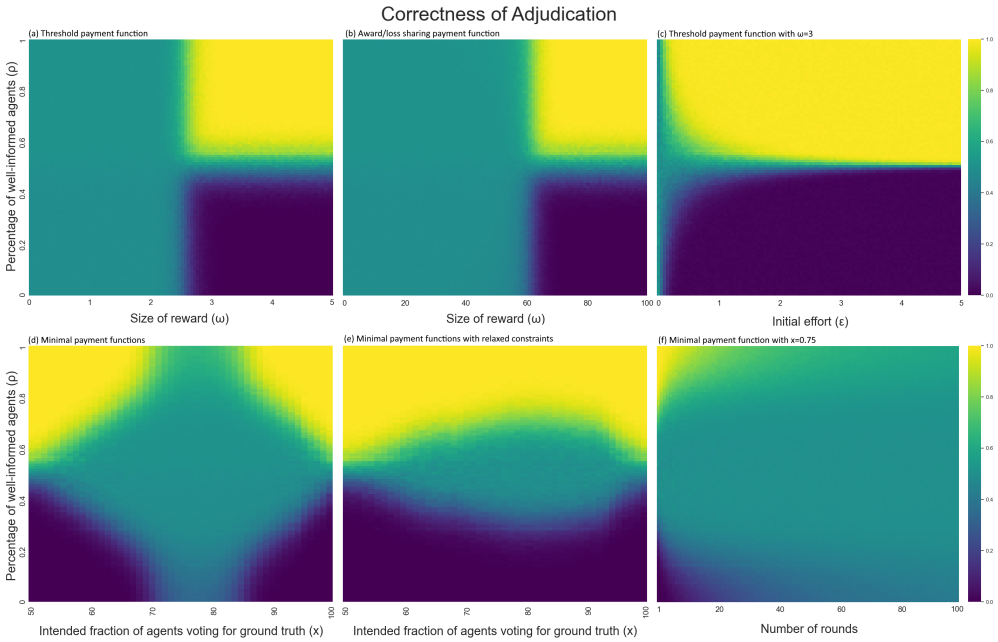

Figure 4.3: Heatmap of the correctness of the adjudication, plotted with the fraction of well-informed agents on the y-axis, with six varying x-axes. In each plot, we run $R = 50$ rounds with a jury of size $n = 100$, using 1000 samples for each data point. The color of a data point indicates the average measured correctness with the given parameters, using the viridis color scale displayed in the legend on the right. Yellow corresponds to good recovery, while dark blue corresponds to poor recovery of the ground truth, while random outcomes are represented by turquoise. The six x-axes are as follows: (a) Size of the reward for the threshold payment function, ranging from $\omega = 0$ to $\omega = 5$, with $\epsilon = 1$. (b) Size of the reward for the award/loss sharing payment function, ranging from $\omega = 0$ to $\omega = 100$, with $\epsilon = 1$. (c) The initial effort ϵ, ranging from $\epsilon = 0$ to $\epsilon = 5$, with the payment function being the threshold payment function with $\omega = 3$. (d) The intended fraction x of agents voting for the ground truth, ranging from $x = 0.51$ to $x = 1$, with the payment function defined by Theorem 4.7. (e) The intended fraction x of agents voting for the ground truth, ranging from $x = 0.51$ to $x = 1$, with the payment functions obtained from Theorem 4.7 by relaxing Eq. (4.13) to an inequality. (f) The number of rounds, ranging from $R = 1$ to $R = 100$, with the payment function being the minimal payment function with $x = 0.75$ from Fig. 4.1.

In the third experiment (Fig. 4.3.c), we observe the effect on the correctness by the initial effort. We fix the threshold payment function with $\omega = 3$ such that the mechanism has a chance to recover the ground truth, and let ϵ range from 0 to 5. We observe that, when ϵ is small, the outcome of the mechanism is mostly random, while the outcome quickly extremifies as ϵ increases. This means the mechanism only works if the agents initially put in sufficient effort. The results are similar for both the award/loss sharing payment function and the minimal payment functions.

In the fourth experiment (Fig. 4.3.d), we consider the payment functions obtained from Theorem 4.7. A subset of the payment functions we use are depicted in Fig. 4.1. Here, instead of varying the size of the reward, we vary the parameter x used as input to the linear program. This parameter represents the intended fraction of agents voting for the ground truth at equilibrium. We let x range from 0.51 to 1 in increments of 0.01. Here, we observe that for x close to 0.5 and for x close to 1, the mechanism is extremified, while for x close to 0.75 and ρ close to 0.5 the outcome of the mechanism is mostly random. This is rather unexpected since if a 0.75-fraction of the agents vote for the ground truth, the majority vote will be for the ground truth almost certainly. Indeed, we observe that in these games when $\rho \approx 0.5$, the agents exert effort close to zero, hence producing the random outcome. We claim that despite this behavior, the ground truth is still an equilibrium, it is just not a stable equilibrium and the parties converge to the trivial equilibrium.

In a fifth experiment (Fig. 4.3.e), we consider a different set of minimal payment functions, obtained by relaxing the equality constraint Eq. (4.13) to a lower bound inequality. This has the effect of no longer requiring an exact x-fraction of the agents vote for the ground truth, but instead gives a lower bound on their number. This slightly changes the payment functions which can be seen in Fig. 4.2, though they are qualitatively similar to those shown in Fig. 4.1. Here, we again vary the fraction ρ of well-informed agents on the y-axis, and the intended fraction x of agents voting for the ground truth, ranging from $x = 0.51$ to $x = 1$ in increments of 0.01. However, we obtain different and considerably better results than those in Fig. 4.3.d. In particular, we obtain a good adjudication outcome for any x when $\rho > 0.75$.

In our sixth and final experiment (Fig. 4.3.f), we aim to explain the enigmatic behavior of the LP-computed payments for $x \approx 0.75$. We fix the payment function to be the minimal payment function with $x = 0.75$ and vary the number of rounds from 1 round to 100 rounds. We do not take into account round 0 where all parties exert $\epsilon > 0$ effort in the estimation of the correctness of the outcome. We observe that the outcome is extremified when the number of rounds is small and decays as we increase the number of round. We can explain this result by considering the payment function for $x = 0.75$ in Fig. 4.1 whose distribution is mostly flat when the outcome is close to being a tie. Here, the value of $Q(m_i)$ is small so the agent will lower the effort they exert, making it more likely that the outcome will be disputed. This creates a pull

towards the trivial equilibrium. By contrast, the curves for $x \in \{0.51, 0.99\}$ have a higher slope close to 0.5, which makes this effect less pronounced. This explains why the adjudication outcome is mostly random for $x \approx 0.75$. By design, the linear program finds minimal payments that ensure there is an equilibrium where an x-fraction of the agents vote in favor of the ground truth. However, it does not constrain the solution to have the property that the good equilibrium is *stable*. In some sense, the fact that the non-trivial equilbrium is stable when x is far from 0.5 is happenstance and begs the deeper question why the solutions to the linear program are of the form we observe. Intuitively, it makes sense that attaining a high accuracy requires large payments. A similar phenomenon seemingly holds for accuracies close to 0.51 which can be explained informally as follows. Combinatorially, there are only a few ways to attain an accuracy of 0.51 which necessitates the use of large punishment and rewards when the vote is close to being a tie. By contrast, for larger ρ, there are more ways to attain an accuracy of 0.75 in the majority, hence loosening the requirements on the payments. This suggests that the case $x = 0.75$ does not provide positive results in practice because of *instability of equilibria*. It would be interesting to explore whether it is possible to extend our approach with additional natural constraints that ensure the non-trivial equilibrium is also stable.

Our experiments suggest that several classes of payment functions can be used to recover the ground truth with high probability, provided the agents are well-informed on average. Clearly, there is much work yet to be done in designing payment functions with desirable properties: while the threshold function and the award/loss sharing function seem to recover the ground truth reliably, it might be difficult in practice to pinpoint the location of the phase transition, as this requires estimating the effort functions used by actual jurors. The same holds for the minimal payment functions.

Chapter 5

Payments

> "*Never underestimate the effectiveness of a straight cash bribe.*"
>
> Claud Cockburn

It is well-known that the equilibrium often does not ensure the best outcome for the agents involved. The most famous example is the prisoner's dilemma where two criminals are arrested and interrogated by police in separate rooms: each criminal can either cooperate with their accomplice, or defect and give them up to the police, resulting in a reduced sentence. It is well-known that cooperation is not an equilibrium, as neither criminal can trust the other not to defect, although it would be in their common interest to do so. In game theory, this inefficiency can be measured using the *price of anarchy* (PoA), defined as the ratio of the social optimum and the worst possible equilibrium. In seminal work [157], Koutsoupias and Paradimitriou consider a simple model of network routing where the PoA is shown to be ≥ 1.5. This means the lack of coordination between the agents leads to a 33% loss of performance compared to the optimal setting in which the agents coordinate. While this may be problematic on its own, consequences may be more severe if the interaction we are trying to model is related to security. Here, a lack of coordination may lead to irreparable damage (such as the leak of private information) if some agents deviate from the intended strategy. Indeed, in cryptography, such complications are the cause of a seemingly irreconcilable gap between the worlds of rational cryptography and the classic cryptographic model: in [130], Halpern and Teague famously show there is no deterministic bounded-time interactive protocol for secure function evaluation on private inputs involving rational agents with a certain class of utility functions, namely agents who prefer to learn the output of the function, but prefer as few other agents as possible learn the output. By contrast, there are simple and efficient *actively secure* protocols when a sufficient subset of the agents are guaranteed to be honest, even when the remaining agents are

allowed to deviate arbitrarily [74]. A weaker notion of security called 'covert security' was proposed by Aumann and Lindell in [16]. Here, agents are allowed to deviate but are caught with some constant non-zero probability. This was extended by Asharov and Orlandi in [12] to publicly verifiable covert (PVC) security where a certificate is output that can be verified by a third agent to determine if cheating has occurred. The underlying assumption of these protocols is that the cost associated with the risk of being caught outweighs the benefit of deviating. Indeed, the problem of misaligned equilibria is usually mitigated in practice by ensuring appropriate punishment for misbehaving, such as fining deviants, banning them from participating again, or subjecting them to other legal repercussions, effectively changing the utilities of the game to ensure being honest is, in fact, an equilibrium. In our example with the prisoner's dilemma, a criminal who defects might face consequences after the other criminal is released from prison, as the adage goes: "snitches get stitches". Sometimes, it is less clear how to punish agents, as when the games are models of interaction on the internet where agents can be anonymous.

In this chapter, we propose a generic mechanism to incentivize behavior in an arbitrary finite game with the use of payments. The mechanism can be implemented using deposits to a smart contract deployed on a blockchain and can be considered a generalization of the smart contract for decentralized commerce that we presented in Chapter 3. Note that our task is trivial if we allow the mechanism to observe all actions taken in the game, as this allows the mechanism to simply punish those agents that deviate from the intended behavior. Hence, our main contribution is proposing a framework that models a mechanism that only probabilistically observes actions taken by the agents and to determine appropriate payments. Indeed, the problem of misaligned equilibria is usually mitigated in practice by ensuring appropriate punishment for misbehaving, such as fining deviants, banning them from participating again, or subjecting them to other legal repercussions, effectively changing the utilities of the game to ensure being honest is, in fact, an equilibrium. In our example with the prisoner's dilemma, a criminal who defects might face consequences after the other criminal is released from prison, as the adage goes: "snitches get stitches". Sometimes, it is less clear how to punish agents, as when the games are models of interaction on the internet where agents can be anonymous.

Attribution. This chapter (including the above introduction) is taken verbatim from [223], with only minor modifications to the formatting and the prose.

Our Contributions

We propose a mechanism for incentivizing intended behavior in an arbitrary finite game with the use of payments. We show that payments can be used to implement any set of utilities if and only if the mechanism can essentially

infer the entire execution of the game (Lemma 5.6). We show that our model generalizes similar models in the literature, such as 'adversarial level agreements' by George and Kamara [109] retained as a special case. We sketch how to implement the payments in a distributed setting by letting agents deploy a smart contract on a blockchain. We demonstrate how to use the framework by applying it to the special case of decentralized commerce, resulting in a smart contract that is qualitatively similar to the one proposed in Chapter 3.

We investigate the computational complexity of computing an optimal payment scheme, in the sense that the payments are minimized. For games of perfect information, we observe that the problem is equivalent to linear programming under logspace reductions, thus showing the following.

Theorem 5.1 (Informal)**.** *Finding an optimal payment scheme for a finite game of perfect information, or showing no suitable payment scheme exists, is* P*-complete.*

For games of imperfect information, it is well-known that even computing an equilibrium is PPAD-complete [81], so it is unlikely there is an efficient algorithm for finding an optimal payment scheme in these cases. As a consequence, we conjecture that finding an optimal payment scheme for finite games of imperfect information is PPAD-hard.

To showcase the applicability of our model, we apply it to the problem of secure multiagent computation. We show that payments can be used, together with what is known as an 'ε-deterrent publicly verifiable covert (PVC) secure protocol' [12, 16], to yield a secure protocol for secure function evaluation involving rational agents, where ε is the probability of an agent getting caught cheating. We stress that this does not violate the impossibility result of Halpern and Teague for the simple reason that they explicitly assume the utilities are not quasi-linear, hence not allowing payments. By contrast, we allow payments to alter an agent's utility function, in such a way that the conditions required to establish the impossibility result no longer hold.

Theorem 5.2 (Informal)**.** *Any function f can be computed with δ-strong game-theoretic security with rational agents with black-box access to any ε-deterrent PVC protocol, by using a payment scheme where each agent pays $O(1 + \delta/\varepsilon)$.*

Finally, we prove a lower bound on the size of the largest punishment (equivalently, *deposit*) for all games that are budget balanced. We show the punishments must be linear in the size of the desired level of security. Note that this matches asymptotically the bound of Theorem 5.2 since n, s, and ε are constant for any fixed PVC protocol.

Theorem 5.3 (Informal)**.** *Any budget balanced payment scheme that achieves δ-strong game-theoretic security in a game of n agents must have a largest punishment that is no smaller than $\Omega(1 + \delta\sqrt{n}/s)$, where s is the number of observable outcomes.*

This chapter paper is organized as follows. We start in Section 5.1 by defining our model of payment schemes. We show how to implement a payment scheme using a smart contract, and prove that payments can be used to implement any set of utilities if and only if the mechanism can essentially infer all information about what happened. In Section 5.2, we consider the computational complexity of finding payment schemes and prove Theorem 5.1. Next, in Section 5.3, we apply the framework to secure MPC and prove Theorem 5.2. Finally, in Section 5.4, we show a lower bound on the size of the maximum deposits and prove Theorem 5.3.

Related Work

Mechanism Design.

The use of payments to incentivize behavior is a well-studied problem in mechanism design, where the payments are often called 'scoring rules'. Such a rule assigns a score (payment) to each outcome of an interaction, and can e.g. be used to elicit truthful responses. In this case, we say the scoring rule is *proper*, examples of which include the quadratic scoring rule and the logarithmic scoring rule [116, 224]. A mechanism for which an agent maximizes their utility by reporting their beliefs truthfully is said to be *truthful*. The logarithmic scoring rule is used by Prelec to implement a truthful mechanism for voting in the Bayesian truth serum (BTS) model [206]. This was extended to 'robust BTS' by Witkowski and Parkes [254] that instead uses the quadratic scoring rule. Scoring rules are also used in peer prediction methods [178], Bayesian markets [20] and choice matchings [75]. Payments are also used more generally in the generic Vickrey-Clarke-Groves (VCG) mechanism for obtaining a socially optimal outcome [67, 127, 248]. However, the VCG mechanism is fundamentally limited to games that involve distributing a set of 'items' among a set of agents. It is not obvious how this would apply to an arbitrary extensive-form game. Another relevant line of work is that of environment design [265] by Zhang, Chen, and Parkes where a designer wants to influence an agent's decisions (arbitrarily) by making changes to their environment. They consider a single agent in a dynamic setting and give an *elicitation algorithm* (see also work by Zhang and Parkes [264]) that maximizes the goal value for the agent. Although expressive, it is not obvious how to apply the framework to arbitrary extensive-form games with any number of agents.

Distributed Computing.

Numerous works in the literature take advantage of payment schemes to incentivize the participants to behave honestly. Such payment schemes are usually implemented by deploying a smart contract on a blockchain, such as the contracts [11, 222] presented in Chapter 3; the protocol starts with each agent submitting a 'deposit' that is repaid only if they are found to act as intended.

The work most related to ours is by George and Kamara [109] who propose a framework for incentivizing honesty using 'adversarial level agreements' that specify damages agents must pay if found to act adversarially. We will show later that their model can be recovered as a special case of our model. Faust, Hazay, Kretzler, and Schlosser propose 'financially backed covert security' [102] to punish agents who are caught deviating in a PVC protocol. Their work is focused on the cryptographic implementation, and as a result, they do not formally analyze the equilibria induced by their mechanism. In [268], Zhu, Ding, and Huang propose a protocol for two-agent computation that incentivizes honesty using a publicly verifiably covert secure protocol augmented with a payment scheme. In [91], Dong *et al.* propose a protocol that uses payment schemes to incentivize honesty in outsourcing cloud computations. BitHalo [269] implements an escrow using deposits and multi sigs that was analyzed in [31] by Bigi *et al.* Deposits have also been used for 'truth-telling mechanisms': in [3], Adler *et al.* propose a system, Astraea, that uses deposits and rewards to incentivize a group of voters to decide the validity of a proposition. Kleros [170] uses a similar mechanism to implement a decentralized court system.

Economics.

In the economics literature, the payment schemes that we study are known as 'deposit-refund systems' [106]. They are often studied in the context of environmental issues for incentivizing compliance with laws and regulations. In [122], Grimes-Casey *et al.* propose a game-theoretic model using such deposit-refund systems to analyze consumer behavior with refillable plastic bottles. Indeed, deposit-refund systems are currently used in many countries for closing the gap between the marginal private cost and the marginal external cost of disposing of e.g. bottles, batteries, tires, and consumer electronics, see e.g. [252] for an overview. Such systems can also be used at a higher level of governance: in [180], McEvoy studies deposit-refund systems as a means of enforcing nations to comply with international environmental agreements.

5.1 Payment Schemes

In this section, we present our model of games with payment schemes and show when they can be used to ensure it is rational to play an intended strategy. We consider only finite games of perfect information as it is unlikely there is an efficient procedure in general (as was argued in Chapter 2). We consider a set of n agents $P_1, P_2, \ldots, P_n$ playing a fixed finite extensive-form game G of perfect information. The agents are assumed to be risk-neutral such that they aim to maximize their expected utility. We assume the agents have quasi-linear utilities such that we can use payments to change their incentives. We take as input a unique pure strategy profile s^* that we want the agents to play that we call the *honest strategy profile*. We denote by $\mathbf{u}^* = u(s^*) \in \mathbb{R}^n$ the utility

vector for the honest strategy profile. Note that s^* is required to be pure since it is impossible to determine (without multiple samples) if an agent plays a mixed strategy. This has the effect that s^* defines, at each branch in the game, a unique 'honest move' that the corresponding agent must play. Our goal is to construct a procedure Γ that takes as input a game G in a black-box way and produces an 'equivalent' game $\Gamma(G)$ that implements a different utility matrix $\mathbf{E}$ such that s^* is an equilibrium.

Information Structures.

To construct the procedure Γ, we need to be able to infer *something* about what happened during the execution of the game, as otherwise we are simply 'shifting' the utilities of the game, not changing the structure of its equilibria. We call a mechanism that enables inferring information from a game an *information structure*. We assume playing the game emits a symbol from a fixed finite alphabet Σ of possible outcomes that can be observed. The alphabet serves as a proxy for how the agents acted in the execution of the game. We associate with each leaf of the game a distribution on Σ. When the game terminates, a symbol is sampled according to the distribution and observed in a one-shot manner (the observation should not be considered a Bayesian update).

Definition 5.4 (Information Structure)**.** *An* information structure *for G is a pair $\langle \Sigma, \mathbf{\Phi} \rangle$ where Σ is a finite alphabet of symbols with some arbitrary but fixed order on its symbols, $\sigma_1, \sigma_2, \ldots \sigma_s$, and where $\mathbf{\Phi} = (\phi_{kj}) \in \mathbb{R}^{s \times m}$ is a matrix of emissions probabilities such that every column of $\mathbf{\Phi}$ is a pdf on the symbols of Σ.*

Given a finite game with an information structure, a *payment scheme* Γ is a mechanism that changes the utilities of the game. At the end of the game, the payment scheme rewards or punishes the agents based on what was emitted by the information structure. We assume *quasi-linear* utility functions, i.e. if for an agent the game ends in an outcome with utility u and we give them x money, their utility is $u + x$. We may also say the utilities of the game are given in arbitrarily divisible currency which the payment scheme can process. An agent P_i is indifferent to obtaining an outcome that gives them u_{ij} utility and receiving u_{ij} money. In other words, we make the implicit assumption that 'everything has a price' and intentionally exclude games that model interactions with events that are not interchangeable with money. This circumvents the impossibility result of Halpern and Teague who implicitly assume a fixed total order on the set of possible outcomes. By contrast, quasi-linearity allows the payment schemes to alter the order by punishing or rewarding agents with money.

Definition 5.5 (Payment Scheme)**.** *A* payment scheme *for* $\langle G, \mathcal{I} \rangle$ *is a matrix* $\mathbf{\Lambda} = \{\lambda_{ik}\} \in \mathbb{R}^{n \times s}$*, where* $\mathcal{I} = \langle \Sigma, \mathbf{\Phi} \rangle$ *is an information structure for* G*, and* λ_{ik} *is the utility lost by* P_i *when observing the symbol* $\sigma_k \in \Sigma$*.*

$\mathbf{\Lambda}$ is a matrix that explicitly defines how much utility λ_{ik} agent P_i loses when the payment scheme observes the symbol $\sigma_k \in \Sigma$. Note that $\mathbf{\Lambda}$ is allowed to contain negative entries which means agents receive back more funds from the payment scheme than they initially deposited. When the game is played reaching the leaf ℓ_j, by quasi-linearity the expected utility of agent P_i is the utility they would have received in a normal execution, minus their expected loss from engaging with the payment scheme:

$$\mathbb{E}[P_i \text{ utility in leaf } \ell_j] = u_{ij} - \sum_{k=1}^{s} \lambda_{ik}\, \phi_{kj} = [\mathbf{U} - \mathbf{\Lambda\Phi}]_{ij} \tag{5.1}$$

Correspondingly, we say $\mathbf{\Lambda}$ *implements the utility matrix* $\mathbf{E}$ if $\mathbf{E} = \mathbf{U} - \mathbf{\Lambda\Phi}$. For a fixed $\mathbf{E}$, we denote by $\mathcal{S}(\mathbf{E}) \subseteq \mathbb{R}^{n \times s}$, the set of all payment schemes that implement $\mathbf{E}$ which constitutes an affine subspace of $\mathbb{R}^{n \times s}$. We now consider the search problem, given $\mathbf{\Phi}$ and $\mathbf{U}$, find an $\mathbf{\Lambda} \in \mathcal{S}(\mathbf{E})$ with minimal payments. We will show this problem is equivalent to linear programming, and thus P-complete.

Lemma 5.6. $\mathcal{S}(\mathbf{E})$ *is nonempty for every* $\mathbf{E}$ *if and only if* $\mathbf{\Phi}$ *is left-invertible.*

Proof. If $\mathbf{\Phi}$ is left-invertible, then for any $\mathbf{E}$ we can let $\mathbf{\Lambda_E} := (\mathbf{U} - \mathbf{E})\, \mathbf{\Phi}^{-1}$ where $\mathbf{\Phi}^{-1}$ is a left-inverse of $\mathbf{\Phi}$. It follows that $\mathbf{U} - \mathbf{\Lambda\Phi} = \mathbf{U} - (\mathbf{U} - \mathbf{E})\mathbf{\Phi}^{-1}\mathbf{\Phi} = \mathbf{U} - \mathbf{U} + \mathbf{E} = \mathbf{E}$.

Suppose instead $\mathcal{S}(\mathbf{E})$ is nonempty for each $\mathbf{E}$. This means that we can always find $\mathbf{\Lambda_E}$ that solves $\mathbf{U} - \mathbf{E} = \mathbf{\Lambda_E\Phi}$. Assume for the sake of contradiction that there are fewer symbols than leaves. Thus, there is a leaf for which the payment vector is a fixed linear combination of the payments of the other leaves; there must be an $\mathbf{E}$ that we cannot implement. But this is a contradiction so we assume there are at least as many symbols as leaves. This means we can choose $\mathbf{E}$ such that $\mathbf{U} - \mathbf{E}$ is left-invertible with left-inverse $\mathbf{F} \in \mathbb{R}^{m \times n}$, which means that $\mathbf{F\Lambda_E\Phi} = \mathbf{I}_m$, a contradiction. □

In particular, we can only implement any $\mathbf{E}$ we want if there are at least as many symbols as leaves in the game tree, and that these symbols are not duplicates, in the sense that the distributions of symbols across the leaves are distinct.

Properties of Payment Schemes.

In general, we may want to pick an $\mathbf{\Lambda} \in \mathcal{S}(\mathbf{E})$ with some desirable properties. If a property is *linear*, it will intersect $\mathcal{S}(\mathbf{E})$ in a (possibly empty) affine subspace of $\mathcal{S}(\mathbf{E})$ and as such does not change the complexity of the search problem. We

shall only consider payment schemes that are budget balanced, in the sense that the total payments are non-negative. Other examples of linear properties include (ex ante/ex post) individual rationality, honest invariance (utility for s^* is unchanged by payments), or strong budget balance (payments are exactly zero). In general, such properties are hard to characterize, as they are not necessarily invariant to scaling/perturbing the utilities (unlike an equilibrium), and as such are sensitive to the precise modeling of utilities used for a given application.

Proposition 5.7. *To implement* $\mathbf{E}$ *with strong budget balance, each column of* $\mathbf{U} - \mathbf{E}$ *must sum to 0.*

Proof. To implement $\mathbf{E}$, we must have $\mathbf{\Lambda\Phi} = \mathbf{U} - \mathbf{E}$. From strong budget balance we have that $\mathbf{1}^\top \mathbf{\Lambda} = \mathbf{0}^\top$, and hence $\mathbf{1}^\top(\mathbf{U} - \mathbf{E}) = \mathbf{1}^\top \mathbf{\Lambda\Phi} = (\mathbf{1}^\top \mathbf{\Lambda})\mathbf{\Phi} = \mathbf{0}^\top \mathbf{\Phi} = \mathbf{0}$. □

A Special Case: Adversarial Level Agreements

We now show how the model of 'adversarial level agreements' (ALAs) by George and Kamara [109] can be recovered as a special case of our model. An ALA for a game with n agents consists of 1) a description of the intended strategy for each agent, and 2) a vector of damages $\mathbf{d} \in \mathbb{R}^n$ that specifies how much utility $\mathbf{d}_i$ agent P_i should lose when found to deviate from the intended strategy. Their model does not explicitly consider deviations by more than a single agent, so we can state this as an information structure with the alphabet $\Sigma = \{\top, \bot_1, \bot_2, \ldots, \bot_n\}$. Here, $\top$ means all agents were honest, and $\bot_i$ means P_i deviated. The emission matrix $\mathbf{\Phi}$ depends on the specific application. An ALA then corresponds to a payment scheme of the following form.

$$\mathbf{\Lambda} = \begin{pmatrix} 0 & \mathbf{d}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{d}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{d}_n \end{pmatrix}$$

Note that we can easily generalize this to deviations by any $t \leq n$ agents by including more symbols, e.g. $\bot_{12}$ or $\bot_{456}$.

Payment Schemes as Smart Contracts.

We now explain informally how a payment scheme can be deployed in practice as a smart contract running on a blockchain. We present a simplified model as there are many subtleties in getting such contracts rigorously secure, see e.g. [102, 152] for more formal cryptographic modeling. At a high level, we want to ensure agent P_i loses λ_{ik} utility when the symbol σ_k is observed. We can implement this by defining $\lambda_i^* := \max_{k \in \{1,2,\ldots s\}} \lambda_{ik}$ and letting each agent

P_i make a deposit of size λ_i^* to a smart contract before playing the game. Afterwards, the agents are repaid appropriately by the payment scheme to ensure their utility is as dictated by $\mathbf{\Lambda}$. Suppose we fix some payment scheme Γ, then the game $\Gamma(G)$ is played as follows:

1. Each P_i makes a deposit of λ_i^* to the payment scheme.
2. The game G is played, and a symbol σ_k is observed.
3. Each agent P_i is repaid $\lambda_i^* - \lambda_{ik}$.

This can be implemented in a fairly straightforward manner using a scripting language and deployed as a smart contract running on a blockchain, assuming access to some information structure with known bounds on the emission probabilities. We stress that this is only one possible implementation of a payment scheme suitable in any scenario, even over the internet when agents are anonymous. The important thing is that agent P_i loses λ_{ik} utility when symbol σ_k is observed. When agents are not anonymous and can be held accountable, the payment scheme can be used in an optimistic manner as was argued by George and Kamara.

Example: Decentralized Commerce

In this section, we demonstrate the applicability of our model by applying it to the problem of decentralized commerce that was introduced in Chapter 3. Consider a seller S who wants to sell an item *it* over the internet to a buyer B for x money. To make the problem non-trivial, we assume *it* is physical such that the protocol cannot be entirely implemented using cryptography (see e.g. [11, 13, 96, 161, 162] for solutions that work with digital goods under computational assumptions). We assume *it* has a value of y to the seller, and a value of x' to the buyer. To make the problem feasible, we assume that $y > x > x' > 0$. We consider a simple game where S first decides whether to send *it* to B, after which B decides whether or not to pay S. The resulting extensive-form game is depicted in Fig. 5.1. In this simple game, the trade will never be completed, as it is evidently rational for the buyer to always reject delivery of the item; consequently, it is rational for the seller not to send the item. This seems to contradict empirical data, as variants of this game are played successfully all the time. The reason for this is that, in practice, buyer and seller are not anonymous and can be held accountable for fraud, and potentially subject to legal repercussions. Also, such trades are typically processed by a middleman that may offer some insurance for either buyer or seller. However, from a cryptographic point of view, centralized solutions are undesired. There are potential privacy risks with using centralized marketplaces, e.g. middlemen using consumer analytics for targeted advertising. Also, such marketplaces potentially have an incentive to engage in monopolistic activity,

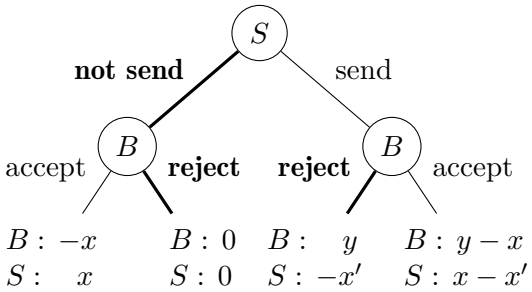

Figure 5.1: Extensive-form representation of the decentralized commerce game. The dominating paths are shown in bold. We observe that the dominating strategy is for the seller not to send the item, and for the buyer to withhold their payment regardless of whether they received the item.

e.g. through removing competitors' products or differential pricing based on consumer demographics. Some of these issues are fixed by a decentralized alternative, for which compliance with laws and regulations can be ensured by instantiating the smart contract on a blockchain with revocable anonymity where users register with an identity provider, and can be deanonymized, e.g. upon request by the court system [76].

We assume the agents use a smart contract to process the trade and have at their disposal an adjudication mechanism with error $\gamma < \frac{1}{2}$, such as the one described in Chapter 4. To proceed, we need to define an information structure on the game. We first define an alphabet of outcomes as $\Sigma = \{\top, \bot_B, \bot_S\}$. Here $\top$ is a symbol emitted if the buyer accepts the trade, and $\bot_B, \bot_S$ are outcomes of the oracle if it is invoked, where $\bot_B$ (resp. $\bot_S$) means 'the buyer (resp. the seller) was dishonest'. We define the emissions matrix of the game as follows.

$$\mathbf{\Phi} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1-\gamma & \gamma & 0 \\ 0 & \gamma & 1-\gamma & 0 \end{pmatrix}$$

Now, let us assume we want to instantiate payments to ensure the game has x-strong game-theoretic security. We proceed using backward induction in Fig. 5.1, letting the corresponding utilities have a difference of $\geq x$. This is e.g. achieved by defining the following 'desired' utility matrix $\mathbf{E}$.

$$\mathbf{E} = \begin{pmatrix} -x & 0 & y-2x & y-x \\ x & -x' & -x' & x-x' \end{pmatrix}$$

Note that despite the payments, the equilibrium is individually rational as $\mathbf{u}^* = (y-x \;\; x-x')^\top$, which is non-negative as $y > x > x'$ by assumption. In

order to implement $\mathbf{E}$, the payment scheme $\mathbf{\Lambda}$ must satisfy Eq. (5.1) as follows.

$$\mathbf{\Lambda\Phi} = \mathbf{U} - \mathbf{E}$$
$$\iff$$
$$\begin{pmatrix} \lambda_{B\top} & (1-\gamma)\lambda_{B\perp_B} + \gamma\lambda_{B\perp_S} & \gamma\lambda_{B\perp_B} + (1-\gamma)\lambda_{B\perp_S} & \lambda_{B\top} \\ \lambda_{S\top} & (1-\gamma)\lambda_{S\perp_B} + \gamma\lambda_{S\perp_S} & \gamma\lambda_{S\perp_B} + (1-\gamma)\lambda_{S\perp_S} & \lambda_{S\top} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2x & 0 \\ 0 & x' & 0 & 0 \end{pmatrix}$$

This immediately gives $\lambda_{B\top} = \lambda_{S\top} = 0$, while the remaining payments are given by four equations with four unknown and can be solved using Gaussian elimination to yield the following payment scheme.

$$\mathbf{\Lambda} = \begin{pmatrix} 0 & -\frac{2\gamma}{1-2\gamma}x & \frac{2(1-\gamma)}{1-2\gamma}x \\ 0 & \frac{1-\gamma}{1-2\gamma}x' & -\frac{\gamma}{1-2\gamma}x' \end{pmatrix}$$

In other words, the buyer must make a deposit of size $\frac{2(1-\gamma)}{1-2\gamma}x$ to the smart contract, while the seller must make a deposit of size $\frac{1-\gamma}{1-2\gamma}x'$. To make this more concrete, suppose we have $x = 100€, x' = 50€$, and $\gamma = 0.1$. Then the buyer must make a deposit of size $\lambda_B^* = 225€$, while the seller must make a deposit of size $\lambda_S^* \approx 57€$. The large difference in the deposits reflects the fact that the protocol is 'biased' in favor of the buyer. In practice, while x is known to the mechanism, x' is usually not. Instead, we can use the optimistic variant of this payment scheme that we proposed in Chapter 3.

5.2 Computational Complexity

In this section, we analyze the computational complexity of finding payment schemes in arbitrary games. For games of perfect information, we observe the problem is equivalent to linear programming (denoted LP) under logspace-reductions, thus showing the problem is complete for P.

More formally, we consider the following optimization problem.

$\mathsf{PaymentScheme}_t^\delta$

- **Instance.** Finite game G with utility matrix $\mathbf{U} \in \mathbb{R}^{n\times m}$ and intended strategy profile s^*; finite alphabet Σ with $s = |\Sigma|$, emission matrix $\mathbf{\Phi} \in \mathbb{R}^{s\times m}$, and cost vector $\mathbf{c} \in (\mathbb{R} \cup \{\infty\})^{ns}$.
- **Output.** Budget balanced payment scheme $\mathbf{\Lambda} \in \mathbb{R}^{n\times s}$ s.t. $\Gamma^{\mathbf{\Lambda}(G)}$ has δ-strong, t-robust game-theoretic security, for which $\mathbf{c}^\top \mathrm{vec}(\mathbf{\Lambda})$ is minimized; or $\perp$ if no such payment scheme exists.

Here, ∞ is a formal symbol in the cost function that 'forces' the corresponding payment to equal zero. It does not contribute to the actual cost function. This can e.g. be used to implement *honest invariance*, to ensure the utility

vector for the intended strategy remains unchanged. We allow this modeling to simplify our reductions, though we can make do without this assumption; we sketch how to do so at the end of the section.

Theorem 5.8. *$\mathsf{PaymentScheme}_t^\delta$ is P-complete for games of perfect information.*

We prove this in the next two subsections, by reducing both to and from LP using logspace-reductions. For games of imperfect information, it is unlikely we can find an optimal payment scheme to change the equilibrium, as even computing the equilibrium for these games is known to be PPAD-complete. As a result, we conjecture the problem to be hard.

Conjecture 5.9. *$\mathsf{PaymentScheme}_t^\delta$ is PPAD-hard for imperfect information games.*

Upper Bound: Reduction to LP

In this section, we show how to reduce $\mathsf{PaymentScheme}_t^\delta$ to LP. Since the feasible region is a convex polyhedron, it is unsurprising that we can use linear programming to decide the minimal size of the deposits necessary to establish security. In particular, we can write the necessary constraints for δ-strong t-robust game-theoretic security as a set of linear constraints. For convenience, we will represent the utility matrix $\mathbf{U}$ as a vector $\mathbf{u} \in \mathbb{R}^{nm}$ in row-major order. We will then collect the set of necessary constraints in a matrix $\mathbf{\Psi}^{(t)} \in \mathbb{R}^{\alpha^{(t)} \times nm}$ where $\alpha^{(t)}$ denotes the number of such constraints. We also let $\boldsymbol{\delta}^{(t)} = [\delta, \delta, \ldots, \delta]^\top \in \mathbb{R}^{\alpha^{(t)}}$ be a vector only containing δ. Note that $\alpha^{(t)}$ is a constant that depends on the structure of the game.

Proposition 5.10. *$\mathsf{PaymentScheme}_t^\delta$ can be reduced to LP in logspace.*

Proof. First note that the set of utility matrices with δ-strong t-robust game-theoretic security can be recovered as the set of solutions to the following equation:

$$\mathbf{\Psi}^{(t)}\mathbf{v} \geq \boldsymbol{\delta}^{(t)} \tag{5.2}$$

The matrix $\mathbf{\Psi}^{(t)}$ can be computed using a simple recursive procedure. In the base case, the leaves, there are no constraints. At each branch owned by an agent P_i, we need to bound the probability of each undesirable outcome in terms of the honest outcome $\mathbf{u}^*$. To do so, we compute the *t-inducible region*, defined as the set of outcomes inducible by a coalition C containing P_i of size $\leq t$. For each outcome $\mathbf{v}$ in the t-inducible region, we add a column $\boldsymbol{\psi} \in \mathbb{R}^{nm}$ to $\mathbf{\Psi}^{(t)}$ that ensures that $\mathbf{u}_i^* \geq \mathbf{v}_i + \delta$. To do so, suppose $\mathbf{u}_i^*$ and $\mathbf{v}_i$ have indices a, b respectively, we then let $\boldsymbol{\psi}_{im+a} \leftarrow 1$, and $\boldsymbol{\psi}_{im+b} \leftarrow -1$ and zero elsewhere, and add an entry containing δ to $\boldsymbol{\delta}^{(t)}$. This procedure can be completed using a single pass of the game tree by keeping track of the t-inducible region as we go along. Note that there is a technical issue since our decision variables $\mathbf{\Lambda}$

are not in vector form, as is usual of linear programming. To remedy this, we also want to collect the deposits in a vector $\boldsymbol{\lambda} \in \mathbb{R}^{ns}$ in row-major order. For a given information structure $\langle \Sigma, \boldsymbol{\Phi} \rangle$, we construct a matrix $\mathbf{R}$ equivalent to $\boldsymbol{\Phi}$ in the following way: for every index ij in $\boldsymbol{\Lambda}$, we construct the 'base matrix' $\mathbf{L}^{ij}$ that is 1 in index ij, and 0 everywhere else. We then compute a row of $\mathbf{R}$ by computing $\mathsf{vec}(\mathbf{L}^{ij}\boldsymbol{\Phi})$ as the product $\mathbf{L}^{ij}\boldsymbol{\Phi}$ put in row-major order. It is not hard to see that the image of $\boldsymbol{\Phi}$ is isomorphic to the column space of $\mathbf{R}$, and hence we say $\boldsymbol{\lambda}$ implements the utility vector $\mathbf{e}$ iff $\mathbf{e} = \mathbf{u} - \mathbf{R}\boldsymbol{\lambda}$. We now substitute this in Eq. (5.2) to get $\boldsymbol{\Psi}^{(t)}(\mathbf{u} - \mathbf{R}\boldsymbol{\lambda}) \geq \boldsymbol{\delta}^{(t)}$. Next, we move the constant terms to the right-hand side to yield the following:

$$-\boldsymbol{\Psi}^{(t)}\mathbf{R}\boldsymbol{\lambda} \geq \boldsymbol{\delta}^{(t)} - \boldsymbol{\Psi}^{(t)}\mathbf{u} \tag{5.3}$$

We also have to ensure the payment scheme is budget balanced, but this is a linear property and thus can be expressed as a set of linear constraints $\sum_{i=1}^{n} \lambda_{is+k}$ for every $k = 1 \dots s$. Finally, note that our objective function is $\mathbf{c}^\top \boldsymbol{\lambda}$, and since all constraints are linear we can produce the following linear program.

$$\begin{aligned} \min \quad & \mathbf{c}^\top \boldsymbol{\lambda} \\ \text{s.t.} \quad & -\boldsymbol{\Psi}^{(t)}\mathbf{R}\boldsymbol{\lambda} \geq \boldsymbol{\delta}^{(t)} - \boldsymbol{\Psi}^{(t)}\mathbf{u} \\ & \sum_{i=1}^{n} \lambda_{is+k} \geq 0 \qquad\qquad \forall\, k = 1 \dots s \end{aligned}$$

To deal with ∞ in the cost function, we may set the corresponding cost of the linear program to an arbitrary value and add an equality constraint to ensure the decision variable equals zero. Note that the linear program can be constructed by maintaining a constant set of pointers to the game given as input, which concludes the proof. □

Note that we can add additional linear constraints to the resulting program to impose contraints on the resulting payments. This can be used to implement strong budget balance (replace ≥ 0 with $= 0$), individual rationality (ensure $\mathbf{u}^* \geq 0$), envy freeness, or honest invariance.

Lower Bound: Reduction from LP

We now show how to reduce LP to $\mathsf{PaymentScheme}_1^0$ using logarithmic space. The resulting game is a two-agent finite game of perfect information. The reduction can easily be adapted to any $\delta \geq 0, t \geq 1$. Consider an arbitrary instance of LP, $\{\min \mathbf{c}^\top \mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where $\mathbf{c} = (\mathbf{c}_i) \in \mathbb{R}^n, \mathbf{A} = (\mathbf{a}_{ij}) \in \mathbb{R}^{m \times n}$, and $\mathbf{b} = (\mathbf{b}_i) \in \mathbb{R}^m$. Without loss of generality, we will assume that the columns of $\mathbf{A}$ have a positive column sum. This can be achieved by shifting $\mathbf{A}$ and $\mathbf{b}$ correspondingly.

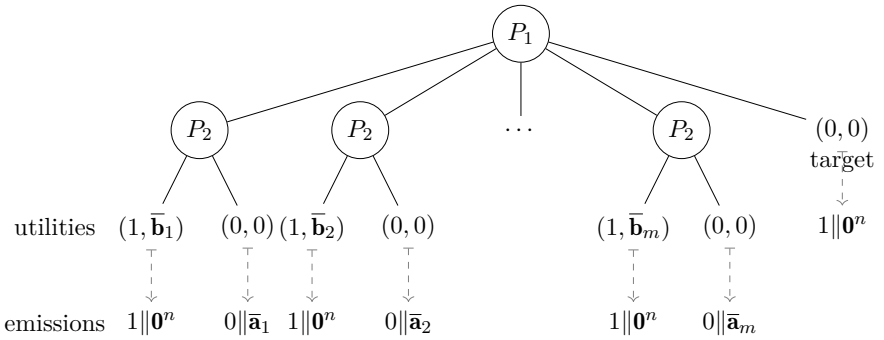

Figure 5.2: Depiction of the reduction from LP to $\mathsf{PaymentScheme}_1^0$. The dashed arrows depict the corresponding information structure (pdf for each leaf). The agent P_1 wants to sabotage satisfiability of the circuit and gains 1 utility for doing so (0 otherwise). The agent P_2 will sabotage the i^{th} gadget (and hence allow P_1 to win) if and only if the i^{th} inequality is not satisfied. A payment scheme corresponds to an assignment of the variables in the LP-instance, with emission probabilities proportional to the weights, such that an equilibrium with the target in its support corresponds to a satisfying assignment of the variables.

Proposition 5.11. *LP can be reduced to* $\mathsf{PaymentScheme}_1^0$ *in logspace.*

Proof. At a high level, the reduction is as follows. We first describe the game, and afterwards derive a suitable information structure. The game consists of two agents P_1, P_2. The root of the game consists of a move for agent P_1 who wants to 'sabotage' satisfaction of the program. They get utility 1 if they sabotage an inequality, and 0 otherwise. They are allowed to choose between a set of m gadgets, one for each inequality $\mathbf{a}_i^\top \mathbf{x} \geq \mathbf{b}_i$. In addition, they can choose a 'target' leaf that pays 0 to both agents. Each gadget consists of a move for the other agent P_2 who can choose whether to satisfy their inequality or not. If they sabotage their inequality (move 'left') they get $\mathbf{b}_i$ utility, otherwise if they move 'right' they get 0 utility. See Fig. 5.2 for an illustration. The SPE of the game is for P_2 to move left in the i^{th} gadget if $\mathbf{b}_i > 0$, and for P_1 to choose any convex combination of the gadgets for which the agents move left. Our goal is to design an information structure for which a payment scheme can ensure that P_1 chooses the target if and only if all inequalities are satisfied.

We now describe the information structure of the game. We have to specify an alphabet and a pdf for each leaf of the game. We will have $\Sigma = \{\top, \bot_1, \bot_2, \ldots \bot_n\}$, where $\top$ means 'all inequalities are satisfied', while $\bot_i$ is associated with the decision variable $\mathbf{x}_i$. When P_2 satisfies their inequality, the symbol $\top$ is outputted with probability 1. When P_2 sabotages their inequality, the column $\overline{\mathbf{a}_i}$ is used a pdf to sample the symbols $\{\mathbf{x}_i\}_{i=1}^n$. Of course, $\mathbf{a}_i$ is not necessarily a pdf, but we can normalize it by defining $\overline{\mathbf{a}}_{ij} = \frac{\mathbf{a}_{ij}}{\sum_{k=1}^n \mathbf{a}_{ik}}$. We

similarly define $\overline{\mathbf{b}}_i = \frac{\mathbf{b}_i}{\sum_{k=1}^{n} \mathbf{a}_{ik}}$ and use $\overline{\mathbf{b}}_i$ in lieu of $\mathbf{b}_i$ in the gadgets. This operation is well-defined since $\mathbf{A}$ was assumed to have positive column sums, and inequalities are preserved under positive scaling. To summarize, when an agent goes left, the corresponding pdf is $[1, 0, 0, \ldots, 0]^\top \in \mathbb{R}^{n+1}$, and when an agent goes right, the pdf is $0 \| \overline{\mathbf{a}} \in \mathbb{R}^{n+1}$. As the intended strategy profile s^*, we consider any strategy profile where P_2 always move right and P_1 chooses an arbitrary gadget. The cost function $\hat{\mathbf{c}}$ of the payment scheme will be defined as follows,

$$\hat{\mathbf{c}} := [\overbrace{\infty, \infty, \ldots, \infty}^{n+1 \text{ terms}}, \infty, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]^\top \in \mathbb{R}^{2n+2},$$

Now, suppose $\mathbf{\Lambda} \in \mathbb{R}^{(n+1)\times(m+1)}$ is output as an optimal payment scheme. Let $\mathbf{\Lambda}_{\bullet i}$ denote the i^{th} *row* of $\mathbf{\Lambda}$ (as a column vector) corresponding to the leaf where P_i goes right, and let $\mathbf{\Phi}_i = (0 \| \overline{\mathbf{a}}_i)$ be the column of $\mathbf{\Phi}$ corresponding to going right in the i^{th} gadget. Now, since some of the weights are ∞, we know that $\mathbf{\Lambda}_1 = \mathbf{0}$ and $\mathbf{\Lambda}_{\bullet 1} = \mathbf{0}$. Hence, the utility vector going right remains $[1, 0, 0, \ldots, 0]^\top$ for each gadget, and the utility for P_1 remains unchanged. By optimality and since $\delta = 0, t = 1$, we know that s^* must an SPE. This means that P_2 must receive (at least) as much utility going left as they do going right (otherwise P_1 would not hit the target). Then by Eq. (5.1), we must have,

$$\forall i.\, (-\mathbf{\Phi}_i^\top \mathbf{\Lambda}_{\bullet 2} \geq \overline{\mathbf{b}}_i) \quad \iff \quad \forall i.((0 \| \overline{\mathbf{a}}_i)^\top (-\mathbf{\Lambda}_{\bullet 2}) \geq \overline{\mathbf{b}}_i \quad \iff \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}$$

where $\mathbf{x} := [\mathbf{\Lambda}_{i2}]_{i=1}^{m}$ is the vector consisting of the non-zero (last m) entries of $\mathbf{\Lambda}_{\bullet 2}$. This means that s^* is an SPE if and only if the inequalities are satisfied. We know further that $\mathbf{x} \geq \mathbf{0}$ since the payment scheme is budget balanced. Minimization of the objective function $\mathbf{c}^\top \mathbf{x}$ comes directly from minimization of $\hat{\mathbf{c}}^\top \text{vec}(\mathbf{\Lambda})$, as some of the weights are ∞. Finally, note that all parts of the reduction can be performed by maintaining a constant set of pointers, thus concluding the proof. □

Removing ∞. To remove ∞ from the optimization problem, we may add an additional dummy agent P_3 who provides the necessary 'liquidity' to P_2, while ensuring P_1's utility is left unchanged (note that the payment scheme must be budget balanced, i.e. column sums of $\mathbf{\Lambda}$ must be non-negative). We assign to P_3 arbitrary utilities in the reduction, and assign to the payments of P_3 the opposite weights given to P_2, i.e. $-\mathbf{c}_i$ instead of $\mathbf{c}_i$. The weights given to the payments of P_1 are all zero. It is not hard to see that the resulting payment scheme has the same set of optimal values, as optimization problems are invariant under scaling. In addition, all payments to P_1 must be zero as any solution with non-zero payments to P_1 are strictly dominated by assigning the payment to either P_2 or P_3 if the corresponding weights are non-zero. If instead, the corresponding weights of P_2, P_3 are both zero, we can slightly perturb the cost of P_1 to e.g. 1 to ensure the utility of P_1 is unchanged.

5.3 Case Study: Secure Rational MPC from PVC

In this section, we apply our framework to a more complicated scenario involving secure multiagent computation (MPC). Our work is similar to [102], in that we also use payments to incentivize honesty from a PVC protocol. Their work focuses mainly on the cryptographic modeling, while our focus is mainly game-theoretic and thus complements their work. We start with a brief and informal definition of MPC for the purpose of self-containment, and refer to [74] for more details and formal definitions.

Secure Multi-agent Computation (MPC). In MPC, a set of n mutually distrusting agents $P_1, P_2, \ldots, P_n$ want to compute a public function f on their private data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The agents engage in an interactive protocol that ends with each of them producing an output y_i. The goal is for the output to be *correct* such that $y_i = f(\mathbf{x})$, and *private*, meaning the protocol leaks no information about the inputs of the agents, other than that which can be gathered from the function output itself. This should hold even if a coalition of t agents are controlled by a monolithic adversary who tries to break security of the protocol. MPC is a large research area with many proposed protocols, depending on the assumptions. One of the weakest notions of security is that of *passive security* where correctness and privacy are guaranteed against an 'honest-but-curious' adversary, who adheres honestly to the protocol description but tries to collect more information than they should. Such protocols are typically comparatively cheap, in contrast to protocols with *active security* that remain secure even if the adversary may deviate arbitrarily from the protocol description. Active protocols are typically orders of magnitude more expensive than their passive counterparts.

To remedy this, Aumann and Lindell [16] propose an intermediate notion of security called *covert security* where the adversary is allowed to cheat, but is caught with some constant non-zero probability. They propose three different definitions, giving different power to the adversary. The weakest notion is 'failed simulation' where the adversary learns the inputs of the honest agents when caught, while the strongest is called 'strong explicit cheat formulation' where they do not. In the present section, we opt for the latter, though our model easily adapts to the former albeit with larger payments. A disadvantage of covert secure protocols is that they do not allow the participants to convince a third agent who was dishonest which means they are not directly applicable to our setting. This was augmented to *publicly verifiable covert security* (PVC) by Asharov and Orlandi [12] where a proof of cheating is output that can be verified by a third agent. The underlying assumption of these protocols is that the adversary suffers some cost from being caught, meaning it is rational for them not to cheat. The typical use-case is that of competing businesses who may wish to perform some joint computation on trade secrets but are not willing to risk tarnishing their name. While this may be a reasonable

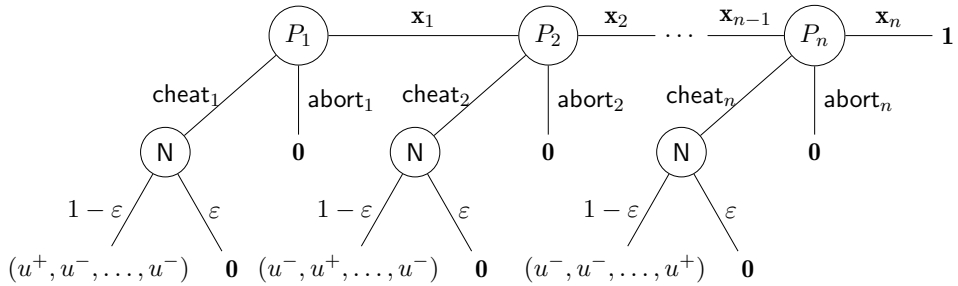

Figure 5.3: The ideal functionality $\mathcal{F}_{\mathsf{PVC}}$ with the strong explicit cheat formulation represented as an extensive-form game G_{PVC} rooted at P_1. Our goal is to augment the functionality with a payment scheme such that it is rational to behave honestly. Note that in this representation, for clarity there are two distinct leaves when an agent P_i attempts cheating, though in the following we 'merge' the two nodes belonging to nature for simplicity.

assumption in many cases, it is unclear that this works in e.g. an anonymous setting where the agents cannot be held accountable. Instead, we will use a payment scheme to prove it is rational for the agents not to cheat. Our plan is to analyze the information structure induced by the definition of covert security. We then apply our payment schemes to the resulting game and derive values for the deposits of the agents.

Secure Rational MPC from PVC

We consider a set of n agents $P_1, P_2, \ldots, P_n$ interacting with the ideal PVC functionality $\mathcal{F}_{\mathsf{PVC}}$. To analyze the interaction using game theory, we need to be able to give some bounds on the utilities of the agents. In order to simplify the presentation, we assume the agents are homogeneous, in that they have the same utility functions. We further disregard the cost of running the protocol, e.g. transaction fees, such that any abort_i gives 0 utility to all agents. Note that we can always normalize the utilities in a game as this preserves the total order. As such, we assume an agent receives 1 utility if they send their input and receive back the correct output. If instead an agent cheats and is successful, they receive u^+ utility, while an agent whose input is revealed receives u^- utility. As $\mathcal{F}_{\mathsf{PVC}}$ does not explicitly punish agents who are caught cheating, we assume an agent who is caught cheating receives 0 utility. As in [12], we are using the strong explicit cheat formulation from [16], so a cheater who is caught does not learn the inputs of the honest agents, and as such earns 0 utility. We are not modeling the fact that agents can send incorrect inputs, for the simple reason that it is impossible for the payment scheme, in general, to detect this. We assume that agents always send their input truthfully, or rather their true input is defined to be whatever they send to the functionality. The corresponding information structure would not be able

to distinguish the two classes of leaves, the distributions would be linearly dependent, making it impossible to instantiate the deposits to ensure security. For some specific applications however, one could imagine a function that allows the determination of an agent providing the wrong input. It is possible to augment our model to accommodate this scenario, though it is out of scope for the present paper.

To make the problem nontrivial, we require that $u^+ > 1 > 0 > u^-$. Note that we are assuming the agents are oblivious to the utility earned by other agents. This is in contrast to [130] who assume agents strictly prefer that as few other agents learn the output as possible. This is not to circumvent their impossibility result, as this is accomplished by quasi-linearity by allowing the deposits to alter the total order of outcomes. Rather, it is for simplicity of exposition, though it would be interesting as future work to augment our model to a setting where we explicitly specify how utility an agent loses by other agents also learning the output. We represent the interaction as an extensive-form game G_{PVC}, and draw the corresponding tree. An illustration of the game tree can be found in Fig. 5.3. Observe that when $(1-\varepsilon)\, u^+ > 1$, the only equilibrium in the game is for P_1 to cheat. Instead, we want all agents to play honestly. First, we need to define an information structure on the game. We first remark that the structure of the game is such that only one agent can deviate in any given strategy profile. This means we can define the following alphabet of possible outcomes as, $\Sigma = \{\top, \mathsf{abort}_1, \mathsf{cheat}_1, \mathsf{abort}_2, \mathsf{cheat}_2, \ldots, \mathsf{abort}_n, \mathsf{cheat}_n\}$. We assume the symbols are ordered left-to-right. Here $\top$ is a symbol emitted when no cheating was detected, and no aborting occurred. Note that this overloads the notation of abort_i and cheat_i. We now analyze the information structure induced by the functionality. For simplicity, we will slightly modify the game tree in Fig. 5.3. Namely, we collapse each subgame corresponding to a move by nature into a single leaf with expected utility $(1-\varepsilon)\, u^+$. This allows us to write a single pdf for that leaf. If we instead insist on having separate leaves, then the columns are no longer linearly independent; hence Lemma 5.6 does not apply directly; however, it still applies if we replace 'the inverse' with 'a left inverse'. This needlessly complicates the analysis, hence the simplifying assumption. If all agents are honest, we reach the outcome $\mathbf{1}$ and the symbol $\top$ is emitted. If some agent P_i aborts, the output of the honest agents will always be abort_i. If instead, an agent attempts to cheat, with probability ε they are caught and the message cheat_i is output. If they are not caught, the symbol $\top$ is also emitted. Suppose the leaves of G_{PVC} are ordered left-to-right in Fig. 5.3, then we can write the information structure as follows. Note that when $\varepsilon > 0$, all columns are linearly independent, and as such $\mathbf{\Phi}_{\mathsf{PVC}}$ is invertible.

$$\mathbf{\Phi}_{\mathtt{PVC}} = \begin{pmatrix} 0 & 1-\varepsilon & 0 & 1-\varepsilon & \cdots & 0 & 1-\varepsilon & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \varepsilon & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \frac{1}{\varepsilon} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\varepsilon} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \frac{1}{\varepsilon} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & \frac{\varepsilon-1}{\varepsilon} & 0 & \frac{\varepsilon-1}{\varepsilon} & 0 & \cdots & \frac{\varepsilon-1}{\varepsilon} & 0 \end{pmatrix}^{-1}$$

By Lemma 5.6, we can implement any utility matrix $\mathbf{E}$. In order to obtain $(\delta+1)$-strong game-theoretic security we could for instance define the following:

$$\mathbf{E} = \begin{pmatrix} -\delta & -\delta & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & -\delta & -\delta & \cdots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\delta & -\delta & 1 \end{pmatrix}$$

Any agent who deviates gains an expected utility of $-\delta$, while they gain 1 utility by following the strategy honestly. Note that $\mathbf{u}^* = \mathbf{1}$, hence the resulting equilibrium will be individually rational. Also, we only consider deviations by a single agent as it is not possible for multiple agents to cheat in our model. In addition, the utility matrix satisfies honest invariance, in that the utility of the honest strategy profile remains unchanged for all agents. In order to compute the deposits, we again apply Lemma 5.6 and compute the appropriate payments:

$$\mathbf{\Lambda}_{\mathtt{PVC}} = (\mathbf{U} - \mathbf{E})\, \mathbf{\Phi}_{\mathtt{PVC}}^{-1} = \begin{pmatrix} 0 & \frac{u^+ + \delta}{\varepsilon} & \delta & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^+ + \delta}{\varepsilon} & \delta & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^+ + \delta}{\varepsilon} & \delta \end{pmatrix}$$

We now briefly analyze the resulting payment scheme. We note that when $\top$ is emitted, all agents are repaid their deposits in full. When the symbol abort_i is emitted, the agent P_i loses part of their deposit, while all other agents are repaid their deposit in full. Finally, when cheat_i is emitted, the agent P_i loses $\frac{u^+ + \delta}{\varepsilon}$, while each P_j for $j \neq i$ loses $\frac{u^-}{\varepsilon}$. Note that we assume $u^- < 0$, meaning P_i actually *gains money* from the payment scheme, i.e. receive back more than they initially deposited. In order for the payment scheme to not mint new money, we need that $\frac{u^+ + \delta}{\varepsilon} \geq -\frac{(n-1)\,u^-}{\varepsilon}$. That is, we must have that $\delta \geq -(u^+ + (n-1)\,u^-) \geq 0$ for the payment scheme to be budget balanced. In other words, there is only sufficient funds left over to compensate the honest agents, if the desired level of security is sufficiently high (and hence the deposits are large). As $\varepsilon < 1$ and $\delta \geq 0$, we have $\frac{u^+ + \delta}{\varepsilon} > \delta$. This means we get a deposit of size $\lambda_i^* = \frac{u^+ + \delta}{\varepsilon}$. Note that the argument is fairly easy to adapt to the non-homogeneous setting, where we would instead get $\lambda_i^* = \frac{u_i^+ + \delta}{\varepsilon}$, where u_i^+ is the utility gained by agent P_i when successful in cheating. This shows the following result.

Theorem 5.12 (Rational MPC)**.** *Let f be a public function and let $P_1, P_2, \ldots, P_n$ be a set of rational agents with the following utility function: namely, each P_i earns 1 utility by learning the output of the function, and u_i^+ utility from learning the inputs of the other agents, while they gain u_i^- utility from another agent learning their input. Then for sufficiently large δ, f can be computed with δ-strong game-theoretic security with black-box access to any ε-deterrent PVC protocol, using a budget balanced payment scheme where P_i makes a deposit of size $(u_i^+ + \delta - 1)/\varepsilon$.*

In the next section, we show a general lower bound on the maximum deposit of any budget balanced payment scheme, namely of size $\Omega(1 + \delta\sqrt{n}/|\Sigma|)$. Note that this matches asymptotically the deposits in our MPC protocol, assuming the PVC protocol is fixed (and hence $\varepsilon, n, |\Sigma|$ are all constant).

5.4 A Lower Bound on the Size of Payments

In this section we prove a lower bound on the size of the largest payment necessary to achieve game-theoretic security. We show that the largest deposit must be linear in the security parameter ε, as well as linear in some of the utilities in the game. To establish our bound, we use properties of matrix norms. We give a brief recap of matrix norms for the purpose of self-containment and refer to [119] for more details. We say a mapping $\|\cdot\| : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a *matrix norm* if it satisfies the following properties for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$, and every scalar $\alpha \in \mathbb{R}$.

1. (*Positivity*). $\|\mathbf{A}\| \geq 0$, and $\|\mathbf{A}\| = 0$ iff $\mathbf{A} = \mathbf{0}$.

2. (*Homogeneity*). $\|\alpha\mathbf{A}\| = |\alpha| \, \|\mathbf{A}\|$.

3. (*Subadditivity*). $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$.

We denote by $\|\cdot\|_p$ the matrix norm induced by the L_p norm $\|\cdot\|_p$ on vector spaces, and is defined as

$$\|\mathbf{A}\|_p = \sup_{\mathbf{x} \neq \mathbf{0}, \|\mathbf{x}\|_2 = 1} \|\mathbf{A}\mathbf{x}\|_p$$

If in addition, it holds that,

$$\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\| ,$$

we say $\|\cdot\|$ is *submultiplicative*. It can be shown that $\|\cdot\|_p$ is submultiplicative for any value of p. Some special cases that we will need are $p = 1, 2, \infty$ which can be characterized as follows. The quantity $\|\mathbf{A}\|_1$ equals the maximum absolute column sum of the columns of $\mathbf{A}$, while the quantity $\|\mathbf{A}\|_\infty$ gives the maximum absolute row sum of the rows of $\mathbf{A}$. Our lower bound is established by noting that we know these sums for the matrices used in our framework. An example of a matrix norm that is not submultiplicative is the max norm, $\|\mathbf{A}\|_{\max} = \max_{i,j} |\mathbf{A}_{ij}|$. We will need the fact that all matrix norms are equivalent up to scalar multiple, in the sense that each pair of matrix norms $\|\cdot\|_a , \|\cdot\|_b$ are related by $\alpha \|\mathbf{A}\|_a \leq \|\mathbf{A}\|_b \leq \beta \|\mathbf{A}\|_a$, for some constants $\alpha, \beta \in \mathbb{R}$. For our purposes, we need the following bounds:

$$\frac{\|\mathbf{A}\|_2}{\sqrt{mn}} \leq \|\mathbf{A}\|_{\max} \leq \|\mathbf{A}\|_2 \tag{5.4}$$

$$\frac{1}{\sqrt{m}} \|\mathbf{A}\|_1 \leq \|\mathbf{A}\|_2 \leq \sqrt{n} \, \|\mathbf{A}\|_1 \tag{5.5}$$

$$\frac{1}{\sqrt{n}} \|\mathbf{A}\|_\infty \leq \|\mathbf{A}\|_2 \leq \sqrt{m} \, \|\mathbf{A}\|_\infty \tag{5.6}$$

Establish the lower bound. Let G be a fixed game with information structure $\langle \Sigma, \mathbf{\Phi} \rangle$. Let (δ, t) be fixed, and let $\mathbf{\Psi}^{(t)}, \boldsymbol{\delta}^{(t)}$ be the corresponding constraints. We denote by $\alpha^{(t)}$ the number of rows in $\mathbf{\Psi}^{(t)}$. Now, let $\mathbf{\Lambda}$ be any feasible payment scheme. We have already seen that any such $\mathbf{\Lambda}$ is a solution to the following equation:

$$\mathbf{\Psi}^{(t)} \mathbf{\Lambda} \mathbf{\Phi} \leq \mathbf{\Psi}^{(t)} \mathbf{U} - \boldsymbol{\delta}^{(t)} \tag{5.7}$$

Applying Eqs. (5.4) and (5.7) and the properties of $\|\cdot\|_2$, we establish the following bound:

$$\|\mathbf{\Lambda}\|_{\max} \geq \frac{1}{\sqrt{n \, |\Sigma|}} \left(\frac{\left\| \mathbf{\Psi}^{(t)} \mathbf{U} \right\|_2 + \left\| \boldsymbol{\delta}^{(t)} \right\|_2}{\left\| \mathbf{\Psi}^{(t)} \right\|_2 \cdot \|\mathbf{\Phi}\|_2} \right) \tag{5.8}$$

Each row of $\boldsymbol{\delta}^{(t)}$ is filled with δ, so the resulting absolute row sum is δn. Similarly, each row of $\boldsymbol{\Psi}^{(t)}$ contains exactly one 1 and one -1, so each absolute row sum is 2. Finally, each column of $\boldsymbol{\Phi}$ is a pdf, so its absolute row sum is 1. Combining these insights with Eqs. (5.5) and (5.6) and substituting in Eq. (5.8) gives the following bound:

$$\|\boldsymbol{\Lambda}\|_{\max} \geq \frac{1}{\sqrt{n\,|\Sigma|}} \left(\frac{\left\|\boldsymbol{\Psi}^{(t)}\mathbf{U}\right\|_2 + \sqrt{\alpha^{(t)}}\delta n}{2\sqrt{\alpha^{(t)}} \cdot \sqrt{|\Sigma|}} \right) = \frac{1}{2\,|\Sigma|} \left(\delta\sqrt{n} + \frac{\left\|\boldsymbol{\Psi}^{(t)}\mathbf{U}\right\|_2}{\sqrt{n\,\alpha^{(t)}}} \right)$$

We note that in general, there is not much to say about $\left\|\boldsymbol{\Psi}^{(t)}\mathbf{U}\right\|_2$, as $\mathbf{U}$ can lie in the kernel of $\boldsymbol{\Psi}^{(t)}$. This occurs if $\mathbf{U}$ already establishes exact δ-strong t-robust game-theoretic security.

Note that the bound, strictly speaking, is a bound on the largest *absolute* deposit necessary to achieve security, while we are interested in bounding the largest *positive* deposit, denoted instead by $\boldsymbol{\Lambda}^*_{\max}$. If the game already is secure, the bound for the largest deposit should be zero, while the above bound is positive for any $\delta > 0$. Indeed, $\|\boldsymbol{\Lambda}\|_{\max} \neq \boldsymbol{\Lambda}^*_{\max}$ iff we can pay more to an agent to misbehave *and still retain security* than what we have to pay another agent to behave properly. We note that this depends on the structure of the game and the intended strategy profile. In particular, it is independent of the security parameter. For this reason, we denote by $\Delta^{(t)}_G(s^*)$ the minmax deposit required to obtain 0-strong t-robust game-theoretic security. We note that $\Delta^{(t)}_G > 0$ iff the game is not secure for any $\delta \geq 0$, while $\Delta^{(t)}_G \leq 0$ iff the game is already secure for $\delta = 0$. We note that by definition, $\Delta^{(t)}_G$ is a trivial lower bound on the size of the maximum deposit. We combine this with the above bounds to yield the following lower bound:

Theorem 5.13. *Let G be a game on n agents with an information structure $\langle \Sigma, \boldsymbol{\Phi} \rangle$, and let s^* be the intended strategy profile. If $\boldsymbol{\Lambda}$ is budget balanced and ensures δ-strong t-robust game-theoretic security, then the maximum deposit must satisfy $\boldsymbol{\Lambda}^*_{max} \geq \Delta^{(t)}_G(s^*) + \Omega\left(\frac{\delta\sqrt{n}}{|\Sigma|}\right)$.*

Chapter 6

Commitments

> "*Mr. President, it is not only possible, it is essential. That is the whole idea of this machine, you know. Deterrence is the art of producing in the mind of the enemy... the fear to attack. And so, because of the automated and irrevocable decision making process which rules out human meddling, the doomsday machine is terrifying. It's simple to understand. And completely credible, and convincing.*"
>
> Dr. Strangelove

We now turn to consider an important aspect of deploying smart contracts in practice on permissionless blockchains such as Ethereum [258]. In these permissionless systems, agents can themselves deploy smart contracts without prior authorization by buying the tokens required to execute the contract. This changes fundamental game-theoretic assumptions about rationality: in particular, it might be rational for an agent to deploy a contract that commits them to act irrationally in certain situations to make credible otherwise non-credible threats. This gives rise to complex games in which agents can commit to strategies, that in turn depend upon other agents' committed strategies. Reasoning about such equilibria is important when considering games that are meant to be played on a blockchain, since the agents - at least in principle - always have the option of deploying such contracts. In the literature, this is known as a Stackelberg equilibrium where a designated leader commits to a strategy before playing the game. In general, because of first-mover advantage, being able to deploy a contract first is never a disadvantage, since an agent can choose to deploy the empty contract that commits them to nothing. It is well-known that it is hard to compute the Stackelberg equilibrium in the general case [171], though much less is known about the complexity when there are several of these contracts in play: when there are two contracts, the first contract can depend on the second contract in what is known as a reverse Stackelberg equilibrium [22, 135, 243].

This is again strictly advantageous for the leader since they can punish the follower for choosing the wrong strategy. In this chapter, we present a model that generalizes (reverse) Stackelberg games, which we believe captures these types of games and which may be of wider interest. In practical terms, we believe that our model is of interest when analyzing distributed systems for game-theoretic security in settings where the agents naturally have the ability to deploy smart contracts.

Attribution. This chapter is based entirely on the paper [129], with most of the text (including this introduction) taken (almost) verbatim from [129], with only minor modifications to the formatting and the prose. Fig. 6.3 was partially redrawn for this thesis. Algorithm 2 was reformatted.

Our Results

We propose a game-theoretic model for games in which agents have shared access to a blockchain that allows the agents to deploy smart contracts to act on their behalf in the games. Allowing an agent to deploy a smart contract corresponds to that agent making a 'cut' in the tree, inducing a new expanded game of exponential size containing as subgames all possible cuts in the game. We show that many settings from the literature on Stackelberg games can be recovered as special cases of our model, with one contract being equivalent to a Stackelberg equilibrium, and two contracts being equivalent to a reverse Stackelberg equilibrium. We prove bounds on the complexity of computing an SPE in these expanded trees. We prove a lower bound, showing that computing an SPE in games of imperfect information with k contracts is Σ_k^{P}-hard by reduction from the true quantified Boolean formula problem. For $k = 1$, it is easy to see that a contract can be verified in linear time, establishing NP-completeness. In general, we conjecture Σ_k^{P}-completeness for games with k contracts, though this turns out to reduce to whether or not contracts can be described in polynomial space. For games of perfect information with an unbounded number of contracts, we also establish PSPACE-hardness from a generalization of 3-COLORING. We show an upper bound for $k = 2$ and perfect information, namely that computing an SPE in a two-contract game of size m with ℓ terminal nodes (and any number of agents) can be computed in time $O(m\ell)$. For $k = 3$, the problem is clearly in NP since we can verify a witness using the algorithm for $k = 2$, and we conjecture the problem to be NP-complete.

Smart Contract Moves

We now give our definition of smart contracts in the context of finite games. We add a new type of node to our model of games, a *smart contract move*. Intuitively, whenever an agent has a smart contract move, they can deploy

Contracts	agents	Information	Strategies	Lower bound	Upper bound
0	2	perfect	pure	P-hard [239]	$O(m)$ [194]
0	2	imperfect	mixed	PPAD-complete [61, 81]	
1	2	perfect	pure	P-hard [239]	$O(m\ell)$ [41]
1	2	perfect	mixed	NP-complete [172]	
1	2	imperfect	-	NP-complete [172]	
2	2	perfect	pure	P-hard [239]	$O(m\ell)$ [Theorem 6.9]
3	3	perfect	pure	Conjectured NP-hard	NP [Theorem 6.9]
k	$2+k$	imperfect	pure	Σ_k^p-hard [Theorem 6.7]	?
unbounded	-	perfect	pure	PSPACE-hard [Theorem 6.10]	?

Figure 6.1: An overview of some existing bounds on the complexity of computing an SPE in extensive-form games and where our results fit in. Here, m is the size of the tree, and ℓ is the number of terminal nodes.

a contract that acts on their behalf for the rest of the game. The set of all such contracts is countably infinite, but fortunately, we can simplify the problem by considering equivalence classes of contracts that "do the same thing". Essentially, the only information relevant to other agents is whether or not a given action is still possible to play: it is only if the contract dictates that a certain action cannot be played, that we can assume a rational agent will not play it. In particular, any contract which does not restrict the moves of an agent is equivalent to the agent not having a contract. Such a restriction is called a *cut*. A cut $c^{(i)}$ for agent P_i is defined to be a union of subtrees whose roots are children of P_i-nodes, such that: (1) every node in $T \setminus c^{(i)}$ has a path going to a leaf; a cut is not allowed to destroy the game by removing all moves for an agent, and (2) $c^{(i)}$ respects information sets, that is it 'cuts the same' from each node in the same information set.

In other words, deploying a smart contract corresponds to choosing a cut in the game tree. This means that a smart contract node for agent P_i in a game T is essentially syntactic sugar for the *expanded tree* that results by applying the set of all cuts $c^{(i)}$ to T and connecting the resulting games with a new node belonging to P_i at the top. Computing the corresponding equilibrium with smart contracts then corresponds to the SPE in this expanded tree. Note that this tree is uniquely determined. See Fig. 6.2 for an example. We use the square symbol in figures to denote smart contract moves. When a game contains multiple smart contract moves, we expand the smart contract nodes recursively in a depth-first manner using the transformation described above.

6.1 Contracts as Stackelberg Equilibria

As mentioned earlier, the idea to let an agent commit to a strategy before playing the game is not a new one: in 1934, von Stackelberg proposed a model for the interaction of two business firms with a designated market leader [251].

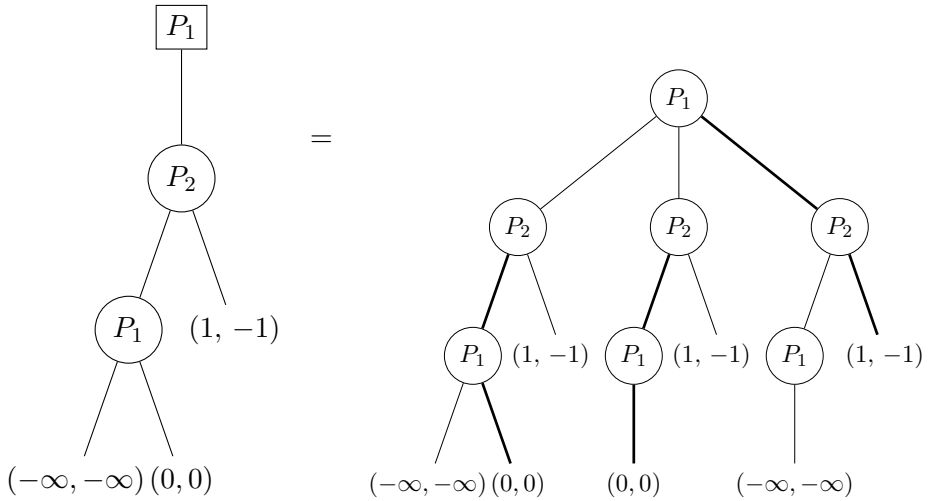

Figure 6.2: Expanding a smart contract node for a simple game. The square symbol is a smart contract move for agent P_1. We compute all P_1-cuts in the game and connect them with a node belonging to P_1. The first coordinate is the leader payoff, and the second is the follower payoff. The dominating paths are shown in bold. We see that the optimal strategy for P_1 is to commit to choosing $(-\infty, -\infty)$ unless P_2 chooses $(1, -1)$.

The market leader holds a dominant position and is therefore allowed to commit to a strategy first, which is revealed to the follower who subsequently decide on a strategy. The resulting equilibrium is called a Stackelberg equilibrium. In this section we show that the Stackelberg equilibrium for a game with leader P_1 and follower P_2 can be recovered as a special case of our model where P_1 has a smart contract. We use the definition of strong Stackelberg equilibria from [45, 167]. We note that since the games are assumed to be in generic form, the follower always has a unique response, thus making the requirement that the follower break ties in favor of the leader unnecessary.

Let T be a game tree. A *path* $\mathbf{p} \subseteq T$ is a sequence of nodes such that for each j, $\mathbf{p}_{j+1}$ is a child of $\mathbf{p}_j$. If $\mathbf{p}$ is a path, we denote by $\mathbf{p}^{(i)} \subseteq \mathbf{p}$ the subset of nodes owned by agent P_i. Now suppose T has a horizon of h. We let $\mathfrak{p} = (\mathfrak{p}_j)_{j=1}^{h} \subseteq T$ denote the *dominating path* of the game defined as the path going from the root $\mathfrak{p}_1$ to the terminating leaf $\mathfrak{p}_h$ in the SPE of the game.

Definition 6.1 (Stackelberg Equilibrium)**.** *Let $i \in [n]$ be the index of an agent, and let $f(s_i)$ be the best response to s_i for agents other than P_i. We say $(s_i^*, f(s_i^*))$ is a* Stackelberg equilibrium with leader P_i *if the following properties hold true:*

- Leader optimality. *For every leader strategy s_i, $u_i(s_i^*, f(s_i^*)) \geq u_i(s_i, f(s_i))$.*

- Follower best response. *For every $j \neq i$, and every s_{-i}, $u_j(s_i^*, f(s_i^*)) \geq u_j(s_i^*, s_{-i})$.*

Proposition 6.2. *The Stackelberg equilibrium with leader P_i is equivalent to P_i having a smart contract move.*

Proof. We show each implication separately:

$\Rightarrow$ An SPE in the expanded tree T induces a Stackelberg equilibrium in the corresponding Stackelberg game where P_i commits to all moves in $\mathfrak{p}^{(i)}$. It is not hard to see that the follower best response $f(s_i^*)$ is defined by the SPE of the subgame arising after P_i makes the move $\mathfrak{p}_1$ choosing the contract in T.

$\Leftarrow$ A Stackelberg equilibrium induces an SPE in the expanded tree T with the same utility: let $(s_i^*, f(s_i^*))$ be a Stackelberg equilibrium, observe that s_i^* corresponds to a cut $c^{(i)} \subseteq T$ where P_i cuts away all nodes in T not dictated by s_i^*. By letting the first move $\mathfrak{p}_1$ of P_i correspond to $c^{(i)}$, the best follower response $f(s_i^*)$ is the SPE in the resulting subgame, and hence $u(\mathfrak{p}) = u(s_i^*, f(s_i^*))$. □

Multi-leader/multi-follower contracts

Several variants of the basic Stackelberg game has been considered in the literature with multiple leaders and/or followers [177, 227]. We can model this using smart contracts by forcing some of the contracts to independent of each other: formally, we say a contract is *independent* if it makes the same cut in all subgames corresponding to different contracts. It is not hard to see that multiple leaders can be modelled by adding contracts for each leader, where the contracts are forced to be independent.

Reverse Stackelberg Contracts

The reverse Stackelberg equilibrium is an attempt to generalize the regular Stackelberg equilibrium: here, the leader does not commit to a specific strategy *a priori*, rather they provide the follower with a mapping f from follower actions to best response leader actions, see e.g. [17, 232] for a definition in the continuous setting. When the follower plays a strategy s_{-i}, the leader plays $f(s_{-i})$. This is strictly advantageous for the leader since as pointed out in [135], they can punish the follower for choosing the wrong strategy.

In the following, if $\mathbf{p}$ is a path of length ℓ, we denote by $G_s(\mathbf{p})$ the subgame whose root is $\mathbf{p}_\ell$.

Definition 6.3. *Let i be the index of the leader, and $-i$ the index of the follower. We say $(f(s_{-i}^*), s_{-i}^*)$ is a* reverse Stackelberg equilibrium with leader i *if the following holds for every leader strategy s_i and follower strategy s_{-i}, it holds:*

- Leader best response*:* $u_i(f(s^*_{-i}), s^*_{-i}) \geq u_i(s_i, s^*_{-i})$.
- Follower optimality*:* $u_{-i}(f(s^*_{-i}), s^*_{-i}) \geq u_{-i}(f(s_{-i}), s_{-i})$.

Proposition 6.4. *The reverse Stackelberg equilibrium for a two-agent game with leader P_i is equivalent to adding two smart contract moves to the game, one for P_i, and another for P_{-i} (in that order).*

Proof. We show each implication separately:

$\Rightarrow$ The SPE in the expanded tree induces a reverse Stackelberg equilibrium: for every possible follower strategy s_{-i}, we define $f(s_{-i})$ as the leader strategy in the SPE in the subgame $G_s(\langle \mathfrak{p}_1, s_{-i} \rangle)$ after the two moves, where we slightly abuse notation to let s_{-i} mean that P_{-i} chooses a cut where their SPE is s_{-i}. Leader best response follows from the observation that $\mathfrak{p}_1$ corresponds to the optimal set of cuts of P_i moves in response to every possible cut of of P_{-i} moves.

$\Leftarrow$ A reverse Stackelberg equilibrium induces an SPE in the expanded tree: let $(f(s^*_{-i}), s^*_{-i})$ be a reverse Stackelberg equilibrium and let f be the strategy of P_i in the reverse Stackelberg game, then P_i has a strategy in the two-contract game with the same utility for both agents: namely, P_i's first move is choosing the subgame in which for every second move s_{-i} by P_{-i} they make the cut $f(s_{-i})$. □

Having defined our model of games with smart contracts, we turn to study the computational complexity of computing equilibria in such games. This section is entirely based on [129], though figures/tables have been redrawn for this thesis. Note that we can always compute the equilibrium by constructing the expanded tree and performing backward induction in linear time. The problem is that the expanded tree is large: the expanded tree for a game of size m with a single contract has $2^{O(m)}$ nodes since it contains all possible cuts. For every contract we add, the complexity grows exponentially. This establishes the rather crude upper bound of Σ_k^{EXP} for computing SPE in games with perfect information and k contracts. The question we ask if we can do better than traversing the entire expanded tree.

In terms of feasibility, our results are mostly negative: we show a lower bound that computing an SPE, in general, is infeasible for games with smart contracts. We start by considering the case of imperfect information where information sets allow for a rather straightforward reduction from CircuitSAT to games with one contract, showing NP-completeness for single-contract games of imperfect information. This generalizes naturally to the k true quantified Boolean formula problem (k-TQBF), establishing Σ_k^{P}-hardness for games of imperfect information with k contracts. On the positive side, we consider games of perfect information where we provide an algorithm for games and two contracts that runs in time $O(m\ell)$. However, when we allow for

an unbounded number of contracts, we show the problem remains PSPACE-complete by reduction from the generalization of 3-COLORING described in [39]. We conjecture the problem to be NP-complete for three contracts.

6.2 Imperfect Information, One Contract, NP-completeness

We start by showing NP-completeness for games of imperfect information by reduction from CircuitSAT. We consider a decision problem version of SPE: namely, whether or not a designated agent can obtain a utility greater than the target value.

Reduction. Let C be an instance of CircuitSAT. Note that we can start from any complete basis of Boolean functions, so it suffices to suppose the circuit C consists only of NAND with fanin 2 and fanout 1. We will now construct a game tree for the circuit: we will be using one agent to model the assignment of variables, say agent 1. The game starts with a contract move for agent 1 who can assign values to variables by cutting the bottom of the tree: we construct the game such that agent 1 only has moves in the bottom level of the tree. In this way, we ensure that every cut corresponds to assigning truth values to the variables. We adopt the convention that a payoff of 1 for agent 1 is *true* ($\top$), while a payoff of 0 for agent 1 is *false* ($\bot$). All nodes corresponding to occurrences of the same variable get grouped into the same information set, which enforces the property that all occurrences of the same variable must be assigned the same value.

For the NAND-gate, we proceed using induction: let T^L, T^R be the trees obtained by induction, we now wish to construct a game tree gadget with NAND-gate logic. To do this we require two agents which we call agent 2 and agent 3. Essentially, agent 2 does the logic, and agent 3 converts the signal to the right format. The game tree will contain multiple different utility vectors encoding true and false, which vary their utilities for agents 2 and 3. Each NAND-gate has a left tree and a right tree, each with their own utilities for true and false: $\bot^L, \bot^R; \top^L, \top^R$. The gadget starts with a move for agent 3 who can choose to continue the game, or end the game with a true value $\top'$. If they continue the game, agent 2 has a choice between false $\bot'$ or playing either T^L or T^R. To make the gadget work like a NAND-gate we need to instantiate the utilities to make backward induction simulate its logic. The idea is to make agent 2 prefer both $\bot^L$ and $\bot^R$ to $\bot'$, which they, in turn, prefer to $\top^L$ and $\top^R$. As a result, agent 2 propagates $\bot'$ only if both T^L, T^R are true, otherwise, it propagates $\bot^L$ or $\bot^R$. Finally, we must have that agent 3 prefers $\top'$ to both $\bot^L$ and $\bot^R$, while they prefer $\bot'$ to $\top', \top^L$ and $\top^R$. This gives rise to a series

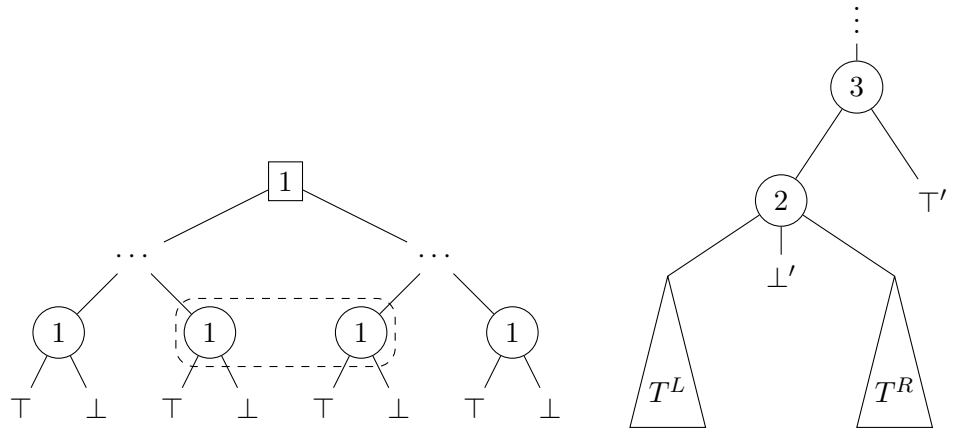

Figure 6.3: The basic structure of the reduction. agent 1 has a smart contract that can be used to assign values to the variables. The dashed rectangle denotes an information set and is used when there are multiple occurrences of a variable in the circuit. On the right, we see the NAND-gate gadget connecting the left subgame T^L and the right subgame T^R. We implement the gadget by instantiating the utility vectors such that agent 2 chooses $\bot'$ if only if both T^L and T^R propagate a utility vector encoding true.

of inequalities:

$$\begin{array}{llll} \bot_2^L > \bot_2' > \top_2^L & \top_3' > \bot_3^L & \bot_3' > \top_3^L & \bot_3' > \top_3' \\ \bot_2^R > \bot_2' > \top_2^R & \top_3' > \bot_3^R & \bot_3' > \top_3^R & \end{array}$$

We can instantiate this by defining $\top, \bot$. For the base case corresponding to a leaf, we let $\bot = (0, 1, 0), \top = (1, 0, 0)$. We then define recursively:

$$\begin{aligned} \top' &= \left(1, 0, 1 + \max(\top_3^L, \top_3^R)\right) \\ \bot' &= \left(0, \frac{\min(\bot_2^L, \bot_2^R) + \max(\top_2^L, \top_2^R)}{2}, 2 + \max\left(\top_3^L, \top_3^R\right)\right) \end{aligned}$$

It is not hard to verify that these definitions make the above inequalities hold true. As a result, the gadget will propagate a utility vector corresponding to true if and only if not both subtrees propagate true.

Theorem 6.5. *Computing an SPE in three-agent single-contract games of imperfect information is NP-complete.*

Proof. We consider the decision problem of determining whether or not in the SPE, agent 1 has a utility of 1. By construction of the information sets, any strategy is a consistent assignment of the variables. It now follows that agent 1 can get a payoff > 0 if and only if there is an assignment of the variables such that the output of the circuit is true. This shows NP-hardness. Now, it

is easily seen that this problem is in NP, since a witness is simply a cut that can be verified in linear time in the size of the tree. Completeness now follows using our reduction from CircuitSAT. □

Remark 6.6. *Our reduction also applies to the two-agent non-contract case by a reduction from circuit value problem. This can be done in logspace since all the gadgets are local replacements. In doing so, we reestablish the result of [239], showing that computing an SPE on two-agent games is P-complete.*

6.3 Imperfect Information, k Contracts, Σ_k^{P}-hardness

In this section, we show that computing the SPE in a game with k contract moves is Σ_k^{P}-complete, in the general case with imperfect information. Generalizing the previous result of NP-hardness to k contracts is fairly straightforward. Our claim is that the resulting decision problem is Σ_k^{P}-hard so we obtain a series of hardness results for the polynomial hierarchy. This is similar to the results obtained in [142] where the value problem for a competitive analysis with $k+1$ agents is shown to be hard for Σ_k^{P}.

Formally, we consider the following decision problem with target value V for a game tree T with k contract agents: let T' be the expanded tree with contracts for agents $P_1, P_2, \ldots P_k$ in ascending order. Can agent P_1 make a cut in T' such that their payoff is $\geq V$?

To show our claim, we proceed using reduction from the canonical Σ_k^{P}-complete problem k-TQBF, see e.g. [108] for a formal definition.

Theorem 6.7. *Computing an SPE in $2+k$ agent games of imperfect information is Σ_k^P-hard.*

Proof (sketch). We extend our reduction from Theorem 6.5 naturally to the quantified satisfiability problem. In our previous reduction, the contract agent wanted to satisfy the circuit by cutting as to assign values to the variables in the formula. Now, for each quantifier in ψ, we add a new agent with a contract, whose moves range over exactly the variables quantified over. The agents have contracts in the same order specified by their quantifiers. The idea is that agents corresponding to $\forall$ try to sabotage the satisfiability of the circuit, while those corresponding to $\exists$ try to ensure satisfiability. We encode this in the utility vectors by giving $\exists$-agents a utility of 1 in $\top$ and 0 utility in $\bot$, while for the $\forall$-agents, it is the opposite. It is not hard to see that ψ is true, only if P_1 can make a cut, such that for every cut P_2 makes, there exists a cut for P_3 such that, ..., the utility of P_1 is 1. This establishes our reduction. □

We remark that it is not obvious whether or not the corresponding decision problem is contained within Σ_k^{P}. It is not hard to see we can write a Boolean formula equivalent to the smart contract game in a similar manner as with a single contract. The problem is that it is unclear if the innermost predicate ϕ

can be computed in polynomial time. It is not hard to see that some smart contracts do not have a polynomial description, i.e. we can encode a string $x \in \{0,1\}^*$ of exponential length in the contract. However, there might be an equivalent contract that *does* have a polynomial-time description. By equivalent, we mean one that has the same dominating path. This means that whether or not Σ_k^{P} is also an upper bound essentially boils down to whether or not every contract has an equivalent contract with a polynomial description.

6.4 Perfect Information, Two Contracts, Upper Bound

In this section, we consider two-agent games of perfect information and provide a polynomial-time algorithm for computing an SPE in these games. Specifically, for a game tree of size m with ℓ terminal nodes with two contract agents (and an arbitrary number of non-contract agents), we can compute the equilibrium in time $O(m\ell)$. Our approach is similar to that of [179], in that we compute the inducible region for the first agent, defined as the set of leaves they are able to 'induce' by making cuts in the game tree.

Let A, B be two sets. We then define the set of outcomes from A reachable using a threat against agent i from outcomes in B as follows:

$$\text{threaten}_i(A, B) = \{x \in A \mid \exists\, y \in B.\, x_i > y_i\}$$

As mentioned, we will compute the *inducible region* for the agent with the first contract, defined as the set of outcomes reachable with a contract. Choosing the optimal contract is then reduced to a supremum over this region.

Definition 6.8 (Inducible Region)**.** *Let G be a fixed game. We denote by $\mathcal{R}(P_1)$ (resp. $\mathcal{R}(P_1, P_2)$) the* inducible region of P_1*, defined as the set of outcomes reachable by making a cut in G in all nodes owned by P_1. $\mathcal{R}(P_1)$ is a tuple $(\mathbf{u}, c_1)$ where $\mathbf{u} \in \mathbb{R}^n$ is the utility vector, and c_1 is the contract (a cut) of P_1.*

Now let G be the game tree in question and let k be a fixed integer. As mentioned, we assume without loss of generality that G is in *generic form*, meaning all non-leaves in G have an out-degree of exactly two and that all utilities for a given agent are distinct such that the ordering of utilities is unique. We denote by P_1, P_2 the agents with contracts and assume that P_i has the i^{th} contract. We will compute the inducible regions in G for P_1 (denoted S for *self*), and for (P_1, P_2) (denoted T for *together*) by a single recursive pass of the tree. In the base case with a single leaf with the label $\mathbf{u}$ we have $S = T = \{\mathbf{u}\}$. For a non-leaf, we can recurse into left and right child, and join together the results. The procedure is detailed in Algorithm 2.

Theorem 6.9. *An SPE in two-contract games of perfect information can be computed in time $O(m\ell)$.*

Data: Extensive-form game G.
Result: Inducible region S for agent 1 (*self*); inducible region T for both agents 1 and 2 (*together*).
function InducibleRegion(G)**:**
  switch G **:**
    case `Leaf`($\mathbf{u}$) **:**
      return $(\{\mathbf{u}\}, \{\mathbf{u}\})$

    case `Branch`(i, G^L, G^R) **:**
      $(S^L, T^L) \leftarrow$ InducibleRegion(G^L)
      $(S^R, T^R) \leftarrow$ InducibleRegion(G^R)
      if $i = 1$ **then**
        $T \leftarrow T^L \cup T^R$
        $S \leftarrow S^L \cup S^R \cup \text{threaten}_2(T^L \cup T^R, S^L \cup S^R)$
      else if $i = 2$ **then**
        $T \leftarrow T^L \cup T^R$
        $S \leftarrow \text{threaten}_2(S^L, S^R) \cup \text{threaten}_2(S^R, S^L)$
      else
        $T \leftarrow \text{threaten}_i(T^L, T^R) \cup \text{threaten}_i(T^R, T^L)$
        $S' \leftarrow \text{threaten}_i(S^L, S^R) \cup \text{threaten}_i(S^R, S^L)$
        $S \leftarrow S' \cup \text{threaten}_2(T, S')$
      return (S, T)

Algorithm 2: Pseudo-code of the algorithm for computing reverse Stackelberg equilibria in games of perfect information. For simplicity, we are assuming that the game tree is bifurcating and is in generic form.

Proof. First, the runtime is clearly $O(m\ell)$ since the recursion has $O(m)$ steps where we need to maintain two sets of size at most ℓ. For correctness, we show something stronger: let $\mathcal{R}(P_1)$ be the inducible region for P_1 in the expanded tree and $\mathcal{R}(P_1, P_2)$ be the inducible region of (P_1, P_2). Now, let $(S, T) =$ `InducibleRegion`(G). Then we show that $S = \mathcal{R}(P_1)$ and $T = \mathcal{R}(P_1, P_2)$. This implies that $\text{argmax}_{u \in S} u_1$ is the SPE. The proof is by induction on the height h of the tree. As mentioned, we assume that games are in *generic form*. This base case is trivial so we consider only the inductive step.

Necessity follows using simple constructive arguments: for S and $i = 1$, then for every $(\mathbf{u}, c) \in S^\ell$, we can form a contract where P_1 chooses the left branch and plays c. And symmetrically for S^R. Similarly, for every $(\mathbf{u}, c_1, c_2) \in T^L$ and $(\mathbf{v}, c') \in S^L$ can form a contract where P_1 plays c_1 in all subgames where P_2 plays c_2; and plays c' otherwise. Then $\mathbf{u}$ is dominating if and only if $\mathbf{u}_2 > \mathbf{v}_2$. Similar arguments hold for the remaining cases.

For sufficiency, we only show the case of $i = 1$ as the other cases are similar. Assume (for contradiction) that there exists $(\mathbf{u}, c_1) \in \mathcal{R}(P_1) \setminus S$, i.e. there is a

P_1-cut c_1 such that $\mathbf{u}$ is dominating. Then,

$$\begin{aligned}(\mathbf{u}, c_1) &\in (T^L \cup T^R) \setminus (S^L \cup S^R \cup \text{threaten}_2(T^L \cup T^R, S^L \cup S^R)) \\ &= \{\mathbf{v} \in (T^L \cup T^R) \setminus (S^L \cup S^R) \mid \forall \mathbf{v}' \in S^L \cup S^R . \mathbf{v}_2 < \mathbf{v}'_2\}\end{aligned}$$

That is, $\mathbf{u}$ must be a utility vector that P_1 and P_2 can only reach in cooperation in a one of the two sub-games, say by P_2 playing c_2. However, for every cut that P_1 makes, the dominating path has utility for P_2 that is $> \mathbf{u}_2$, meaning P_2 strictly benefits by not playing c_2. But this is a contradiction since we assumed $\mathbf{u}$ was dominating. □

6.5 Perfect Information, Unbounded Contracts, PSPACE-hardness

We now show that computing an SPE remains PSPACE-complete when considering games with an arbitrary number of contract agents. We start by showing NP-hardness and generalize to PSPACE-hardness in a similar manner as we did for Theorem 6.7. The reduction is from 3-COLORING: let (V, E) be an instance of 3-COLORING and assume the colors are $\{R, G, B\}$. The intuition behind the NP-reduction is to designate a coloring agent P_{color}, who picks colors for each vertex $u \in V$ by restricting his decision space in a corresponding move using a contract. They are the first agent with a contract. This is constructed using a small stump for every edge $e \in E$ with three leaves R_u, G_u, B_u. We also have another agent P_{check} whose purpose is to ensure no two adjacent nodes are colored the same. We attach all stumps to a node owned by P_{check} such that P_{check} can choose among the colors chosen by P_{color}. If P_{color} is able to assign colors such that no adjacent nodes share a color, then P_{color} maximizes their utility, however, if no such coloring exists then P_{check} can force a bad outcome for P_{color}. It follows that P_{color} can obtain good utility if and only if there is a valid coloring.

Formally, we add six contract agents for every edge in the graph. Specifically, for every edge $(u, v) \in E$ and every color $c \in \{R, G, B\}$, we introduce two new contract agents $P_{u,c}$ and $P_{v,c}$ who prefer any outcome except c_u (resp. c_v) being colored c. That is, if $c = R$, then the leaf R_u has a poor utility for $P_{u,R}$. We add moves for $P_{u,c}$ and $P_{v,c}$ at the top of the tree, such that if they cooperate, they can get a special utility vector $\perp_{u,v}$ which has a poor utility for P_{color} and great utility for P_{check}, though they themselves prefer any outcome in the tree (except c_u, resp. c_v) to $\perp_{u,v}$. We ensure that P_{check} has a contract directly below P_{color} in the tree. If no coloring exists, then P_{check} can force a bad outcome for both $P_{u,c}, P_{v,c}$ in all contracts where they do not commit to choosing $\perp_{u,v}$. Specifically, P_{check} first threatens $P_{u,c}$ with the outcome c_u, and subsequently threatens $P_{v,c}$ with c_v. Though they prefer any other node in the tree to $\perp_{u,v}$, they still prefer $\perp_{u,v}$ to c_u, c_v, meaning they will comply

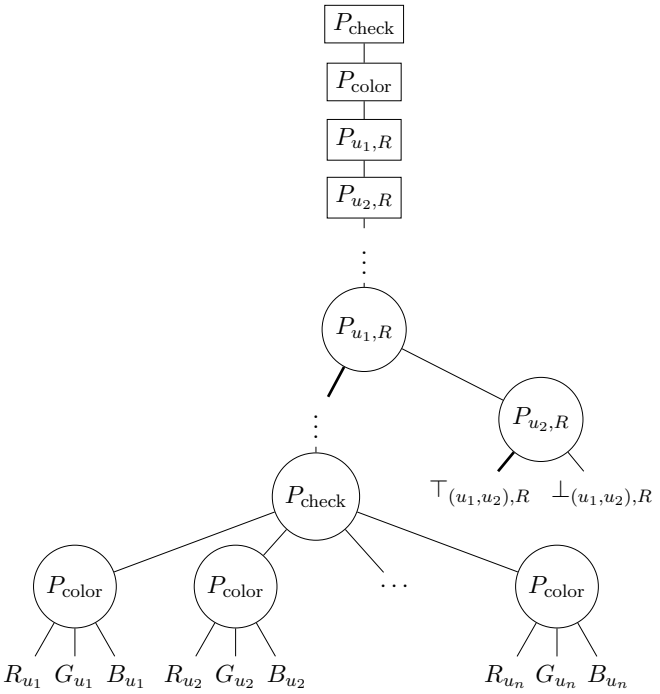

Figure 6.4: The structure of the reduction. First, P_{color} is allowed to assign a coloring of all vertices. If there is no 3-COLORING of the graph, there must be some vertex (u_1, u_2) where both vertices are colored the same color c. In this case, P_{check} can force both c_{u_1}, c_{u_2}, which are undesirable to $P_{u_1,c}$, resp. $P_{u_2,c}$: then in every $P_{u_1,c}$-contract where they do not commit to choosing $P_{u_2,c}$, P_{check} cuts as to ensure c_{u_1} and analogously for P_2. It follows that P_{check} can get $\bot$ if and only if the graph is not 3-colored. Then P_{color} can get a different outcome from $\bot$ if and only if they can 3-color the graph.

with the threat. This means P_{color} will receive a poor outcome if the coloring is inconsistent. It follows that P_{color} will only receive a good payoff if they are able to 3-color the graph, see e.g Section 6.5 for an illustration.

Theorem 6.10. *Computing an SPE in smart contract games of perfect information is PSPACE-hard when we allow for an unbounded number of contract agents.*

Proof. Let (V, E) be an instance of 3-COLORING. Our above reduction works immediately for $k = 1$, showing NP-hardness. To show PSPACE-hardness we reduce from a variant of 3-COLORING as described in [39] where agents alternately color an edge and use a similar trick as Theorem 6.7 by introducing new agents between P_{color} and P_{check}. □

It remains unclear where the exact cutoff point is, though we conjecture it to be for three contracts: clearly, the decision problem for three-contract games

of perfect information is contained in NP as the witness (a cut for the first contract agent) can be verified by Algorithm 2.

Conjecture 6.11. *Computing an SPE for three-contract games is NP-complete.*

Compliance with Threats. One problem with our model is that we assume agents are always able to comply with the threat, which might be realistic in practice (say due to incompetence). Instead, we can refine the notion of equilibrium to trembling hand equilibria where the follower might not be able to respond to the threat, say with probability ε [225]. This has implications for modeling e.g. smart contract ransomware.

A Hierarchy of Contracts? Designated-verifier non-interactive zero-knowledge proofs [73, 151, 207] can be used by agents to prove properties of their contracts in zero-knowledge to selected subsets of other agents, which essentially corresponds to adding information sets to the contracts, such that in addition to choosing a contract, each agent also chooses a subset of all possible contracts containing their contract. This is further complicated by letting an agent prove to other agents what they proved to others. This induces a hierarchy of equilibria (the 'NIZK hierarchy') where agents relay information they were proven by other agents. It is unknown if all layers of this hierarchy are distinct.

Chapter 7

Threats

> "*I'm afraid I don't understand something, Alexei. Is the Premier threatening to explode this if our planes carry out their attack?*"
>
> - President Merkin Muffley

> "*No sir. It is not a thing a sane man would do. The doomsday machine is designed to trigger itself automatically.*"
>
> - Alexei de Sadeski

We now study the role of smart contract capability in changing the equilibria of games. This naturally leads us to characterize those games that are unaffected by the addition of smart contracts. We say such games are *Stackelberg resilient*. In this chapter, we show various properties of Stackelberg resilience: we analyse a variety of contracts, and find that only some of them are resilient.

Attribution. This chapter is based on the papers [164, 165]. Most of the chapter is taken verbatim from [164] (including all the figures), with only minor modifications to the formatting and the prose. Section 7.4 and all of its figures is taken verbatim from [165].

7.1 Stackelberg Resilience

Let us now formally what it means for a game to be resilient. We first need to define algebraically the notion of smart contracts introduced in Chapter 6.

Definition 7.1 (Contract Moves)**.** *Let G be an extensive-form game on n agents. We define $\mathcal{C}_i(G)$ as the game that starts with a smart contract move for agent i whose only subgame is G. Similarly, if $P : [m] \to [n]$ is a list of agents of length m that specifies the order of the contracts, we denote by $\mathcal{C}_P(G)$ the game with m contracts belonging to the agents specified by the list.*

Generally speaking, being the first agent to deploy a smart contract is an advantage. As a result, the equilibrium of the game may be sensitive to the order of the agents in a given list. We call the agent $P(1)$ with the first contract the *leading contract agent*. In practice, the order of the agents is determined by the consensus protocol used by the underlying blockchain and may be non-deterministic. For the purposes of this paper, we assume the consensus protocol is agnostic to the agents such that the order is random. Thus, we say a game is Stackelberg resilient if only if the equilibrium remains the same for any order of the contracts. An example of a game that is not Stackelberg resilient is shown in Fig. 7.1. We will now make this notion more formal.

Definition 7.2 (Equivalent Games)**.** *Let G, G' be two games on n agents. We say that G and G' are* equivalent*, written $G \cong G'$, if for every equilibrium s^* in G (respectively, in G') there exists an equilibrium $s^{*\prime}$ in G' (respectively, in G) such that $u_i(s^*) = u_i(s^{*\prime})$ for every $i \in [n]$.*

Note that this is an equivalence relation. For two equivalent games, for each equilibrium in either game, there is an equilibrium in the other game with the same payoffs. This implicitly means that we regard any two outcomes with the same utility vector as equivalent. While this is not necessarily the case in general, it will be the case for the types of games we consider. Namely, in our case we would have G as an extensive-form game in generic form, meaning that all its utility vectors are distinct, and $G' = \mathcal{C}_P(G)$ the same game with contracts in the order specified by P. In the game G', we will have multiple copies of each utility vector from G, however all its appearances represent the 'same' underlying leaf from the game G.

Definition 7.3 (Stackelberg Resilience)**.** *A game G is said to be* Stackelberg k-resilient *for an integer $k > 0$ if, for any list P of k distinct agents, it holds that $\mathcal{C}_P(G) \cong G$. We say G is* (full) Stackelberg resilient *if it is Stackelberg n-resilient.*

If the SPE of a game is unique and labeled with some utility vector $u \in \mathbb{R}^n$, then side-contract resilience says that every SPE in the expanded tree also has to be labeled with u.

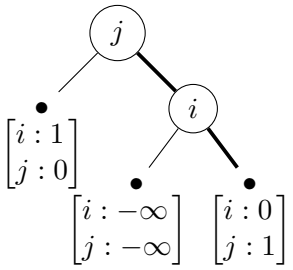

Figure 7.1: An example of a game that is not 1-resilient. agent j has a coin that they can choose to give to agent i. agent i is subsequently given the option to trigger a doomsday machine. Without contracts, the SPE is the node $(0, 1)$ where agent j keeps the coin because the doomsday machine is an empty threat. However, when agent i has a contract, they can commit to detonating the doomsday machine if they do not receive the coin, thus changing the SPE to $(1, 0)$. Such an equilibrium is called a Stackelberg equilibrium.

Note that in this definition, we require the list P to consist of distinct agents (i.e. P is injective). If this were not the case, $(k + 1)$-resilience would trivially imply k-resilience for the uninteresting reason that adding contracts for a fixed agent is an idempotent operation, i.e. the same agent having two nested contracts is equivalent to them having only the topmost contract. Also note that if a game is not Stackelberg resilient, there exists a Stackelberg attack that some agent can deploy to obtain better utility (or force worse utility for others) as compared to the situation without contracts. Only Stackelberg resilient games are not susceptible to Stackelberg attacks.

There are a few observations that we can see immediately. First, if every agent has the same most preferred outcome and this outcome is the SPE of the original game, there cannot be a viable attack and the game is trivially resilient. We also observe that if an agent has the last contract and their only node is the root of the original game tree, then the choice of contract and the choice of move 'collapse' and they cannot affect change through commitment to a contract. We observe that in general, it is hard to reason about Stackelberg resilience.

Proposition 7.4. *If $\mathsf{P} \neq \mathsf{NP}$, there is no efficient algorithm to compute Stackelberg 1-resilience for games with more than two agents. However, full Stackelberg resilience can be computed efficiently for two-agent games of perfect information.*

Proof (sketch). The first part follows using the same reduction as in the proof of Theorem 6.5, noting that since agent 1 always moves last and thus have to play the SPE, the equilibrium of the game does not change by giving either agent 2 or agent 3 a contract — they obtain their preferred outcome among the set of feasible outcomes (given that agent 1 always chooses $\top$). Thus, the game is not 1-resilient if and only if the circuit is satisfiable, which shows

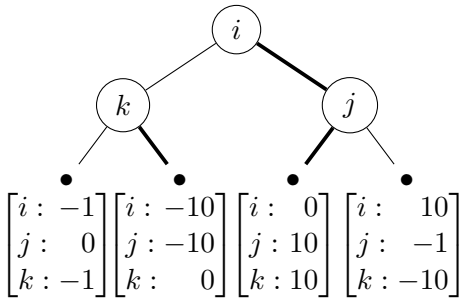

Figure 7.2: An example of a game that is 1-resilient, but neither 2-resilient nor 3-resilient. We denote by u^ℓ the ℓ^{th} utility vector from left-to-right, for $\ell = 1 \dots 4$.

that computing 1-resilience is NP-hard for three-agent games. For the latter part, note that by Theorem 6.9, full Stackelberg resilience can be computed efficiently for games of two agents by invoking InducibleRegion twice, relabeling the agents in the second invocation. □

7.2 Downward Transitivity

It is a natural question to wonder if, given resilience in the case with k contracts, we have resilience in $(k-1)$ case, and thus, inductively, for all subsequent removals of contracts. The contrapositive of this question is also interesting in its own right: can resilience be restored by adding a contract? What if this added contract is in the least favorable, last position? Indeed, we need to address the contrapositive to answer the original question.

Consider a game G^0 that is not $(k-1)$-resilient. Then some agent has a Stackelberg attack against the others that results in a better equilibrium for the attacker. These attacks work by allowing the attacker to commit to a worse outcome if the others do not comply. Thus, the attacking agent can coerce the other agents into obeying some threat. We must now ask if there can be some sort of *threat to the threat*. As we will illustrate in the following example, there can indeed be a threat to the threat if the new contract agent gets to go first. On the other hand, if the new contract agent has the last contract, they cannot threaten the original threat. To intuitively see why going last nullifies any potential threat to the threat, recall that the last contract move is effectively the last move. Thus, if this final k^{th} agent could make a threat, it cannot commit until it is too late, in some sense 'keeping the threat a secret' until all other agents have already made their moves. This means that if the k^{th} agent makes a threat, they already know whether they will have to play it.

To illustrate this, we introduce the example illustrated in Fig. 7.2. We label the utility vectors $u^1, \dots, u^4$ from left to right for ease of notation. Note

the following inequalities, which will be useful going forward.

$$u_i^4 > u_i^3 > u_i^1 > u_i^2,$$
$$u_j^3 > u_j^1 > u_j^4 > u_j^2,$$
$$u_k^3 > u_k^2 > u_k^1 > u_k^4.$$

The SPE in this game is easy to read off: we can see that if k gets a choice, they will choose to play right, resulting in u^2 and, if j gets a choice, they will choose left, resulting in u^3. Seeing this, agent i will choose right, yielding u^3 as the SPE. The addition of just one contract cannot change the equilibrium. Neither agent j nor k could do better than the SPE and, since i owns the root, committing to a move and going first are effectively the same. A second contract can break resilience. If i has the first contract and j the second, we can have the following contract threat from i:

i: "*if agent j does not* commit to *playing right, I will play left.*"

This contract can easily be converted into a cut for agent i: cut away the right branch in every subgame where agent j did not cut away their left branch. Moving forward, we will not formally convert the contracts into cuts and trust that it is clear from the context what is intended.

If j does not comply, then i plays left and k will play right, resulting in u^2. If j does comply, the outcome will be u^4, which is a better option for j than u^2. Thus, j must comply. We label this game, with the specific contract order of i then j, G^1. So far, agent k has had no impact on i's antics as both other agents know that, should the game come to k's node, k has no choice but to play left for u^2. In fact, i's threat is predicated on this.

Both j and k are worse off in G^1 as compared to G^0. If k gets the first contract, before those of i and j, the G^1 threat can be nullified. agent k commits to a *threat to the threat* wherein their contract commits them to play right if i plays the contract from G^1. This means that i's 'threat' results now u^1, not u^2, and j would actually prefer u^1 to u^4. Thus if i deployed the contract from G^1, j would not be threatened into committing the contract stipulated by i. Now i can infer that if they try the G^1 contract as second contract agent after k, the equilibrium will be u^1, which is worse for i than the old SPE u^3, so it would be better to not try the threat and we end up on u^3 again.

However, suppose we allow k only the last contract, yielding the order i, j, k. This order means that k is still last to move. Intuitively, the reason that the *threat to the threat* will not work is that k moves last and cannot commit to an action that they know will leave them worse off. To make this precise, we first expand the game tree for first k's then j's possible contracts in Fig. 7.3. As pictured, j first makes a contract move, which can predicate on k's contract, but cannot see their move. Next, k can make a contract, with full knowledge

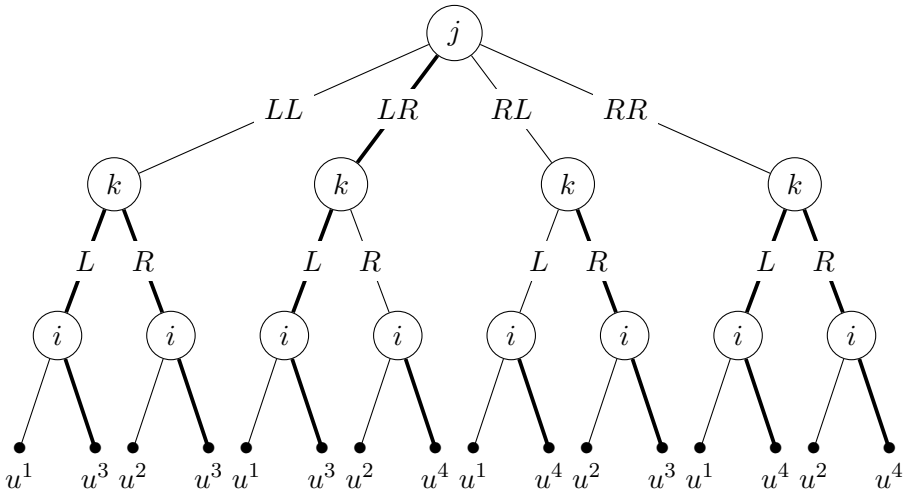

Figure 7.3: The game Fig. 7.2 expanded with a contract for j and then for k. All subgames consisting only of moves for j and k have been collapsed to the SPE. agent j can commit to going L or R depending on whether or not k commits to L or R. Now suppose we add a contract for agent i at the beginning. By making appropriate cuts in this tree, agent i can commit to actions that force j to commit to contract RR. The expanded game with also a contract for agent i has roughly four orders of magnitude more nodes and has been omitted for brevity.

of j's contract. agent j's contracts are labeled with their commitments based on whether k commits to left or right, respectively. That is, in contract LL, agent j commits to playing left regardless of k's commitment and, in LR, j commits to left if k does and right if k commits right. For the simplicity of this illustration, we assume all the contracts fully commit the agents to actions in all cases. Notice that it is always i's local SPE to move right, which would result in u^3, the old SPE and the other two's favorite, in j's right three branches. This is not so in RR, but it is not clear that j would ever commit to that contract, given that the others all appear to offer a better opportunity.

Rather than expand the tree for i, which results in a large graph, we instead explore the cuts that i can make. First, note that k's least favorite outcome, u^4, is i's best. This means that no combination of i's cuts can be used to persuade k to move toward u^4. For example, if j makes the LR contract, i could cut away u^3 in favor of u^1 unless k moves right, but the choice for k, between u^1 and u^4 would still see k opting for u^1. Instead, i's best contract does not take into account k's contract at all and is as follows:

i: "*Unless j commits to RR, I will cut away the right branch.*"

This means that when j does commit to RR, i.e. when j commits to i's desired u^4 regardless of what k commits to, i can be guaranteed u^4 if they

either commit to or play left. For every j contract that is not RR, i commits to playing left, towards k's node. With this set up, we now go through the backwards induction of the other agents, given the knowledge of i's contract. Starting with the final agent, k, who has the following position:

> k: "*Unless j commits to RR, agent i will play toward my node, giving me a choice between u^1 and u^2. Among these I prefer u^2. Regardless of whether or not I deploy a contract, if I go, I will go last, so whichever outcome I commit or play for I will certainly receive. So, unless agent j commits to RR I will opt for u^2.*"

agent j can then make the following inference:

> j: "*Unless I commit to RR, agent i will play toward agent k, leaving agent k with only the choice between u^1 and u^2 and they will certainly pick u^2. If I do commit to RR, then agent i will play towards my node and I will be obligated to play u^4, regardless of what agent k might commit to. Thus I have a choice between u^2 and u^4 and will pick u^4 and commit to obeying the threat.*"

Thus j will commit to the RR contract and the game will result in the same equilibrium as G^1. The interesting observation about this situation is that i's threat does not depend on k making a specific commitment. Instead, i is counting on the fact that if k were to make a *threat to the threat*, as in the case when k has the first contract, the commitment to that threat comes after all other agents have made commitments. With this timing, k knows they certainly will have to follow through with their threat if they make it. Thus j knows that the help k offered when they had the first contract is no longer viable and the old threat from G^1 still stands.

Definition 7.5 (Downward Transitivity)**.** *Stackelberg resilience has the property of* downward transitivity *if k-resilience implies ℓ-resilience for all $\ell \leq k$.*

The property of *downward transitivity* means that if a game is resilient for some number of contracts, that resilience will still hold with fewer contracts. We will now show that this property holds. As a warm-up, consider the case of 1-resilience from 2-resilience: here, we claim that omitting the second contract from a 2-resilient game still yields the same equilibrium. If the game in which only agent 1 has a contract had a different outcome, there must be at least one node in G for which the corresponding set I is different. Let G^* be the lowest such node and observe that by definition this cannot be a leaf. Suppose G^* were owned by 2; given that 2 has no contract, 2 will pick the local SPE child from $I^L \cup I^R$. It is easy to see that $\mathsf{threaten}(I^L, I^R) \subseteq I^L \subseteq L^L$, where

L^L denotes the leaves of the left subtree, and analogously so on the right, $\mathsf{threaten}(I^R, I^L) \subseteq I^R \subseteq L^R$. Since this game is 2-resilient, the optimal choice for 2 will correspond to both the universal SPE and the 2-contract choice and therefore also the local SPE. If 2 played their local SPE in the 2-contract case, any contract they had with regards to that particular node was trivial so the removal thereof will not affect the potential threats. Given that the I^L and I^R are the same, but G^* is the lowest deviant node, we have a contradiction. If instead, G^* is owned by 1, $I^L \cup I^R$ will be the same as in the 2-contract case since I^L and I^R are the same. Furthermore, the threaten mechanism is unchanged and 1 may use any inducible leaf to threaten for any leaf below that node, just as before. Since I^L and I^R are the same by assumption, there are no new threats to make. Thus I is unchanged and we again have a contradiction. Thus removing the second contract in a 2-resilient game will not change the equilibrium. Applying this to both orders of contract arrangement gives the desired result.

Theorem 7.6. *Stackelberg resilience is downward transitive.*

Proof. We show the contrapositive: assume there is game G^0 which is not $(k-1)$-resilient, then we claim it is also not k-resilient. We can assume w.log. that G^0 is in generic form. Let $u^{SPE,0}$ be the utility vector of the subgame perfect equilibrium. We know that G^0 is not $(k-1)$-resilient, so there is a list P of $(k-1)$ agents such that giving these agents contracts in the order specified by P results in a utility vector $u^{SPE,1} \neq u^{SPE,0}$. Let G^1 be the game that starts with these contract moves and ends with G^0 (see Fig. 7.4). Let $I \subseteq P$ be the agents for whom $u_i^{SPE,1} > u_i^{SPE,0}$, and let $J = P \setminus I$ be its complement. We know that both $I, J \neq \emptyset$, because if I is empty then we would not have $u^{SPE,1} \neq u^{SPE,0}$ and similarly for J.

Now let $k \in [n] \setminus P$ be arbitrary and define the game G^2 that starts with contract moves in the order specified by L, then has a contract move for agent k, and finally a subgame with G^0 (see Fig. 7.4 (b)). Let $u^{SPE,2}$ be the utility vector of the SPE. We can assume w.log. that the only agents in G^0 are $I \cup J \cup \{k\}$ (as we can collapse the subgames otherwise). Our claim is that $u^{SPE,2} \neq u^{SPE,0}$. If $u^{SPE,2} = u^{SPE,1}$ then we are done, so assume $u^{SPE,2} \neq u^{SPE,1}$. Now consider a subgame where the root node is owned by k. Define the *local SPE* as the subgame perfect equilibrium for this subgame and call the arrived at utility vector u^{LSPE}. In order for $u^{SPE,2} \neq u^{SPE,1}$, there must exist a subgame, the *key tree*, and its corresponding local SPE, such that agent k commits to an action that results in a different utility vector (for that subgame), say u^T. By subgame perfection, we must have $u_k^T < u_k^{LSPE}$.

Since the equilibrium changed from G^0 to G^1 there must be some subset of agents, say $P \subseteq J$, who committed in G^1 to actions that they would not have played in G^0, and that these changes resulted in $u^{SPE,1}$. These agents have to have been threatened, as they are now worse off. We know there is a threat

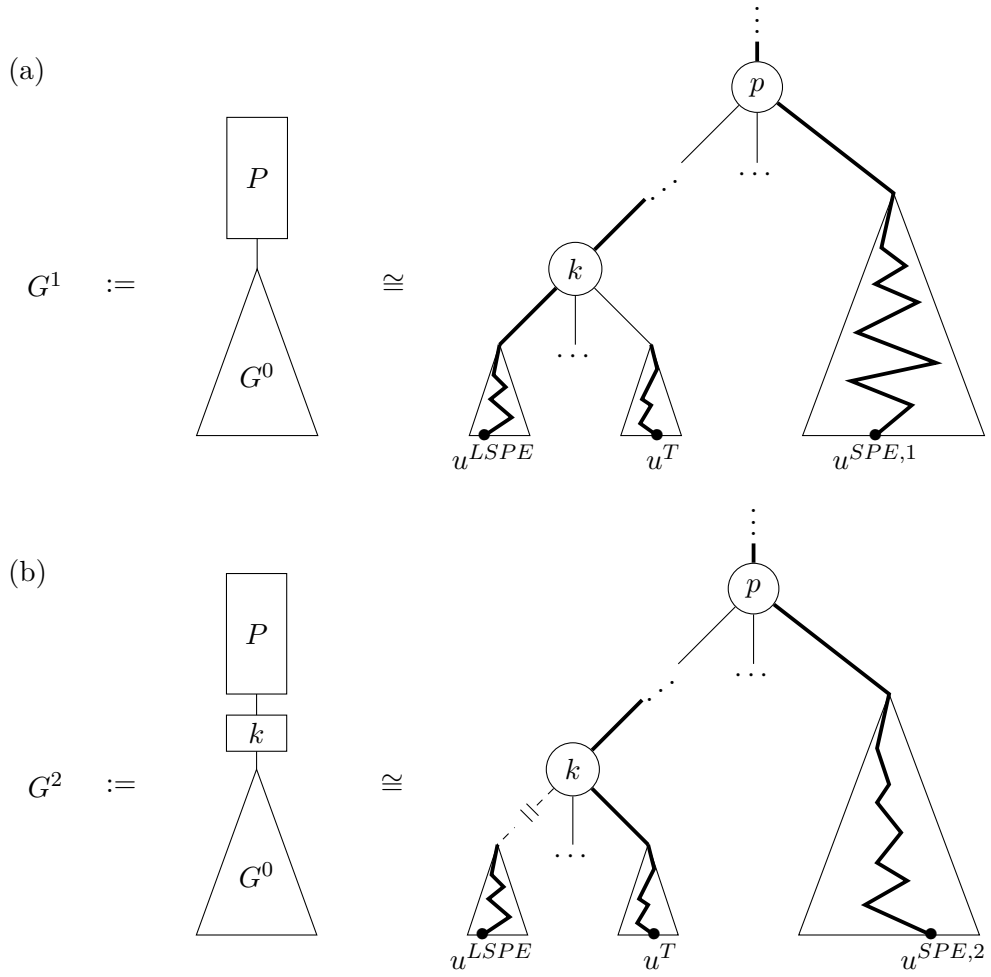

Figure 7.4: (a) The game G^1 as an extensive-form game. The agent $p \in J$ at some point chooses between the subgame that leads to $u^{SPE,1}$ or the subgame that leads to the key tree, the subgame in which agent k chooses between u^{LSPE} and u^T. (b) The game G^2 as an extensive-form game. The agent $p \in J$ at some point chooses between the subgame that leads to $u^{SPE,2}$ or the subgame that leads to the subgame where agent k chooses between u^{LSPE} and u^T. In order for the SPE to have changed from $u^{SPE,1}$ to $u^{SPE,2}$, agent k must have cut away the subgame that corresponds to u^{LSPE}.

from I against J in G^1. If this threat were not u^{LSPE} then the equilibrium would not change from G^1 to G^2 by the introduction of the contract for k that moves from u^{LSPE} to u^T in that subgame. Let $p \in P$ be any agent who is being threatened using u^{LSPE}. Assume for simplicity there is only one such p. In order for u^T to nullify the threat of u^{LSPE}, we have to have $u_p^T > u_p^{LSPE}$. Thus, the only disruption that k can affect is to commit to u^T. Now, consider the following contract for I:

> I: "*I will cut as to necessarily reach the subgame owned by p in every subgame where p does not commit to* $u^{SPE,1}$*; otherwise I will make the same cut as I did in the contract that resulted in* $u^{SPE,1}$*.*"

Consider the choice faced by k. Since they are the last agent with a contract they can see the actions stipulated in the contracts of all the other agents. As argued, agent k can either commit to playing towards u^{LSPE} or they can commit to playing towards u^T. If p commits to $u^{SPE,1}$ then we are done. Then we know, based on I's contract, that p will be faced with the choice of the key tree or $u^{SPE,1}$. Thus if k commits towards u^T, they can infer that p will choose the key tree since $u_p^T > u_p^{SPE,1}$. Thus, k sees that a commitment towards u^T will result in that outcome. Now consider the choice faced by p. If they comply with I and commit to $u^{SPE,1}$ they will get it. If they do not comply with I and play for the key tree, they know that agent k – seeing this contract – will not commit to u^T (as they would prefer u^{LSPE} when given this choice). This gives p the choice between u^{LSPE} and $u^{SPE,1}$, for which the latter outcome is already known to be their best response. Thus, we have demonstrated a contract for I in which they can prevent going back to the old equilibrium. In particular, this means that G^0 is not k-resilient, as desired. □

This can also be viewed as a monotonicity property of the contracts: namely, that any outcome that is inducible with the use of contracts for some agent, is still inducible when adding a contract at the end for another agent. We note that it is not necessary that the G^2 equilibrium be the same as the G^1. The addition of a contract for k could easily open up even better opportunities for the agents that benefited from the G^1 contracts — we simply cannot go back to the G^0 SPE.

7.3 Decentralized Commerce

In this section, we demonstrate non-triviality of Stackelberg resilience by analyzing Contracts 3.1 and 3.3. Both contracts solve a problem in decentralized commerce where two agents want to exchange a good using a blockchain. As before, the contracts involve a seller S and a buyer B that want to exchange an item *it* for a price of x. We let y denote the value of *it* to the buyer and x'

the value to the seller, and assume $y > x > x' > 0$. Both contracts are shown to securely implement honest decentralized commerce with ε-strong game-theoretic security, for arbitrarily large ε. Note that both contracts are similar and implement almost the same functionality. However, it turns out that only one of them is Stackelberg resilient which demonstrates that Stackelberg resilience is a non-trivial property.

Theorem 7.7. *Contract 3.1 is not Stackelberg resilient.*

Proof. Label the utility vectors at the leaves of contract, seen as the tree in Fig. 1(a), u^j for $j \in [4]$ from left to right, where u^1 is the leftmost leaf, corresponding to strategy send-dispute, and so on across the base of the tree. Let u_i^j for $i \in \{B, S\}$ be the utility of the buyer and seller at leaf j.

Consider the case where the first agent is S. We apply Algorithm 1 to the tree and keep track of set I for each node as we move up the tree. At the leaf level, each leaf is its own set I. The next level up depends on the left and right child, G^L and G^R, respectively, of the root. We denote I^L and I^R to be the associated sets. Both nodes are owned by the buyer, which is agent 2. It is easy to see that $u_B^2 > u_B^1$, which yields $I^L = \{u^2\}$, and $u_B^4 > u_B^3$, which yields $I^R = \{u^4\}$, as the inducible regions. Moving up the tree, we arrive at the root, which is owned by the seller. Here the seller can use any of the $I^L \cup I^R$ utilities to threaten the buyer into any of the L leaves. Thus the seller can use $u^4 \in I^R$ to threaten the buyer into u^1 as long as $y - x - \lambda > 0$, which is allowed for by the stipulations on parameters in Contract 3.1. So $I = \{u^1, u^2, u^4\}$. The seller has equal utility in u^1 and u^2 and, since we assume the agents are weakly malicious and $u_B^2 > u_B^1$, the seller will choose u^1, which is not the SPE. Thus the contract is not Stackelberg 2-resilient. If instead B has the first contract, the SPE and reverse Stackelberg equilibria coincide. Since the buyer owns the middle level of nodes, we have $I^L = \{u^1, u^2\}$ and $I^R = \{u^3, u^4\}$. At the root, the buyer can threaten the seller into u^1 and u^2 with u^4 and into u^3 with either element of S^L because the seller is weakly malicious. From $S = \{u^1, u^2, u^3\}$, the buyer will pick u^2, which is the SPE. □

Note the S Stackelberg attack would still be viable if we accounted for the loss of the sale item in u_S^1 and u_S^2, as is the case in the game from Contract 3.3. In the full version of the contract in Contract 3.1, which allows agents to additionally play a garbage string, it can be readily observed that S can threaten with garbage to get the u^1 equivalent regardless of the value of the deposit λ.

Theorem 7.8. *Contract 3.3 is full Stackelberg resilient.*

Proof. In keeping with Theorem 7.7, we label the leaves u^j, $j \in [6]$ from left to right, with reference to Fig. 1(b). Let I^{LL} be the set associated with left child of the left child of the root and I^{RR} symmetrically so on the right.

Suppose first that S is agent 1. Since both nodes at the third level, G^{LL} and G^{RR} belong to the seller, we have $I^{LL} = \{u_1, u_2\}$ and $I^{RR} = \{u_5, u_6\}$. The next level is comprised of nodes owned by the buyer. On the LHS, the seller can use u^2 to threaten the buyer into u^3 and u^3 to threaten for u^1 because $u_B^1 > u_B^3 > u_B^2$. We have $u_B^3 > u_B^2$ if $y - x > y\gamma - (x + \lambda)(1 - \gamma)$. Since $\gamma < 1/2$, the desired result is true if $y - x > y - (x + \lambda)(1 - \gamma)$. Simplifying and rearranging yields $\frac{\gamma}{1-\gamma}x < \lambda$, which is a stipulation of Chapter 3. So we have $I^L = \{u^1, u^3\}$. On the RHS, we can readily see $u_B^6 > u_B^4$. We also have $u_B^5 > u_B^4$ if $-x < -(x + \lambda)\gamma$. Solving this expression for λ, we have $\frac{1-\gamma}{\gamma}x > \lambda$, which is again required by the contract. So B will always want to move left, yielding $I^R = \{u^5, u^6\}$. At the root, S can threaten using any strategy in $I^L \cup I^R$. It is easy to see that the element with the lowest utility for B is u^5. We have already shown that $u_B^5 > u_B^4$. Thus $I = \{u^1, u^3, u^5, u^6\} \cup \{u^2 \mid u_B^5 < u_B^2, \}$, that is, u^2 is in I if $u_B^5 < u_B^2$. While it is true that $u_B^5 > u_B^2$, it is easier to see that even if u^2 were inducible, it would be a less optimal result for S compared to u^3, a fact easily seen with the observation that $\gamma < 1/2$. Of the remaining choices, it immediate that u^1 and u^6 will not be optimal strategies for S. All that is left to show is $u_S^3 > u_S^5$. The desired result is true if $x - x' > x\gamma - \lambda(1 - \gamma)$, which can be rearranged to $x > \frac{1}{1-\gamma}x' - \lambda$. Since we have the assumption that $x > x'$, the previous statement is true if $x > \frac{1}{1-\gamma}x - \lambda$, given that the coefficient $\frac{1}{1-\gamma}$ must be positive. Solving for λ yields $\lambda > \frac{\gamma}{1-\gamma}x$, which is a requirement of Chapter 3. Thus, S will pick u^5, which is the SPE found in Chapter 3.

Suppose now that B has the first contract. S owns G^{LL} and G^{RR} and thus $I^{LL} = \{u^2\}$ because $u_S^2 > u_S^1$, which can be easily seen. We have $I^{RR} = \{u^6\}$ because $u_S^6 > u_S^5$, which can readily be seen to be true given the condition $\lambda > \frac{\gamma}{1-\gamma}x$. It is easy to see that $I^L = \{u^2, u^3\}$ because u_S^3 is the largest of the three utilities for S and cannot be used to threaten for any other. On the RHS, we have an analogous situation; it is immediate that u^4 cannot be used for threats. Thus $I^R = \{u^4, u^6\}$. At the root, we notice that neither u^4 nor u^6 can be used to threaten for u^1. Thus, $u^2, u^3, u^4, u^6 \subseteq I$ and $u^1 \notin I$. It is not clear if $u_S^5 > u_S^2$, but we can immediately see that u^5 will not be the optimal choice for B if it is in I. In fact we see that the only possibly positive inducible choices for B are u^2 and u^3. We proved $u_B^3 > u_B^2$ above. Thus B picks u^3, which again coincides with the SPE. Since both arrangements of contracts yield the SPE, it follows from Theorem 7.9 that the game is full Stackelberg resilient. □

Theorem 7.9. *Both contracts are Stackelberg 1-resilient.*

Proof. For Contract 3.3, the result follows from Theorems 7.6 and 7.8. For Contract 3.1, if the sole contract is owned by S, there is no action of the buyer between when S determines its contract and when S moves. Thus the contract makes no difference in the game and it reduces to the SPE. By the proof of

Theorem 7.6, given that the order B-S is 2-resilient, we have that a B-contract will coincide with the SPE. □

In order for a Stackelberg attack to be feasible and worthwhile, there needs to be a reachable threat and a more desirable outcome for the threatening agent. In Contract 3.1, we see that threatening not to send is a viable threat against the buyer, regardless of what the buyer later plays. This can be used to threaten the buyer into an erroneous dispute resulting in them losing their deposit. Contract 3.3 has the further mechanism of the oracle, which both weakens the threat and removes the incentive for S to attempt to instigate a different outcome. Since the oracle has some error rate, there is a chance that it punishes S if B untruthfully disputes, thus removing any benefit to S of the threat. The threat is also no longer viable, that is $u_B^5 > u_B^2$, given further conditions in Contract 3.3. Thus Contract 3.3 frustrates this type of attack both from the demand prospective and the supply.

7.4 Auctions and Transaction Fee Mechanisms

In this section, we demonstrate the existence of a Stackelberg attack on auctions and the transaction fee mechanisms that are used by most major blockchains. This section is entirely based on [165], with only minor modifications to the text.

Consider n agents participating in an auction with m copies of the same item. Each agent i receives utility $v_i > 0$ by obtaining one of the copies. Assume that all v_i are distinct and ordered $v_1 < v_2 < \cdots < v_n$. Each agent places a bid $b_i \geq 0$ and the m agents with the highest bids receive a copy of the item, at the cost of paying some function of the bids. If there are multiple agents with the same bid, the mechanism chooses uniformly at random among these agents. If $m \geq n$ then all agents receive a copy of the item, in which case the optimal strategy for each agent is to bid $b_i = 0$. Thus, we will assume that $n = (1 + \alpha)\, m$ for some *congestion constant* $\alpha > 0$.

In a first-price auction, an agent pays their own bid which results in untruthful behavior: it is well-known that the best response for an agent i is to slightly outbid agent $n - m$ if their valuation exceeds this bid. That is, agent i will place the following bid.

$$b_i = \begin{cases} v_{n-m} + \varepsilon & \text{if } i > n - m, \\ 0 & \text{if } i \leq n - m. \end{cases} \tag{7.1}$$

Where $\varepsilon > 0$ is some small constant, representing a negligible amount of money. It is not hard to see that this bidding strategy is indeed an equilibrium (at least up to ε). Of course, this requires that the agents are able to estimate the valuations of other agents. In some applications, this might not be a realistic assumption. Instead, the mechanism can be made truthful by letting each

agent with a winning bid pay b_{n-m}, a second-price[1] auction [248]. In this case, it can be shown that the proposed mechanism is truthful so that each agent will bid their valuations [67, 127, 248]. While truthfulness is a desirable property, these auctions may be vulnerable to collusion [212].

The auction described is also known as a *transaction fee mechanism* and is used in blockchains to determine which transactions to include in the next block of data to be included in the chain [66]. Here, all pending transactions are public, so it is reasonable to assume agents know the valuations of other agents. Although blockchains canonically store transactions of cryptocurrency between different accounts [18, 58, 59], many blockchains have since generalized this to support arbitrary execution of code, so-called smart contracts [257]. Smart contracts are decentralized programs that run on a virtual machine implemented by the blockchain. A smart contract maintains state, can transfer funds between agents, and responds to queries. A smart contract is guaranteed to be faithful to its implementation by security of the underlying blockchain [152, 153].

We proceed to study transaction fee mechanisms involving agents who can universally commit to strategies such as by using smart contracts, either on the blockchain in question or a parallel one. As demonstrated previously, this can alter the equilibrium of the game which leads us to ask whether or not transaction fee mechanisms are vulnerable to these attacks. Indeed, we show that these commitments drastically change the structure of equilibria for various types of auctions, thus showing they are vulnerable to a Stackelberg attack wherein the buyers spontaneously organize to conspire against the auctioneer. In the attack, some agent commits to a strategy that ensures that they receive one of the items for free, while the remaining agents enter into a lottery for the remaining space on the block. The attack benefits all the buyers, but is detrimental to the auctioneer, who stands to lose most of their revenue. Note that while blockchains and smart contracts provide a natural setting in which to study these attacks, in principle the same framework can be used to analyze any setting in which agents can credibly commit to strategies, e.g. through reputation or by staking money. Understanding these attacks may also be important in predicting the behavior of advanced intelligent systems that have access to the internet (and hence access to a blockchain). We stress that the effectiveness of this attack is limited to the effectiveness of the commitment strategy. If the commitments can be undermined, say by the auctioneer, then so too can the attack be undermined.

An Attack on EIP-1559. We demonstrate the existence of a Stackelberg attack on the transaction fee mechanism EIP-1559, which is used by Ethereum. This mechanism is a generalization of first-price auctions intended to fix various

[1]Technically, the auction should be called a $(n-m)^{\text{th}}$-price auction, or a Vickrey auction; we stick to second-price for simplicity.

problems with first-price auctions in the context of transaction fee mechanisms [212]. By corollary, we show an attack on first-price auctions, which are used as transaction fee mechanisms in most other blockchains. The attack allows any agent to ensure they receive the item almost for free, while forcing (most of) the other agents to participate in a lottery for the remaining items. The attack works as long as the valuations are concentrated, in the sense that (most of) the largest values are not too much larger than the middle values. In this case, each agent voluntarily chooses the lottery because doing so will award them the item for free at some cost, while in the auction they would have to pay an amount commensurate with their valuation. If instead the valuations were spread out, the agents with a high valuation would not participate because they would be getting the item for a price much lower than their valuation, but with a degree of uncertainty. This is shown by explicitly demonstrating a strategy that an agent may commit to for which the equilibrium involves most agents entering into a lottery as described. The strategy extends also to second-price auctions. We evaluate the economic efficiency of this new situation and show that, while the attack benefits all users, it is detrimental to the auctioneer. This impact on auctioneer suggests that successful and widespread deployment of the attack would be detrimental to the viability of running the auctions. Therefore, our analysis is grounds for reevaluation of the auctions for transaction fee mechanisms. Formally, we define the *price of defiance* as the ratio between the utility an agent receives by cooperating versus the utility they would receive by deviating (or *defying* the attacker). We give a probabilistic bound on the price of defiance for the attack. Finally, we show that the conditions required to apply the attack are natural, in the sense that they are satisfied with high probability at certain levels of congestion when the valuations are sampled from two natural distributions.

The problem we study is natural in Web3 systems where agents natively interact using a blockchain. Thus, the agents are capable of deploying smart contracts that commit them to placing certain bids. In particular, the setting of an auction with multiple identical items models the transaction fee mechanisms that are used by blockchains to determine which transactions to include in the next block. Our work demonstrates that these mechanisms, in theory, are vulnerable to these attacks and may be cause for re-evaluation of the use of auctions in transaction fee mechanisms, at least when the networks are not too congested. Our work highlights the difficulty in designing smart contracts and suggests that other smart contracts that have already been deployed on major blockchains may be susceptible to Stackelberg attacks.

Properties of Transaction Fee Mechanisms. In recent years, there has been increased interest in analyzing blockchain transaction fee mechanisms using techniques from classic mechanism design. A line of work, [166, 212], identifies three desiderata of such mechanisms:

1. *user-incentive compatibility (UIC).* The users are incentivized to bid truthfully;
2. *miner-incentive compatibility (MIC).* The miners are incentivized to implement the mechanism as prescribed;
3. *off-chain agreement proofness (OCA proofness).* No coalition of miners and users can increase their joint utility by deviating from the mechanism.

In [212], Roughgarden shows that EIP-1559 satisfies MIC and OCA proofness when the block size is large and shows that it is not UIC, in the sense that users may benefit by bidding strategically. Here, OCA proofness means that the users and the miner cannot benefit by agreeing to off-chain payments and thus captures a specific type of commitment to strategies. Chung and Shi [66] show that no mechanism can simultaneously be UIC and 1-OCA proof. These results are shown in a model where agents cannot universally commit to strategies, and indeed we show that, arguably, neither of these three properties hold in a model where the agent can universally commit to strategies.

Modeling of Auctions. We will consider a set of n transactions competing for space on a block of size m. We assume for simplicity that each transaction is owned by exactly one agent, which we identify with the integers $\{1, 2, \ldots, n\}$. Each agent i has a valuation $v_i > 0$ of their transaction, which is the utility they gain by having their transaction included in the block for free. We assume the agents are rational, risk-neutral, and have a quasi-linear utility functions. We will take each v_i to be sampled i.i.d. from some known underlying distribution D. It will be convenient to assume that agents know each others' valuations precisely, i.e. we assume the values $v_1, v_2, \ldots, v_n$ are public and known to all the agents. Although this assumption is false in practice, by fixing D, the agents can mostly infer the valuations of the other agents, as these values tend to be concentrated around their expectations (if the number of agents is sufficiently large). This approach is used in practice on Ethereum, where several services provide tip estimations based on the current network congestion [92].

We assume each agent is capable of deploying a smart contract capable of bidding on their behalf, and that can condition on the smart contracts deployed by the other agents. To formalize this, we use the model of Chapter 6. First, fix some extensive-form representation of the sealed-bid auction, which could be done as follows: (1) choose an arbitrary order of the n agents, (2) construct the n-horizon game with agents in the specified order and where each layer sees the corresponding agent make a bid, which is born out in the ensuing subgame, (3) add information sets to ensure agents are not aware of the bids made by the other agents, (4) add utility vectors corresponding to the type of auction (first-price, second-price, etc.). Then, add 'smart contract moves' to the top of the game tree for each agent. These moves are special nodes that are

syntactic sugar for the larger 'expanded tree' that results from computing all appropriate cuts in the game tree and reattaching them with a node belonging to that agent. Expanding these moves in a bottom-up fashion yields a natural way for contracts to condition on the contracts deployed by other agents and is shown to generalize (reverse) Stackelberg equilibria. For more details, we refer to Chapter 6, though we trust that the intuitive understanding of 'contracts that depend on other contracts' suffices for the purposes of this work. An auction that is weakly strategically equivalent (i.e. the equilibrium payoffs are equal) to itself with smart contract moves is said to be *Stackelberg resilient*.

We now give our model of the transaction fee mechanism EIP-1559 used by Ethereum since 2021[2]. It generalizes first-price auctions by including a *base fee* $B \geq 0$ that each agent has to pay, which is burned. The base fee is continuously adjusted by the network to balance the demand to ensure each block is half full (in expectation). A first-price auction with m identical items is retained as a special-case when $B = 0$.

Mechanism 7.10. (EIP-1559).

1. *Each agent $i \in [n]$ submits a transaction of value $v_i > 0$ and makes a deposit of $B + \tau_i$ funds where $\tau_i \geq 0$ is an optional tip.*

2. *A miner finds a block, and selects a $T \subseteq [n]$ with $|T| = m$ that maximizes $\sum_{i \in T} \tau_i$. If there are multiple such T's, it selects T uniformly at random from all suitable sets.*

3. *Each agent $i \in T$ has their transactions included in the block and pays their deposit, in total gaining $v_i - B - \tau_i$ money; each agent $j \notin T$ is returned their deposit of $B + \tau_j$ currency and gains 0.*

4. *The miner receives $\sum_{i \in T} \tau_i$ currency.*

5. *The network adjusts the base fee B depending on m and n.*

In keeping with auction terminology, moving forward we will refer to the miner as the *auctioneer*. As per the introduction, we will let $n = (1 + \alpha)\, m$ for some *congestion constant* $\alpha > 0$. Let $\varepsilon > 0$ be the smallest unit of currency, and assume it is sufficiently small, i.e. $\varepsilon \ll v_i$, to mostly be ignored in calculations. In practice, on Ethereum, as of 2022, we have $\varepsilon \approx €10^{-12}$.

[2]In practice, the block size of EIP-1559 is variable and we shall let m denote its maximum possible value. In practice, the base fee would be adjusted to ensure that $\mathbb{E}[n] = m/2$, however the case of $n \leq m$ is not interesting (as all transactions will simply be included) so we take m to be the maximum value and assume $n > m$.

Modeling the Attack. We now propose a Stackelberg attack on Mechanism 7.10: essentially, the leading contract agent commits to paying 2ε, conditioned on everyone else committing to bidding ε. In this case, the leading contract agent has their transaction included at almost zero cost, while everyone else enters into a lottery. If anyone does not comply, the leading contract agent instead submits the bid they would have submitted without the contracts, or one slightly higher. This forces each other agent to decide between a lottery and a first-price auction. We will show that when the valuations of the transactions are somewhat concentrated, the agents prefer the lottery over the first-price auction, as they would otherwise have to pay a bid commensurate with their valuation, while in the auction they may receive the item for free.

As a warm-up and ongoing example, we look at the case where there are three agents and two slots up for auction, that is $n = 3$ and $m = 2$. This models a case where there are three buyers who wish to purchase two identical items — we may imagine these big buyers to be exchanges that control large quantities of user transactions, such as Coinbase or Binance, which are juggernauts in the industry [5]. Note that in this example we have $\alpha = \frac{1}{2}$. Suppose that agents $1, 2, 3$ have valuations $0 < v_1 < v_2 < v_3$, respectively. In a first price auction, where the valuations of the respective agents are known, the m agents with the highest valuations only need to outbid the agent with $m + 1$ highest valuation, who is unwilling to bid beyond their valuation and receive negative utility. In our example, agents 2 and 3 will bid slightly higher than the valuation of agent 1, yielding the following utilities: $u_1 = 0$, $u_2 = v_2 - v_1 - \varepsilon$, $u_3 = v_3 - v_1 - \varepsilon$.

We now equip these three agents with contracts. If the agent with the leading contract can make a credible and enforceable threat with the contract, they may force other agents to accept the lottery at the price ε, thereby guaranteeing the leading agent space an item at price of 2ε. The viability of such a threat depends on the agents' valuations. Agents will only comply if their expected utility is higher when they cooperate compared to when the threat is executed.

Consider first the case when agent 3 is the leading contract agent. The contract will commit agent 3 to bidding either 2ε, if the two other agents commit to playing ε, or to bidding the usual first price bid of $v_1 + \varepsilon$ otherwise. If the contract works, agent 3 enjoys utility $v_3 - 2\varepsilon$, a better result than the first price auction utility of $v_3 - v_1 - \varepsilon$. The desirable outcome is also clear for agent 1: the lottery case yields utility $\frac{1}{2}(v_1 - \varepsilon)$, which is better than the first price auction utility of 0. Therefore, both 1 and 3 will submit to the contract. Agent 2 will cooperate if the first price utility is lower than the lottery utility, that is if $v_2 - v_1 - \varepsilon < \frac{1}{2}(v_2 - \varepsilon)$, which reduces to $v_1 + \frac{1}{2}\varepsilon > \frac{1}{2}v_2$. The attack would not work if the valuations were less concentrated. If agent 2 is the lead contract holder, the attack works if $v_1 + \frac{1}{2}\varepsilon > \frac{1}{2}v_3$, a more stringent concentration requirement. If agent 1 has the leading contract, they may threaten to bid $v_2 + \varepsilon$, knowing they will likely not have to pay it. In this scenario, agent 1 has a credible threat if $v_2 + \frac{1}{2}\varepsilon > \frac{1}{2}v_3$, similar to the conditions for agent 3.

The attack generalizes readily to a larger number of agents, although the requirement on the valuations becomes stronger with more agents. In particular, the attack no longer works if even a single agent has a valuation that is significantly higher than the median. However, the leading contract agent may persuade such agents to participate by promising them a free item from the auction, taking some of the spots intended for the lottery. We denote by $C \subseteq [n]$ the *coalition* of agents (with $|C| = k$ for some $k < m$) who are given free items. This significantly loosens the valuation requirement and allows us to show that the attack works even if $k < m$ of the agents have large valuations. The set C may also be used to capture those agents who are oblivious to the attack, thus modeling the realistic scenario where some of the agents are not aware of the attack and cannot respond accordingly. We have not explicitly accounted for this; doing so would give a slightly stronger bound in the following, but would not fundamentally change the analysis. We now describe the attack in more detail.

Theorem 7.11. *Consider m identical items, and let $\varepsilon \ll v_1 < v_2 < \cdots < v_n$ be the valuations of the n buyers, with $n = (1+\alpha)\, m$ for some $\alpha > 0$. If for some $k < m$ it holds that,*

$$\frac{v_{n-k+1} - B}{v_{n-m}} < \frac{n-k}{n-m}, \tag{7.2}$$

then EIP-1559 is not Stackelberg resilient.

Proof. Assume that each agent has exactly one transaction, and let agent i be the agent associated with the transaction of valuation v_i. Suppose the contract agents are ordered $i_1, i_2, \ldots, i_n$, where i_1 is the leading contract agent. Now consider the following contract A_u^C, parameterized by an integer $u \in [n]$ that represents the index of the contract order and a set $C \subseteq [n]$ with $i_1 \in C$ and $|C| = k$ for some $k \leq m$.

Contract 7.12. (A_u^C).

1. *If $u = n$, play ε.*
2. *If $u < n$, play $v_{n-m} + \varepsilon$ if $v_{i_u} > v_{n-m} + \varepsilon$ and 0 otherwise in every subgame where any agent i_v with $v > u$ does not play the contract A_v^C; otherwise play 2ε if $u \in C$, and ε if $u \notin C$.*

Now suppose the leading contract agent deploys the contract A_1^C with $|C| = k < m$ and $i_1 \in C$. If they are successful, their transaction will be added with certainty for a cost of 2ε, thus gaining $v_{i_1} - 2\varepsilon$. Consider the strategy of agent j when every other agent submits, that is, plays Contract 7.12. If $j \in C$, then clearly for small ε, agent j will comply with the threat. If instead $j \notin C$, they will play Contract 7.12 to obtain a value of $v_j - \varepsilon$ with probability $\frac{m-k}{n-k}$. If they do not play Contract 7.12, all agents revert to a first-price auction, in accordance with their contracts. Then agent j can either bid too little to win

or bid at least $v_{n-m} + \varepsilon$ to have their transaction included. If $j \leq n - m$, this exceeds their valuation, and will thus prefer Contract 7.12, as its expected payoff is $\frac{(m-k)(v_i - B - \varepsilon)}{n-k} > 0$. If instead $j > n - m$, they can choose not to comply with the threat to gain $v_j - v_{n-m} - B - 2\varepsilon$ utility. It follows that such an agent will comply with the threat if $v_j - v_{n-m} - B - \varepsilon > \frac{(m-k)(v_j - B - \varepsilon)}{n-k}$, which, when ignoring εs, solves to $\frac{v_j - B}{v_{n-m}} < \frac{n-k}{n-m}$. But this is guaranteed to hold by Eq. (7.2), since $v_j \leq v_{n-k+1}$ for any j. Thus, complying with the threat is an equilibrium, implying EIP-1559 is not Stackelberg resilient. □

Note that by letting $B = 0$ we obtain a regular first-price auction, and hence Theorem 7.11 implies that the transaction fee mechanisms of Ethereum, Bitcoin, and most other blockchains are not Stackelberg resilient, regardless of whether there is a base fee or not. We observe that the attack works also for second-price auctions.

Theorem 7.13. *Consider a second-price auction with m identical items, and n buyers, in keeping with Theorem 7.11. If Eq.* (7.2) *holds, then the auction is not Stackelberg resilient.*

Proof (sketch). Consider the same attack, Contract 7.12. As we have seen, in the first price setting, bidders who have perfect information must only bid just enough to outbid the $(n-m)^{\text{th}}$ highest agent, with a bid of $v_{n-m} + \varepsilon$. In the first price auction, these agents will be charged the amount they bid. In the second price auction, they can either bid their valuation or stick with $v_{n-m} + \varepsilon$. In any case, if they are included, the agent will pay v_{n-m}, a slight discount on the $v_{n-m} + \varepsilon$ cost in the first price setting. Thus Contract 7.12 can be used, and the scenario in which the attack works will look the same. If the attack does not work, agents revert to the equilibrium as it would be without contracts. This equilibrium would have the slightly different, second price cost. Note that, in the proof of Theorem 7.11, we drop the epsilons that constitute the difference between the first and second price auctions. So by the proof of Theorem 7.11, second price auctions are also not Stackelberg resilient. □

Risk Aversion. It is natural to wonder if the attack will still work if the agents are risk averse. To model risk aversion, agents have some concave utility function $u = U(\cdot)$. If, for example, an agent gets a slot for free at valuation v_i, their utility would be defined to be $u = U(v_i)$. For $U(\cdot)$ to be concave, we must have $U((1-p)x + py) \geq (1-p)U(x) + pU(y)$ where $(x, U(x))$ and $(y, U(y))$ are two points on the utility function and $p \in [0, 1]$. Graphically, this implies that any point on the line between $(x, U(x))$ and $(y, U(y))$ is on or below the utility curve. This straight line below the curve traces out the utility of a coin toss with probability p between $U(x)$ and $U(y)$. This models risk aversion because the utility of any outcome based on a coin toss between two outcomes will be on or below the curve, which in turn represents the utility of outcomes

that are certain. If we make the assumption that $x = U(x) = 0$ and set $y = v_i$, we have $U(pv_i) \geq pU(v_i)$. Note that, in the proof of Eq. (7.2), we required the condition, here simplified, that $v_i - v_{n-m-k+1} > pv_i$. If we instead had some concave utility function, this would be $U(v_i - v_{n-m}) > pU(v_i)$. Given that $U(pv_i) \geq pU(v_i)$, the condition found in Eq. (7.2) is necessary, but not necessarily sufficient, for the contact attack to still be viable. Finding the exact condition requires $U(\cdot)$ to be known.

Everyone Benefits Except for the Auctioneer

In the following, we will assume that $k = 1$ and that $\varepsilon = 0$. As k increases, more agents with high valuations get free entry when $\varepsilon = 0$. Thus, their relatively high valuations are counted into social welfare. As long as this elite group is relatively small, this will have little impact on the chances of the lottery agents, meaning allowing a relatively small $k > 1$ would only increase social welfare.

We define the *price of defiance*, a ratio of sets of equilibrium that is related to the price of anarchy [157]. Let S be the set of all strategy profiles in the game and take two sets $C \subseteq S$, some set of strategies, and $E \subseteq S$, the set of all equilibria of the game. We take the set C to be the set of equilibria after a successful contract attack has been deployed. Define,

$$PoD = \frac{\max_{s \in C} \mathrm{Welf}(s)}{\min_{s \in E} \mathrm{Welf}(s)}. \tag{7.3}$$

This is the ratio between the best of a subset of possible outcomes and the worst equilibrium. It differs from the price of anarchy in that we compare some subset of strategies, here those that become equilibria due to the introduction of a contract attack, rather the optimal solution, to the game's usual equilibria. We have $PoD \leq PoA$.

Our set C is the set of equilibrium arising from agents having and complying with Contract 4.1. There are up to n equilibria in the set, one for each choice of agent with leading contract. To analyze the price of defiance we will need concentration bounds on the valuations of the agents. Order the agents with valuations $v_1 < v_2 < \ldots < v_n$, then $v_i \sim \mathrm{Beta}(i, n+1-i)$. Say a function f is *negligible* if $f(x) = o(x^c)$ for every constant $c \in \mathbb{R}$, i.e. if it grows slower than the inverse of any polynomial. We will make use of the following concentration bound on order statistics from the uniform distribution.

Lemma 7.14 (Skorski, [231])**.** *Let $X \sim Beta(\alpha, \beta)$ for $\alpha, \beta > 0$, and define,*

$$v^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+2)}, \qquad c_0 = \frac{|\beta - \alpha|}{(\alpha+\beta)(\alpha+\beta+2)}.$$

Then for any $\varepsilon > 0$, it holds that,

$$\Pr\left[|X - \mathbb{E}[X]| > \varepsilon\right] \leq 2\exp\left(-\frac{\varepsilon^2}{2v^2 + 2\varepsilon \max\{v, c_0\}}\right).$$

Lemma 7.15. *Let $X_1, X_2, \ldots, X_n \sim U[0,1]$, and let $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$ be the n order statistics. Then,*

$$\left| X_{(i)} - \frac{i}{n+1} \right| = \tilde{O}(1/n), \qquad \text{for every } i = 1 \ldots n,$$

except with negligible probability in n.

Proof. We make use of Lemma 7.14 to bound the error term and must therefore first find the relevant values of v and c_0. It is a fact that such order statistics have the distribution $\text{Beta}(i, n+1-i)$, in particular $\alpha = i$ and $\beta = n+1-i$. Thus, for all values of i, we must have $\alpha + \beta = n+1$. It is easy to see that we find the largest value v^2 from Lemma 7.14 when $\alpha = \beta = \frac{n+1}{2}$. This case yields

$$v^2 \leq \frac{\frac{n+1}{2}\frac{n+1}{2}}{\left(\frac{n+1}{2} + \frac{n+1}{2}\right)^2 \left(\frac{n+1}{2} + \frac{n+1}{2} + 2\right)} = \frac{1}{4(n+3)}.$$

The value of c_0 is largest when the numerator is largest, which is clearly when $|\beta - \alpha| = n-1$. Note that this is a specifically different case from when v^2 is largest. When we go on to find the error bounds on specific v_i's, we will refine the bound at this step. Thus, we have the following bounding value,

$$c_0 \leq \frac{n-1}{(n+1)(n+3)}.$$

It is easy to see that $c = \max\{v, c_0\} = c_0$. Thus, we can write down the bound for any i,

$$\begin{aligned}
\Pr\left[\left|X_{(i)} - \mathbb{E}[X_{(i)}]\right| > \delta\right] &< 2\exp\left(-\frac{\delta^2}{2v^2 + 2c\delta}\right) \\
&\leq 2\exp\left(-\frac{\delta^2}{2\frac{1}{4(n+3)} + \frac{2\delta(n-1)}{(n+1)(n+3)}}\right) \\
&= 2\exp\left(-\frac{\delta^2\, 2(n+3)(n+1)}{(n+1) + 4\delta(n-1)}\right) \\
&< 2\exp\left(-\frac{\delta^2\, 2n^2}{(n+1) + 4\delta n}\right) \\
&\approx 2\exp\left(-\frac{\delta^2\, 2n}{1 + 4\delta}\right) \\
&= 2\exp(-\Omega(\delta n)).
\end{aligned}$$

If we take $\delta = \frac{\log^2 n}{n+1} = \tilde{O}(1/n)$, we obtain the bound,

$$\Pr\left[\left|X_{(i)} - \mathbb{E}[X_{(i)}]\right| > \delta\right] < 2\exp(-\omega(\log n)), \tag{7.4}$$

which is negligible in n. We conclude with a union bound on all n valuations. □

Theorem 7.16. *For uniformly distributed valuations, the price of defiance is at least* $1 + \alpha - o(1)$*, except with probability negligible in* n*.*

Proof. It is easy to see that the maximal choice $s \in C$ occurs when the agent with the highest valuation has the contract. There is only one choice for equilibrium $s \in C$. Thus we have,

$$\begin{aligned} PoD &= \frac{\max_{s\in C} \text{Welf}(s)}{\min_{s\in E} \text{Welf}(s)} = \frac{\left(\sum_{j=1}^{n-1} \frac{m-1}{n-1}(v_j - \varepsilon)\right) + v_n - 2\varepsilon}{\left(\sum_{i=n-m+1}^{n} v_i - v_{n-m} - \varepsilon\right)} \\ &\approx \frac{\frac{m-1}{n-1}\left(\sum_{j=1}^{n-1} v_j\right) + v_n}{\left(\sum_{i=n-m+1}^{n} v_i\right) - m v_{n-m}}. \end{aligned} \tag{7.5}$$

If the contract attack works, that is if the valuations are in keeping with the condition from Theorem 7.11, we have $PoD > 1$. This can be seen mathematically by substituting the condition into the denominator of Eq. (7.5) above. Intuitively, given that the threat is just the usual first price auction, the other agents will acquiesce only if their utility would be higher in the lottery. Thus, total lottery welfare, the numerator, must be higher than the auction, the denominator, leading to a $PoD > 1$ in the general case. Each v_i is the i^{th} order statistic of a uniformly distributed random variable, that is $v_i = X_{(i)}$ where X_i is sampled i.i.d. from the uniform distribution on $[0, 1]$. By linearity of expectation, we have that,

$$\begin{aligned} \mathbb{E}\left[\sum_{i=n-m+1}^{n} v_i\right] = \sum_{i=n-m+1}^{n} \frac{i}{n+1} &= \frac{1}{n+1}\left(\sum_{i=0}^{n} i - \sum_{k=0}^{n-m} k\right) \\ &= \frac{n}{2} - \frac{(n-m)(n-m+1)}{2(n+1)}, \end{aligned}$$

and that,

$$\mathbb{E}\left[\sum_{j=1}^{n-1} v_j\right] = \sum_{j=1}^{n-1} \frac{j}{n+1} = \frac{(n-1)n}{2(n+1)}.$$

We proceed to lower bound PoD using Lemma 7.15 to yield,

$$\begin{aligned} PoD &\geq \frac{\frac{m-1}{n-1}\left(\frac{(n-1)n}{2(n+1)} - (n-1)\delta\right) + \frac{n}{n+1} - \delta}{\frac{n}{2} - \frac{(n-m)(n-m+1)}{2(n+1)} + m\delta - m\left(\frac{n-m}{n+1} + \delta\right)} \\ &= \frac{n(m+1) - 2m(n+1)\delta}{m(m+1) + 4m(n+1)\delta} \end{aligned}$$

We now condition on the errors of the valuations being bounded by $\delta = \frac{(m+1)\log^2 n}{2m(n+1)}$, which we know to happen except with negligible probability by

Lemma 7.15. Then we obtain the following bound,

$$PoD \geq \frac{n - \log^2(n)}{m + \log^2(n)} = 1 + \alpha - o(1),$$

as desired. □

Arguably, this suggests that lotteries should be used as transaction mechanisms when the valuations are believed to be of similar size. In the $n = 3$, $m = 2$ case, we have

$$PoD = \frac{\frac{v_1}{2} + \frac{v_2}{2} + v_3 - 3\varepsilon}{v_2 + v_3 - 2v_1}$$

If the condition for the contract to work from the example in Section 7.4 holds, that is, if $v_1 + \frac{1}{2}\varepsilon > \frac{1}{2}v_2$, the ratio becomes

$$PoD > \frac{\frac{v_1}{2} + \frac{v_2}{2} + v_3 - 3\varepsilon}{v_3 + \varepsilon},$$

which is clearly larger than one.

It is important to note that while the attack benefits all the agents with transactions, it is detrimental to the auctioneer, who loses essentially all of their revenue. Auctioneers that find themselves subject to such an attack might respond by not allowing the smart contracts to be deployed, but this could be remedied if agents use a different blockchain to deploy the attack. Auctioneers may also find themselves obligated to include the contract moves due to a staking scheme [47]. Continuing with our $n = 3$, $m = 2$ example, we readily see the auctioneer will earn $2(v_1 + \varepsilon)$ in the auction case. If the contract attack is successfully executed, the auctioneer income will be 3ε, 2ε from the leading contract holder, regardless of which agent this is, and ε from the winner of the lottery. Thus, almost all the revenue is lost; the auctioneer will miss out on $2v_1 - \varepsilon$ income. If there were a base fee and all agents had a valuation larger than said base fee, i.e. $v_1 > B$, the first price revenue would be $2(v_1 + \varepsilon - B)$. The lottery revenue will continue to be 3ε and the income lost to the attack will be $2\,(v_1 - B) - \varepsilon$.

The Attack Works for Natural Distributions

In this section, we show that the conditions required for the attack are satisfied with high probability under reasonable assumptions. We will assume that $B = 0$. The results obtained are qualitatively similar when the valuations are much larger than the base fee.

We continue with our illustration of the $n = 3$, $m = 2$ case, now assuming that the agents have valuations that are uniformly distributed on $[0, 1]$. As before, we have three valuations $v_1 < v_2 < v_3$ and we can now make use

of the distribution. The ordered valuations in are order statistics, that is $v_i = X_{(i)}$ where all X_i are sampled i.i.d. from the uniform distribution on $[0, 1]$. Using the well known fact that order statistics on the uniform distribution follow specific beta distributions, we get the following distributions and their expectations: $v_1 \sim \text{Beta}(1, 3)$ yielding, $\mathbb{E}[v_1] = \frac{1}{4}$; $v_2 \sim \text{Beta}(2, 2)$, yielding $\mathbb{E}[v_2] = \frac{1}{2}$; and $v_3 \sim \text{Beta}(3, 1)$, with $\mathbb{E}[v_3] = \frac{3}{4}$. Note that the variance for all the distributions is $\text{Var}[v_i] \leq 1/20$ and we will not take it into account moving forward. In the first price auction, we can see that if agents 2 and 3 bid just enough to outbid agent 1, i.e. $\frac{1}{4} + \varepsilon$, they will secure their slots as cheaply as possible. So in the first price auction the agents will have the expected utilities $\mathbb{E}[u_1] = 0$, $\mathbb{E}[u_2] = \frac{1}{4} - \varepsilon$, and $\mathbb{E}[u_3] = \frac{1}{2} - \varepsilon$.

If agent 1 has the leading contract, they can threaten to outbid agent 2 with a bid of $\frac{1}{2} + \varepsilon$. If the threat were to be carried out, agent 2 would lose their slot and receive utility 0 and agent 3, secure in the top spot, but given they outbid agent 2, will receive $\frac{1}{4} - \varepsilon$. If agents 2 and 3 comply with the threat, i.e. bid ε and enter a lottery, they will have expected utilities $\frac{1}{4} - \frac{\varepsilon}{2}$ and $\frac{3}{8} - \frac{\varepsilon}{2}$, respectively. These utilities are more desirable than ignoring the threat, and the attack can be executed. Agent 1 will enjoy an expected utility of $\frac{1}{4} - 2\varepsilon$. Note that agents 1 and 2 have higher utility than they would have had in the first price auction, but agent 3 is hurt by the attack.

If agent 2 has the leading contract, their best threat is outbidding agent 1 with a bid of $\frac{1}{4} + \varepsilon$. This is no threat at all as it simply coincides with their first price strategy. If instead agent 3 has the leading contract, we once again have a viable attack. Since agent 3 already outbids the others, their contract-endowed strategy is more a proposition for mutual benefit than a greedy attack. If the other two agents enter into a lottery at price ε and agent 3 bids 2ε, we have expected utilities $\mathbb{E}[u_1] = \frac{1}{8} - \frac{\varepsilon}{2}$,$\mathbb{E}[u_2] = \frac{1}{4} - \frac{\varepsilon}{2}$, and $\mathbb{E}[u_3] = \frac{3}{4} - 2\varepsilon$. It can be easily seen that everyone benefits in this situation and the attack will work. It is an easy calculation to find that $PoD \approx 3/2$. Regardless of which agent has the leading contract, if the attack works, the total tip paid to the auctioneer will be 3ε. In the first price auction, the expected auctioneer payout is $\frac{1}{2} + 2\varepsilon$. The difference constitutes an almost complete loss of revenue.

Lemma 7.17 (Xu, Mei, Miao, [259])**.** *Let $X_1, X_2, \ldots, X_n \sim U(0, 1)$ be i.i.d. Let $i < j$ and define $R_{ij} = \frac{X_{(j)}}{X_{(i)}}$ and let $f(\cdot)$ be its density function with support $[1, \infty)$. Then for every $r \geq 1$,*

$$f(r) = \frac{n!(r-1)^{j-i-1}}{(i-1)!(j-i-1)!(n-j)!r^j} \int_0^1 (1-u)^{j-1} u^{n-j} \, \mathrm{d}u.$$

Theorem 7.18. *Suppose n buyers participate in an auction of m identical items where $n = (1 + \alpha) m > m + 1$. If the valuations of the items are sampled uniformly at random and $0 \leq \alpha < 0.53$, then first-price auctions are not Stackelberg resilient, except with probability negligible in m.*

Proof. We will show that Eq. (7.2) holds except with probability $\mathsf{negl}(n)$. Suppose w.log. that the valuations are sampled uniformly from $[0,1]$ and let $v_1 < v_2 < \cdots < v_n$ be their valuations. The value v_i equals the i^{th} order statistic, the distribution of which is well-known for uniform values. We are interested in the ratio $R = v_{n-k+1}/v_{n-m}$, so let $f(\cdot)$ be its density function. Let $k = m\delta$ for some $0 < \delta < 1$. By Lemma 7.17, noting that we have $j = (1+\alpha-\delta)m+1, i = \alpha m$, we get that,

$$\begin{aligned} f(r) &= \frac{n!\,(r-1)^{m-1}}{(\alpha m-1)!\,z((1-\delta)m)!\,r^n}\int_0^1 (1-u)^{(1+\alpha-\delta)\,m-1}u^{\delta m-1}\,\mathrm{d}u \\ &= \frac{((1+\alpha-\delta)\,m)!}{((1-\delta)\,m)!\,(\alpha m-2)!}\frac{(r-1)^{(1-\delta)\,m}}{r^n}. \end{aligned}$$

We denote by $H(p) = -p\lg p - (1-p)\lg(1-p)$, the binary entropy function, defined on $[0,1]$. Note that $H(p) \le 1$ for every $p \in [0,1]$. A useful upper bound is given by the following.

$$H(x) \le 2\sqrt{x\,(1-x)}. \tag{7.6}$$

The binary entropy function is useful because it allows us to upper bound the binomial coefficient as follows.

$$\binom{n}{k} \le 2^{nH(k/n)} \tag{7.7}$$

We bound the probability that Eq. (7.2) does not hold as follows.

$$\begin{aligned} &\Pr\left[R' \ge \frac{n-k}{n-m}\right] = \int_{\frac{1+\alpha-\delta}{\alpha}}^{\infty} f_R(r)\,\mathrm{d}r \\ &= \frac{((1+\alpha-\delta)\,m)!}{((1-\delta)\,m)!\,(\alpha m-2)!}\int_{\frac{1+\alpha-\delta}{\alpha}}^{\infty}\frac{(r-1)^{(1-\delta)\,m}}{r^n}\,\mathrm{d}r \\ &\le \frac{\alpha}{\alpha+\delta}\binom{(1+\alpha-\delta)\,m}{\alpha m}\left(\frac{1+\alpha-\delta}{\alpha}\right)^{1-(\alpha+\delta)\,m} \end{aligned}$$

We now apply Eq. (7.7) and collect the terms in the exponent.

$$\begin{aligned} &\le \frac{\alpha}{\alpha+\delta}\exp\left(H\left(\frac{\alpha}{1+\alpha-\delta}\right)(1+\alpha-\delta)\,m\right. \\ &\qquad\left. + \log\left(\frac{1+\alpha-\delta}{\alpha}\right)(1-(\alpha+\delta)m)\right) \end{aligned}$$

We use the fact that $H(p) \le 2\sqrt{p(1-p)}$ as per Eq. (7.6) to obtain,

$$\begin{aligned} &\le \frac{\alpha}{\alpha+\delta}\exp\left(\log\left(\frac{1+\alpha\delta}{\alpha}\right)\right. \\ &\qquad\left. + m\left[2\sqrt{\alpha(1-\delta)} - (\alpha+\delta)\log\left(\frac{1+\alpha\delta}{\alpha}\right)\right]\right). \end{aligned}$$

We note that the exponent is negative for sufficiently large m, and hence the probability is negligible if,

$$1 + \alpha - \delta - (\alpha + \delta) \log \left(\frac{1 + \alpha\delta}{\alpha} \right) < 0.$$

Which solves to $0 < \alpha < 0.529914$ for $\delta = 0.69$. □

Lemma 7.19 (Adler, [2])**.** *Let $X_1, X_2, \ldots X_n$ be i.i.d. Pareto distributed with parameter $p > 0$. Let $i < j$ and define $R_{ij} = \frac{X_{(j)}}{X_{(i)}}$ and let $f(\cdot)$ be its density function with support $[1, \infty)$. Then for every $r \geq 1$,*

$$f(r) = \frac{p\,(n-i)!}{(j-i-1)!(n-j)!} \left(1 - \frac{1}{r^p}\right)^{j-i-1} \frac{1}{r^{p(n-j+1)+1}}.$$

Theorem 7.20. *Suppose n buyers participate in an auction of m identical items where $n = (1+\alpha)\,m$ for some $\alpha > 0$. If the valuations of the items are sampled according to a Pareto distribution with parameter $p > 1$ and $0 \leq \alpha \leq \alpha(p) < 0.69$, then first-price auctions are not Stackelberg resilient, except with probability negligible in m.*

Proof. Suppose for the sake of the argument that n is even, and let C be the $m\delta$ agents with the largest valuations for some constant $0 < \delta < 1$. Let $R = (v_{n-k+1}/v_{n-m})$ and let $f(\cdot)$ be its density function. By Lemma 7.19, it is given by,

$$\begin{aligned} f(r) &= \frac{pm!}{(m-k-2)!(k-1)!} \left(1 - \frac{1}{r^p}\right)^{m-k-2} \frac{1}{r^{pk+1}} \\ &= pk(m-k-1)(m-k)\binom{m}{k} \left(1 - \frac{1}{r^p}\right)^{m-k-2} \frac{1}{r^{pk+1}}. \end{aligned}$$

We bound the probability that Eq. (7.2) does not hold as follows.

$$\begin{aligned} &\Pr\left[R \geq \frac{n-k}{n-m}\right] \\ &= pk(m-k-1)(m-k)\binom{m}{k} \int_{\frac{1+\alpha-\delta}{\alpha}}^{\infty} \frac{\left(1 - \frac{1}{r^p}\right)^{m-k-2}}{r^{pk+1}}\,\mathrm{d}r \\ &\leq pk(m-k-1)(m-k)\binom{m}{k} \int_{\frac{1+\alpha-\delta}{\alpha}}^{\infty} \frac{1}{r^{pk+1}}\,\mathrm{d}r \\ &= m((1-\delta)m-1)(1-\delta)\binom{m}{\delta m} \left(\frac{1+\alpha-\delta}{\alpha}\right)^{-p\delta m} \end{aligned}$$

We now bound the binomial coefficient using Eq. (7.7) and collect the terms in the exponent.

$$\leq m((1-\delta)m-1)(1-\delta) \exp\left(m\left[H(\delta) - \delta p \log\left(\frac{1+\alpha-\delta}{\alpha}\right)\right]\right)$$

We note that the exponent is negative, and hence the function negligible, if the following inequality is satisfied.

$$\delta p \log\left(\frac{1+\alpha-\delta}{\alpha}\right) > H(\delta).$$

By Eq. (7.6), it suffices then to establish the following bound.

$$\delta p \log\left(\frac{1+\alpha-\delta}{\alpha}\right) > 2\sqrt{\delta(1-\delta)}.$$

We now let $\delta = \frac{5}{p^2+4}$, and note that this inequality is satisfied for any $p > 1$ whenever the following inequality holds.

$$0 < \alpha < \frac{p^2-1}{(4+p^2)\left(\exp\left(\frac{2\sqrt{\frac{p^2-1}{p^2}}}{\sqrt{5}}\right)-1\right)}$$

Denote the rhs by $\alpha(p)$. Note that $\alpha(p) > 0$ for any $p > 1$ and evaluates to $\frac{1}{2}(\coth(1/\sqrt{5}) - 1) \approx 0.69$ in the limit as $p \to \infty$. □

The Pareto distribution, which follows the 80/20 rule, is the more natural distribution in this context. It is widely used in economics and it makes intuitive sense that transactions, and therefore valuations, would tend to be small, with a tail of rarer, but large transactions. The uniform distribution can be seen as a lower bound, because the real distribution would have fewer high valuation transactions, and this can only favor the attack.

Conclusion

Bibliography

[1] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, page 53–62, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933840. doi: 10.1145/1146381.1146393. URL https://doi.org/10.1145/1146381.1146393. 37

[2] André Adler. Limit theorems for arrays of ratios of order statistics. *BULLETIN-INSTITUTE OF MATHEMATICS ACADEMIA SINICA*, 33(4):327, 2005. 135

[3] John Adler, Ryan Berryhill, Andreas Veneris, Zissis Poulos, Neil Veira, and Anastasia Kastania. Astraea: A decentralized blockchain oracle. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1145–1152, 2018. doi: 10.1109/Cybermatics_2018.2018.00207. 7, 77

[4] U.S. Energy Information Administration. Frequently asked questions (faqs) - how much electricity does an american home use?, 2022. URL https://www.eia.gov/tools/faqs/faq.php?id=97&t=3. 12

[5] Carol Alexander, Daniel F Heck, and Andreas Kaeck. The role of binance in bitcoin volatility transmission. *Applied Mathematical Finance*, 29(1): 1–32, 2022. 126

[6] M. Allais. Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'ecole americaine. *Econometrica*, 21 (4):503–546, 1953. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1907921. 29

[7] ArbStore. User guide for contracts with arbitration, 2023. URL https://arbstore.org/guide. 6

[8] James E Arps and Nicolas Christin. Open market or ghost town? the curious case of openbazaar. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pages 561–577. Springer, 2020. 6, 42

[9] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley & Sons, 1951. 7, 57

[10] Kenneth J Arrow. The theory of risk aversion. *Essays in the theory of risk-bearing*, pages 90–120, 1971. 27

[11] Aditya Asgaonkar and Bhaskar Krishnamachari. Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 262–267, 2019. doi: 10.1109/BLOC.2019.8751482. 20, 41, 43, 44, 45, 76, 81

[12] Gilad Asharov and Claudio Orlandi. Calling out cheaters: Covert security with public verifiability. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 681–698, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34961-4. 16, 74, 75, 88, 89

[13] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pages 86–99, 1998. doi: 10.1109/SECPRI.1998.674826. 81

[14] Robert J Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974. 26

[15] Robert J Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica: Journal of the Econometric Society*, pages 1–18, 1987. 26

[16] Yonatan Aumann and Yehuda Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. In Salil P. Vadhan, editor, *Theory of Cryptography*, pages 137–156, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-70936-7. 16, 74, 75, 88, 89

[17] Yurii Averboukh. Inverse stackelberg solutions for games with many followers. *Mathematics*, 6, 04 2014. doi: 10.3390/math6090151. 17, 18, 99

[18] Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? rational protocol design treatment of bitcoin. 4 2018. 4, 10, 11, 12, 122

[19] Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? rational protocol design treatment of bitcoin. In *Advances in Cryptology — EUROCRYPT 2018*, volume 10821 (Proceedings Part II) of *LNCS*, pages 34–65. Springer, 4 2018. 12, 13

[20] Aurélien Baillon. Bayesian markets to elicit private information. *Proceedings of the National Academy of Sciences*, 114(30):7958–7962, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1703486114. URL `https://www.pnas.org/content/114/30/7958`. 76

[21] Matthew Ball and Roderic Broadhurst. Data capture and analysis of darknet markets. *Available at SSRN 3344936*, 2021. 4

[22] T. Basar and H. Selbuz. Closed-loop stackelberg strategies with applications in the optimal control of multilevel systems. *IEEE Transactions on Automatic Control*, AC-24:166 – 179, 04 1979. 95

[23] Tamer Basar and Hasan Selbuz. Closed-loop stackelberg strategies with applications in the optimal control of multilevel systems. *IEEE Transactions on Automatic Control*, 24(2):166–179, 1979. 17

[24] Carsten Baum, James Hsin yu Chiang, Bernardo David, and Tore Kasper Frederiksen. Eagle: Efficient privacy preserving smart contracts. Cryptology ePrint Archive, Paper 2022/1435, 2022. URL `https://eprint.iacr.org/2022/1435`. `https://eprint.iacr.org/2022/1435`. 13

[25] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 1–10, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912640. doi: 10.1145/62212.62213. URL `https://doi.org/10.1145/62212.62213`. 16

[26] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, 2014. doi: 10.1109/SP.2014.36. 14

[27] Jeremy Bentham. *The Works of Jeremy Bentham*, volume 7. W. Tait, 1843. 25

[28] B Douglas Bernheim. Rationalizable strategic behavior. *Econometrica: Journal of the Econometric Society*, pages 1007–1028, 1984. 29

[29] Daniel Bernoulli. Commentarii academiae scientiarum imperialis petropolitanae. *Petropoli. Chap. De vibrationibus et sono laminarum elasticarum*, 1751. 27, 28

[30] Daniel Bernoulli. Exposition of a new theory on the measurement of risk. In *The Kelly capital growth investment criterion: Theory and practice*, pages 11–24. World Scientific, 2011. 28

[31] Giancarlo Bigi, Andrea Bracciali, Giovanni Meacci, and Emilio Tuosto. *Validation of Decentralised Smart Contracts Through Game Theory and Formal Methods*, pages 142–161. Springer International Publishing, Cham, 2015. ISBN 978-3-319-25527-9. doi: 10.1007/978-3-319-25527-9_11. URL `https://doi.org/10.1007/978-3-319-25527-9_11`. 77

[32] Stefano Bistarelli, Ivan Mercanti, Francesco Faloci, and Francesco Santini. Highlighting poor anonymity and security practice in the blockchain of bitcoin. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 265–272, 2021. 13

[33] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 326–349, 2012. 14

[34] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a nash equilibrium. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1480–1498, 2015. doi: 10.1109/FOCS.2015.94. 30

[35] Michael Bloem, Tansu Alpcan, and Tamer Basar. A stackelberg game for power control and channel allocation in cognitive radio networks. In *1st International ICST Workshop on Game theory for Communication networks*, 2010. 17

[36] Erica Blum, Aggelos Kiayias, Cristopher Moore, Saad Quader, and Alexander Russell. The combinatorics of the longest-chain rule: Linear consistency for proof-of-stake blockchains. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1135–1154. SIAM, 2020. 12

[37] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 15(1):23–27, January 1983. ISSN 0163-5700. 52

[38] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 103–112,

New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912640. doi: 10.1145/62212.62222. URL https://doi.org/10.1145/62212.62222. 14

[39] Hans L. Bodlaender. On the complexity of some coloring games. In Rolf H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science*, pages 30–40, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. ISBN 978-3-540-46310-8. 101, 107

[40] Romain Bourneuf, Lukáš Folwarczný, Pavel Hubáček, Alon Rosen, and Nikolaj Ignatieff Schwartzbach. PPP-Completeness and Extremal Combinatorics. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-263-1. doi: 10.4230/LIPIcs.ITCS.2023.22. 22

[41] Branislav Bošanský, Simina Brânzei, Kristoffer Arnsfelt Hansen, Troels Bjerre Lund, and Peter Bro Miltersen. Computation of stackelberg equilibria of finite sequential games. *ACM Trans. Econ. Comput.*, 5(4), December 2017. ISSN 2167-8375. doi: 10.1145/3133242. 18, 97

[42] Branislav Bošanský, Simina Brânzei, Kristoffer Arnsfelt Hansen, Troels Bjerre Lund, and Peter Bro Miltersen. Computation of stackelberg equilibria of finite sequential games. *ACM Trans. Econ. Comput.*, 5(4), December 2017. ISSN 2167-8375. doi: 10.1145/3133242. 17, 30

[43] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. 7, 57

[44] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, Sergey Nazarov, Alexandru Topliceanu, Floran Tramèr, and Fan Zhang. Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks. *v1.0*, 2021. https://research.chain.link/whitepaper-v2.pdf. 7

[45] M. Breton, A. Alj, and A. Haurie. Sequential stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications*, 59 (1):71–97, Oct 1988. ISSN 1573-2878. doi: 10.1007/BF00939867. 98

[46] Julian Broséus, Damien Rhumorbarbe, Caroline Mireault, Vincent Ouellette, Frank Crispino, and David Décary-Hétu. Studying illicit drug trafficking on darknet markets: structure and organisation from a canadian perspective. *Forensic science international*, 264:7–14, 2016. 4, 14

[47] Vitalik Buterin. Slasher: A punitive proof-of-stake algorithm, 2014. 132

[48] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017. 10

[49] Giulio Caldarelli. Understanding the blockchain oracle problem: A call for action. *Information*, 11(11):509, 2020. 7

[50] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 93–118, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44987-4. 4, 5

[51] Burak Can, Jens Leth Hougaard, and Mohsen Pourpouneh. On reward sharing in blockchain mining pools. *Games and Economic Behavior*, 136: 274–298, 2022. 12

[52] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001. doi: 10.1109/SFCS.2001.959888. 13

[53] Ioannis Caragiannis and Nikolaj I. Schwartzbach. Outsourcing adjudication to strategic jurors, 2023. 56

[54] Ioannis Caragiannis and Nikolaj Ignatieff Schwartzbach. Outsourcing Adjudication to Strategic Jurors. To appear in *32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*., 2023. 8, 22, 23, 56

[55] Ioannis Caragiannis, Ariel D. Procaccia, and Nisarg Shah. When do noisy votes reveal the truth? *ACM Transactions on Economics and Computation*, 4(3):15:1–15:30, 2016. 7, 8, 57

[56] Ignacio Cascudo and Bernardo David. Albatross: publicly attestable batched randomness based on secret sharing. Cryptology ePrint Archive, Report 2020/644, 2020. 53

[57] Arthur Cayley. Mathematical questions with their solutions. *The Educational Times*, 23:18–19, 1875. 36

[58] David Chaum. Verification by anonymous monitors. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, pages 138–139. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1981. 122

[59] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, pages 199–203. Plenum Press, New York, 1982. doi: 10.1007/978-1-4757-0602-4_18. URL https://doi.org/10.1007/978-1-4757-0602-4_18. 122

[60] David Chaum, Claude Crépeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 11–19, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912640. doi: 10.1145/62212.62214. URL https://doi.org/10.1145/62212.62214. 16

[61] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, 2006. doi: 10.1109/FOCS.2006.69. 30, 97

[62] Xi Chen, Xiaotie Deng, and Shang-hua Teng. Computing nash equilibria: Approximation and smoothed complexity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 603–612, 2006. doi: 10.1109/FOCS.2006.20. 30

[63] In-Koo Cho and David M Kreps. Signaling games and stable equilibria. *The Quarterly Journal of Economics*, 102(2):179–221, 1987. 29

[64] Arka Rai Choudhuri, Pavel Hubáček, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. Finding a nash equilibrium is no easier than breaking fiat-shamir. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 1103–1114, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi: 10.1145/3313276.3316400. URL https://doi.org/10.1145/3313276.3316400. 30

[65] Timothy Y Chow. The surprise examination or unexpected hanging paradox. *The American Mathematical Monthly*, 105(1):41–51, 1998. 36

[66] Hao Chung and Elaine Shi. Foundations of transaction fee mechanism design. 2021. doi: 10.48550/ARXIV.2111.03151. URL https://arxiv.org/abs/2111.03151. 122, 124

[67] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11 (1):17–33, 1971. doi: 10.1007/BF01726210. URL https://doi.org/10.1007/BF01726210. 76, 122

[68] CoinMarketCap. Cryptocurrency prices, charts and market capitalizations | coinmarketcap. Accessed: 28 June 2023, 2023. URL `https://coinmarketcap.com/`. 12

[69] Vincent Conitzer and Tuomas Sandholm. Complexity results about nash equilibria. *arXiv preprint cs/0205074*, 2002. 30

[70] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 145–152, 2005. 7, 8, 57

[71] Antoine-Augustin Cournot. *Recherches sur les principes mathématiques de la théorie des richesses par Augustin Cournot*. chez L. Hachette, 1838. 29

[72] Nicolas T Courtois. On the longest chain rule and programmed self-destruction of crypto currencies. *arXiv preprint arXiv:1405.0534*, 2014. 12

[73] Geoffroy Couteau and Dennis Hofheinz. Designated-verifier pseudorandom generators, and their applications. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part II 38*, pages 562–592. Springer, 2019. 108

[74] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. doi: 10.1017/CBO9781107337756. 16, 74, 88

[75] Jakša Cvitanić, Dražen Prelec, Blake Riley, and Benjamin Tereick. Honesty via choice-matching. *American Economic Review: Insights*, 1(2):179–92, September 2019. doi: 10.1257/aeri.20180227. URL `https://www.aeaweb.org/articles?id=10.1257/aeri.20180227`. 76

[76] Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. Balancing privacy and accountability in blockchain identity management. In Kenneth G. Paterson, editor, *Topics in Cryptology – CT-RSA 2021*, pages 552–576, Cham, 2021. Springer International Publishing. ISBN 978-3-030-75539-3. 4, 14, 82

[77] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School, Aarhus, Denmark, July 1998*, page 63–86, Berlin, Heidelberg, 1999. Springer-Verlag. ISBN 3540657576. 52

[78] Ivan Bjerre Damgård, Boyang Li, and Nikolaj Ignatieff Schwartzbach. More Communication Lower Bounds for Information-Theoretic MPC. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-197-9. doi: 10.4230/LIPIcs.ITC.2021.2. 22

[79] Ivan Damgård, Hans Gersbash, Ueli Maurer, Jesper Buus Nielsen, Claudio Orlandi, and Torbern Pryds Pedersen. Concordium White Paper, vol. 1.0. Technical report, 04 2020. 4, 14

[80] George Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963. ISBN 9781400884179. doi: doi:10.1515/9781400884179. URL `https://doi.org/10.1515/9781400884179`. 32

[81] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. The complexity of computing a nash equilibrium. *SIAM J. Comput.*, 39: 195–259, 02 2009. doi: 10.1137/070699652. 30, 75, 97

[82] George Louis Leclerc de Buffon. Essai d'arithmétique morale. *Euvres philosophiques*, 1777. 28

[83] M.J.A.N.C. de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. De l'Imprimerie royale, 1785. 55

[84] Alfredo De Santis and Giuseppe Persiano. Zero-knowledge proofs of knowledge without interaction. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 427–436. IEEE Computer Society, 1992. 14

[85] Chrysanthos Dellarocas. Reputation mechanisms. In *Handbook on Economics and Information Systems*, page 2006. Elsevier Publishing, 2006. 6

[86] Roger Dingledine, Nick Mathewson, Paul F Syverson, et al. Tor: The second-generation onion router. In *USENIX security symposium*, volume 4, pages 303–320, 2004. 4

[87] Thomas Dinsdale-Young, Bernardo Magri, Christian Matt, Jesper Buus Nielsen, and Daniel Tschudi. Afgjort: A partially synchronous finality layer for blockchains. In *International Conference on Security and Cryptography for Networks*, pages 24–44. Springer, 2020. 11

[88] Hien Thanh Doan, Jeongho Cho, and Daehee Kim. Peer-to-peer energy trading in smart grid through blockchain: A double auction-based game theoretic approach. *Ieee Access*, 9:49206–49218, 2021. 17

[89] Diana S Dolliver. Evaluating drug trafficking on the tor network: Silk road 2, the sequel. *International Journal of Drug Policy*, 26(11):1113–1123, 2015. 5

[90] Diana S. Dolliver. Evaluating drug trafficking on the tor network: Silk road 2, the sequel. *The International journal on drug policy*, 26 11: 1113–23, 2015. 40

[91] Changyu Dong, Yilei Wang, Amjad Aldweesh, Patrick McCorry, and Aad van Moorsel. Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 211–227, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134032. URL `https://doi.org/10.1145/3133956.3134032`. 77

[92] Anil Donmez and Alexander Karaivanov. Transaction fee economics in the ethereum blockchain. *Economic Inquiry*, 60(1):265–292, 2022. 124

[93] Quinn Dupont. *Experiments in Algorithmic Governance: A history and ethnography of " The DAO, " a failed Decentralized Autonomous Organization*. 01 2017. 13, 55

[94] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*, pages 139–147. Springer, 1992. 11, 12

[95] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 585–605. Springer, 2015. 12

[96] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 967–984, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356930. 6, 41, 45, 81

[97] Lisa Eckey, Sebastian Faust, and Benjamin Schlosser. Optiswap: Fast optimistic fair exchange. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 543–557, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367509. doi: 10.1145/3320269.3384749. URL `https://doi.org/10.1145/3320269.3384749`. 6

[98] Francis Ysidro Edgeworth. The hedonical calculus. *Mind*, 4(15):394–408, 1879. 25

[99] Daniel Ellsberg. Risk, ambiguity, and the savage axioms. *The Quarterly Journal of Economics*, 75(4):643–669, 1961. ISSN 00335533, 15314650. URL `http://www.jstor.org/stable/1884324`. 29

[100] Romain Espinosa. Scamming and the reputation of drug dealers on darknet markets. *International Journal of Industrial Organization*, 67: 102523, 2019. 13

[101] Boi Faltings and Goran Radanovic. *Game Theory for Data Science: Eliciting Truthful Information*. Morgan & Claypool Publishers, 2017. 7, 57

[102] Sebastian Faust, Carmit Hazay, David Kretzler, and Benjamin Schlosser. Financially backed covert security. In *Public-Key Cryptography – PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part II*, page 99–129, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-030-97130-4. doi: 10.1007/978-3-030-97131-1_4. URL `https://doi.org/10.1007/978-3-030-97131-1_4`. 77, 80, 88

[103] Prastudy Fauzi, Sarah Meiklejohn, Rebekah Mercer, and Claudio Orlandi. Quisquis: A new design for anonymous cryptocurrencies. In *Advances in Cryptology–ASIACRYPT 2019: 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8–12, 2019, Proceedings, Part I 25*, pages 649–678. Springer, 2019. 14

[104] Alexis FONTAINE. Solution d'un problème sur les jeux de hasard. *Histoire de l'Académie royale des sciences, Paris pour 1764, avec les Mémoires pour la même année*, pages 429–431, 1764. 28

[105] Konstantinos Fouskas, Olga Pachni-Tsitiridou, and Chrysa Chatzicharistou. A systematic literature review on e-commerce success factors. *Strategic Innovative Marketing and Tourism: 8th ICSIMAT, Northern Aegean, Greece, 2019*, pages 687–694, 2020. 3

[106] Don Fullerton and Ann Wolverton. Two generalizations of a deposit-refund systems. *American Economic Review*, 90(2):238–242, 2000. 77

[107] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. ISBN 978-3-662-46803-6. 12

[108] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990. ISBN 0716710455. 103

[109] Marilyn George and Seny Kamara. Adversarial level agreements for two-party protocols. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, page 816–830, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391405. doi: 10.1145/3488932.3517385. URL https://doi.org/10.1145/3488932.3517385. 15, 75, 77, 80

[110] Nicholas Georgescu-Roegen. Revisiting marshall's constancy of marginal utility of money. *Southern Economic Journal*, pages 176–181, 1968. 26

[111] Dino Gerardi and Leeat Yariv. Information acquisition in committees. *Games and Economic Behavior*, 62(2):436–459, 2008. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2007.06.007. URL https://www.sciencedirect.com/science/article/pii/S0899825607001200. 7

[112] Hans Gersbach. Information efficiency and majority decisions. *Social Choice and Welfare*, 12(4):363–370, 1995. ISSN 01761714, 1432217X. URL http://www.jstor.org/stable/41106142. 7

[113] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973. 57

[114] Itzhak Gilboa and David Schmeidler. Maxmin expected utility with non-unique prior. In *Uncertainty in economic theory*, pages 141–151. Routledge, 2004. 26

[115] Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989. 30

[116] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437. 7, 76

[117] Naman Goel, Cyril van Schreven, Aris Filos-Ratsikas, and Boi Faltings. Infochain: A decentralized, trustless and transparent oracle on blockchain. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4604–4610. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/635. URL https://doi.org/10.24963/ijcai.2020/635. Special Track on AI in FinTech. 7

[118] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery. ISBN 0897911512. doi: 10.1145/22145.22178. URL `https://doi.org/10.1145/22145.22178`. 14

[119] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996. ISBN 0801854148. 92

[120] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Pure nash equilibria: Hard and easy games. *J. Artif. Int. Res.*, 24(1):357–406, sep 2005. ISSN 1076-9757. 30

[121] Raymond Greenlaw, H James Hoover, and Walter L Ruzzo. *Limits to parallel computation: P-completeness theory*. Oxford University Press, USA, 1995. 15

[122] Hilary Grimes-Casey, Thomas Seager, Thomas Theis, and Susan Powers. A game theory framework for cooperative management of the bottle life cycle. *Journal of Cleaner Production*, 15:1618–1627, 11 2007. doi: 10.1016/j.jclepro.2006.08.007. 77

[123] Noortje Groot, Bart De Schutter, and Hans Hellendoorn. Reverse stackelberg games, part ii: Results and open issues. In *2012 IEEE International Conference on Control Applications*, pages 427–432. IEEE, 2012. 18

[124] Noortje Groot, Bart De Schutter, and Hans Hellendoorn. Toward system-optimal routing in traffic networks: A reverse stackelberg game approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):29–40, 2014. 18

[125] Noortje Groot, Georges Zaccour, and Bart De Schutter. Hierarchical game theory for system-optimal control: Applications of reverse stackelberg games in regulating marketing channels and traffic routing. *IEEE Control Systems Magazine*, 37(2):129–152, 2017. 18

[126] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*, pages 305–326. Springer, 2016. 14

[127] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/1914085`. 76, 122

[128] Mathias Hall-Andersen. Fastswap: Concretely efficient contingent payments for complex predicates. Cryptology ePrint Archive, Paper 2019/1296, 2019. URL https://eprint.iacr.org/2019/1296. https://eprint.iacr.org/2019/1296. 6

[129] Mathias Hall-Andersen and Nikolaj Ignatieff Schwartzbach. Game Theory on the Blockchain: A Model for Games with Smart Contracts. In Ioannis Caragiannis and Kristoffer Arnsfelt Hansen, editors, *Algorithmic Game Theory*, pages 156–170, Cham, 2021. Springer International Publishing. ISBN 978-3-030-85947-3. 22, 23, 24, 96, 100

[130] Joseph Halpern and Vanessa Teague. Rational secret sharing and multiparty computation: Extended abstract. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 623–632, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. 17, 37, 73, 90

[131] Joseph Y. Halpern, Rafael Pass, and Lior Seeman. Computational extensive-form games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, page 681–698, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450339360. doi: 10.1145/2940716.2940733. URL https://doi.org/10.1145/2940716.2940733. 30

[132] Mikkel Alexander Harlev, Haohua Sun Yin, Klaus Christian Langenheldt, Raghava Mukkamala, and Ravi Vatrapu. Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. 2018. 13

[133] Justus Haucap and Ulrich Heimeshoff. Google, facebook, amazon, ebay: Is the internet driving competition or market monopolization? *International Economics and Economic Policy*, 11(1-2):49–61, 2014. 39

[134] Justus Haucap and Ulrich Heimeshoff. Google, facebook, amazon, ebay: Is the internet driving competition or market monopolization? *International Economics and Economic Policy*, 11:49–61, 2014. 4

[135] Y.C. Ho and G.J. Olsder. Aspects of the stackelberg problem — incentive, bluff, and hierarchy1. *IFAC Proceedings Volumes*, 14(2):1359–1363, 1981. ISSN 1474-6670. 8th IFAC World Congress on Control Science and Technology for the Progress of Society, Kyoto, Japan, 24-28 August 1981. 18, 95, 99

[136] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 63

[137] Wassily Hoeffding. *Probability Inequalities for sums of Bounded Random Variables*, pages 409–426. Springer New York, New York, NY, 1994. ISBN 978-1-4612-0865-5. doi: 10.1007/978-1-4612-0865-5_26. URL https://doi.org/10.1007/978-1-4612-0865-5_26. 7, 10

[138] Greg Hou. Cryptocurrency money laundering and exit scams: Cases, regulatory responses and issues. *Understanding Cryptocurrency Fraud*, page 83, 2021. 5

[139] Umar Iqbal, Pouneh Nikkhah Bahrami, Rahmadi Trimananda, Hao Cui, Alexander Gamero-Garrido, Daniel Dubois, David Choffnes, Athina Markopoulou, Franziska Roesner, and Zubair Shafiq. Your echos are heard: Tracking, profiling, and ad targeting in the amazon smart speaker ecosystem. *arXiv preprint arXiv:2204.10920*, 2022. 4

[140] Roslan Ismail and Audun Jøsang. The beta reputation system. In *Bled eConference*, 2002. 7

[141] Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium*, pages 258–272. Springer, 1999. 12

[142] Robert G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32(2):146–164, Jun 1985. ISSN 1436-4646. doi: 10.1007/BF01586088. 103

[143] M Johnson. Principia qualia. *URL https://opentheory. net/2016/11/principia-qualia*, 100:013002, 2016. 26

[144] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013. 28

[145] Daniel Kahneman, Peter P Wakker, and Rakesh Sarin. Back to bentham? explorations of experienced utility. *The quarterly journal of economics*, 112(2):375–406, 1997. 25

[146] Atsushi Kajii and Stephen Morris. The robustness of equilibria to incomplete information. *Econometrica: Journal of the Econometric Society*, pages 1283–1309, 1997. 29

[147] Shizuo Kakutani. A generalization of Brouwer's fixed point theorem. *Duke Mathematical Journal*, 8(3):457 – 459, 1941. doi: 10.1215/S0012-7094-41-00838-4. URL https://doi.org/10.1215/S0012-7094-41-00838-4. 30

[148] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003. 7

[149] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. An empirical analysis of anonymity in zcash. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 463–477, 2018. 13

[150] Debarun Kar, Thanh H Nguyen, Fei Fang, Matthew Brown, Arunesh Sinha, Milind Tambe, and Albert Xin Jiang. Trends and applications in stackelberg security games. *Handbook of dynamic game theory*, pages 1–47, 2017. 17

[151] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing nizks from diffie-hellman assumptions. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part II 38*, pages 622–651. Springer, 2019. 108

[152] Thomas Kerber, Aggelos Kiayias, and Markulf Kohlweiss. Kachina – foundations of private smart contracts. *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, pages 1–16, 2021. 13, 80, 122

[153] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 357–388, Cham, 2017. Springer International Publishing. ISBN 978-3-319-63688-7. 122

[154] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19(1), 2012. 12

[155] Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011. 17

[156] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*, pages 839–858. IEEE, 2016. 13

[157] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2009.04.003. URL `https://www.`

sciencedirect.com/science/article/pii/S1574013709000203. 73, 129

[158] David M Kreps and Robert Wilson. Sequential equilibria. *Econometrica: Journal of the Econometric Society*, pages 863–894, 1982. 29

[159] Saul A Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963. 34

[160] Harold William Kuhn and Albert William Tucker. *Contributions to the Theory of Games*. Number 28. Princeton University Press, 1953. 32, 35

[161] Alptekin Küpçü and Anna Lysyanskaya. Optimistic fair exchange with multiple arbiters. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *Computer Security – ESORICS 2010*, pages 488–507, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15497-3. 81

[162] Alptekin Küpçü and Anna Lysyanskaya. Usable optimistic fair exchange. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010*, pages 252–267, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-11925-5. 81

[163] Wesley Lacson and Beata Jones. The 21st century darknet market: lessons from the fall of silk road. *International Journal of Cyber Criminology*, 10(1):40, 2016. 4

[164] Daji Landis and Nikolaj I. Schwartzbach. Stackelberg Attacks or: How I Learned to Stop Worrying and Trust the Blockchain, 2023. 23, 24, 109

[165] Daji Landis and Nikolaj I. Schwartzbach. Stackelberg Attacks on Auctions and Blockchain Transaction Fee Mechanisms, 2023. To appear in *26th European Conference on Artificial Intelligence (ECAI 2023)*. 22, 23, 24, 109, 121

[166] Ron Lavi, Or Sattath, and Aviv Zohar. Redesigning bitcoin's fee market. *ACM Trans. Econ. Comput.*, 10(1), may 2022. ISSN 2167-8375. doi: 10.1145/3530799. URL https://doi.org/10.1145/3530799. 123

[167] G. Leitmann. On generalized stackelberg strategies. *Journal of Optimization Theory and Applications*, 26(4):637–643, Dec 1978. ISSN 1573-2878. doi: 10.1007/BF00933155. 98

[168] Carlton E Lemke and Joseph T Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964. 30, 32

[169] Clément Lesaege, Federico Ast, and William George. Kleros Short Paper v1.0.7. Technical report, Kleros, 09 2019. 8, 10, 42, 55, 56

[170] Clément Lesaege, William George, and Federico Ast. Kleros Long Paper v2.0.2. Technical report, Kleros, 07 2021. 8, 10, 42, 55, 56, 77

[171] Joshua Letchford. *Computational Aspects of Stackelberg Games*. PhD thesis, Duke University, Durham, NC, USA, 2013. 17, 95

[172] Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, page 83–92, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605588223. doi: 10.1145/1807342.1807354. 17, 30, 97

[173] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pages 919–927, 2015. 12

[174] Paul Leydier. Proof-of-stake rewards and penalties, 2023. URL `https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/`. 12

[175] Sen Li, Wei Zhang, Jianming Lian, and Karanjit Kalsi. On reverse stackelberg game and optimal mean field control for a large population of thermostatically controlled loads. In *2016 American Control Conference (ACC)*, pages 3545–3550. IEEE, 2016. 18

[176] Marvin B. Lieberman and David B. Montgomery. First-mover advantages. *Strategic Management Journal*, 9(S1):41–58. doi: https://doi.org/10.1002/smj.4250090706. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/smj.4250090706`. 17

[177] Baoding Liu. Stackelberg-nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computers & Mathematics with Applications*, 36(7):79–89, 1998. 17, 99

[178] Yang P. Liu and Yiling Chen. Sequential peer prediction: Learning to elicit effort using posted prices. In *AAAI*, 2017. 76

[179] Peter B. Luh, Shi-Chung Chang, and Tsu-Shuan Chang. Brief paper: Solutions and properties of multi-stage stackelberg games. *Automatica*, page 251–256, March 1984. 104

[180] David M. McEvoy. Enforcing compliance with international environmental agreements using a deposit-refund system. *International Environmental Agreements: Politics, Law and Economics*, 13(4):481–496,

2013. doi: 10.1007/s10784-013-9209-2. URL https://doi.org/10.1007/s10784-013-9209-2. 77

[181] Richard D. McKelvey and Thomas R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, July 1995. doi: 10.1006/game.1995.1023. URL https://doi.org/10.1006/game.1995.1023. 29

[182] Richard D. Mckelvey and Thomas R. Palfrey. *Experimental Economics*, 1 (1):9–41, 1998. doi: 10.1023/a:1009905800005. URL https://doi.org/10.1023/a:1009905800005. 29

[183] Andrew McLennan. The expected number of nash equilibria of a normal form game. *Econometrica*, 73(1):141–174, 2005. doi: https://doi.org/10.1111/j.1468-0262.2005.00567.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0262.2005.00567.x. 30

[184] Sarah Meiklejohn. The limits of anonymity in bitcoin. In *Routledge Handbook of Crime Science*, pages 280–287. Routledge, 2018. 13

[185] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013. 4, 13

[186] Karl Menger. Das unsicherheitsmoment in der wertlehre: Betrachtungen im anschliß an das sogenannte petersburger spiel. *Zeitschrift für Nationalökonomie/Journal of Economics*, pages 459–485, 1934. 28

[187] Matteo Michelini, Adrian Haret, and Davide Grossi. Group wisdom at a price: Jury theorems with costly information. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 419–425, 2022. 7, 57

[188] John Stuart Mill. Utilitarianism (1863). *Utilitarianism, Liberty, Representative Government*, pages 7–9, 1859. 25

[189] Nolan Miller, Paul Resnick, and Richard Zeckhauser. *Eliciting Informative Feedback: The Peer-Prediction Method*, pages 185–212. Springer London, London, 2009. ISBN 978-1-84800-356-9. doi: 10.1007/978-1-84800-356-9_8. URL https://doi.org/10.1007/978-1-84800-356-9_8. 7

[190] Tal Moran and Ilan Orlov. Simple proofs of space-time and rational proofs of storage. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA,*

August 18–22, 2019, Proceedings, Part I 39, pages 381–409. Springer, 2019. 12

[191] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. doi: 10.1073/pnas.36.1.48. URL https://www.pnas.org/doi/abs/10.1073/pnas.36.1.48. 29, 30

[192] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. 31

[193] Caspar Oesterheld. Robust program equilibrium. *Theory and Decision*, 86 (1):143–159, Feb 2019. ISSN 1573-7187. doi: 10.1007/s11238-018-9679-3. URL https://doi.org/10.1007/s11238-018-9679-3. 18

[194] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, Cambridge, USA, 1994. ISBN 0-262-65040-1. electronic edition. 25, 97

[195] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. ISSN 0022-0000. doi: https://doi.org/10.1016/S0022-0000(05)80063-7. URL https://www.sciencedirect.com/science/article/pii/S0022000005800637. 30

[196] Vilfredo Pareto. *Trattato di sociologia generale...*, volume 2. G. Barbèra, 1916. 26

[197] David G Pearce. Rationalizable strategic behavior and the problem of perfection. *Econometrica: Journal of the Econometric Society*, pages 1029–1050, 1984. 29

[198] Nicola Persico. Committee Design with Endogenous Information. *The Review of Economic Studies*, 71(1):165–191, 01 2004. ISSN 0034-6527. doi: 10.1111/0034-6527.00280. URL https://doi.org/10.1111/0034-6527.00280. 7

[199] Ole Peters. The time resolution of the st petersburg paradox. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1956):4913–4931, 2011. 28

[200] Ole Peters. The ergodicity problem in economics. *Nature Physics*, 15 (12):1216–1221, 2019. 28

[201] Jack Peterson, Joseph Krug, Micah Zoltu, Austin Williams, and Stephanie Alexander. Augur: a decentralized oracle and prediction market platform, 02 2018. 8, 55

[202] Andrew Poelstra, Adam Back, Mark Friedenbach, Gregory Maxwell, and Pieter Wuille. Confidential assets. In *International Conference on Financial Cryptography and Data Security*, pages 43–63. Springer, 2018. 14

[203] Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008. 30, 32

[204] Fedor Poskriakov, Maria Chiriaeva, and Christophe Cavin. Cryptocurrency compliance and risks: A european kyc/aml perspective. *Blockchain & Cryptocurrency Regulation 2020*, 2020. 14

[205] Eric A. Posner and E. Glen Weyl. *Radical Markets: Uprooting Capitalism and Democracy for a Just Society.* Princeton University Press, 2018. 57

[206] Drazen Prelec. A bayesian truth serum for subjective data. *Science (New York, N.Y.)*, 306:462–6, 11 2004. doi: 10.1126/science.1102081. 7, 76

[207] Willy Quach, Ron D Rothblum, and Daniel Wichs. Reusable designated-verifier nizks for all np from cdh. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part II 38*, pages 593–621. Springer, 2019. 108

[208] revoof. 7-piece Syzygy tablebases are complete, 2018. URL `https://lichess.org/blog/W3WeMyQAACQAdfAL/7-piece-syzygy-tablebases-are-complete`. 31

[209] Perri Reynolds and Angela SM Irwin. Tracking digital footprints: anonymity within the bitcoin system. *Journal of Money Laundering Control*, 20(2):172–189, 2017. 13

[210] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2): 358–366, 1953. ISSN 00029947. 18

[211] Tim Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 104–113, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133499. doi: 10.1145/380752.380783. URL `https://doi.org/10.1145/380752.380783`. 17

[212] Tim Roughgarden. Transaction fee mechanism design. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 792, New York, NY, USA, 2021. Association for Computing

Machinery. ISBN 9781450385541. doi: 10.1145/3465456.3467591. URL https://doi.org/10.1145/3465456.3467591. 12, 122, 123, 124

[213] Sagnik Saha, Nikolaj Ignatieff Schwartzbach, and Prashant Nalini Vasudevan. The Planted k-SUM Problem: Algorithms, Lower Bounds, Hardness Amplification, and Cryptography, 2023. 23

[214] JOSEPH SANDERS. A norms approach to jury nullification: Interests, values, and scripts. *Law & Policy*, 30(1):12–45, January 2008. doi: 10.1111/j.1467-9930.2008.00268.x. URL https://doi.org/10.1111/j.1467-9930.2008.00268.x. 55

[215] Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 19–36, 2003. 57

[216] Mark. A. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975. 57

[217] Leonard J Savage. *The foundations of statistics*. Courier Corporation, 1972. 26

[218] Rahul Savani and Bernhard Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74(2):397–429, March 2006. doi: 10.1111/j.1468-0262.2006.00667.x. URL https://doi.org/10.1111/j.1468-0262.2006.00667.x. 32

[219] T.C. Schelling. *The Strategy of Conflict*. Harvard University Press, 1960. 8, 42

[220] David Schmeidler. Subjective probability and expected utility without additivity. In *Uncertainty in Economic Theory*, pages 124–140. Routledge, 2004. 26

[221] Nikolaj Ignatieff Schwartzbach. An incentive-compatible smart contract for decentralized commerce, 2020. URL https://arxiv.org/abs/2008.10326. 40

[222] Nikolaj Ignatieff Schwartzbach. An Incentive-Compatible Smart Contract for Decentralized Commerce. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2021. doi: 10.1109/ICBC51069.2021.9461077. 22, 23, 38, 40, 76

[223] Nikolaj Ignatieff Schwartzbach. Payment Schemes from Limited Information with Applications in Distributed Computing. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22,

page 129–149, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391504. doi: 10.1145/3490486.3538342. 22, 23, 24, 38, 74

[224] Reinhard Selten. Axiomatic characterization of the quadratic scoring rule. *Experimental Economics*, 1(1):43–61, Jun 1998. ISSN 1573-6938. doi: 10.1023/A:1009957816843. URL `https://doi.org/10.1023/A:1009957816843`. 76

[225] Reinhard Selten and R Selten Bielefeld. *Reexamination of the perfectness concept for equilibrium points in extensive games*. Springer, 1988. 29, 108

[226] Nihar B. Shah, Dengyong Zhou, and Yuval Peres. Approval voting and incentives in crowdsourcing. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, page 10–19, 2015. 57

[227] Hanif D Sherali. A multiple leader stackelberg model and analysis. *Operations Research*, 32(2):390–404, 1984. 17, 99

[228] Elaine Shi. Analysis of deterministic longest-chain protocols. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 122–12213. IEEE, 2019. 12

[229] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, USA, 2008. ISBN 0521899435. 34

[230] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: Looking beyond a decade of success. IJCAI, 2018. 17

[231] Maciej Skorski. Bernstein-type bounds for beta distribution. 2021. 129

[232] Katerina Stankova. On Stackelberg and Inverse Stackelberg Games & Their Applications in the Optimal Toll Design Problem, the Energy Market Liberalization Problem, and in the Theory of Incentives. Post-Print hal-00391650, HAL, February 2009. 17, 99

[233] Statista. Bitcoin average energy consumption per transaction compared to that of visa as of may 1, 2023, 2023. URL `https://www.statista.com/statistics/881541/bitcoin-energy-consumption-transaction-comparison-visa/`. 12

[234] George J Stigler. The development of utility theory. i. *Journal of political economy*, 58(4):307–327, 1950. 26

[235] George J Stigler. The development of utility theory. ii. *Journal of political economy*, 58(5):373–396, 1950. 26

[236] Di Suo and Fuan Wen. A reversed stackelberg approach to electronic commerce logistics based on supernetwork theory. In *2009 Second International Symposium on Information Science and Engineering*, pages 114–118. IEEE, 2009. 18

[237] Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), September 1997. doi: 10.5210/fm.v2i9.548. URL `https://doi.org/10.5210/fm.v2i9.548`. 13

[238] Ákos Szigeti, Richard Frank, and Tibor Kiss. Trust factors in the social figuration of online drug trafficking: A qualitative content analysis on a darknet market. *Journal of Contemporary Criminal Justice*, 39(2): 167–184, 2023. 13

[239] Jakub Szymanik. Backward induction is ptime-complete. In *Logic, Rationality, and Interaction - 4th International Workshop, LORI 2013, Hangzhou, China, October 9-12, 2013, Proceedings*, volume 8196 of *Lecture Notes in Computer Science*, pages 352–356. Springer, 2013. 18, 97, 103

[240] Mohammad Amin Tajeddini, Hamed Kebriaei, and Luigi Glielmo. Decentralized hierarchical planning of pevs based on mean-field reverse stackelberg game. *IEEE Transactions on Automation Science and Engineering*, 17(4):2014–2024, 2020. 18

[241] Moshe Tennenholtz. Program equilibrium. *Games and Economic Behavior*, 49(2):363–373, 2004. ISSN 0899-8256. doi: https://doi.org/10.1016/j.geb.2004.02.002. URL `https://www.sciencedirect.com/science/article/pii/S0899825604000314`. 18

[242] Ken Thompson. Retrograde analysis of certain endgames. *J. Int. Comput. Games Assoc.*, 9(3):131–139, 1986. 36

[243] B. Tolwinski. Closed-loop stackelberg solution to a multistage linear-quadratic game. *Journal of Optimization Theory and Applications*, 34: 484 – 501, 08 1981. 95

[244] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992. 26, 28

[245] J v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928. 31

[246] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Theory of Cryptography: Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008. Proceedings 5*, pages 1–18. Springer, 2008. 11

[247] Nicolas Van Saberhagen. Cryptonote v 2.0. 2013. 14

[248] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961. ISSN 00221082, 15406261. URL `http://www.jstor.org/stable/2977633`. 76, 122

[249] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947. 31

[250] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior (60th Anniversary Commemorative Edition)*. Princeton university press, 2007. 26, 32

[251] Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Verlag von Julius Springer, 1934. 17, 97

[252] M. Walls. Deposit-refund systems in practice and theory. *Environmental Economics eJournal*, 2011. 77

[253] W.L. Warren. *Henry II*. English monarchs. University of California Press, 1973. ISBN 9780520022829. URL `https://books.google.dk/books?id=C8KrkVOxaTOC`. 55

[254] Jens Witkowski and David C. Parkes. A robust bayesian truth serum for small populations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 1492–1498. AAAI Press, 2012. 76

[255] Jens Witkowski, Sven Seuken, and David C. Parkes. Incentive-compatible escrow mechanisms. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 751–757. AAAI Press, 2011. 41

[256] Jens Witkowski, Rupert Freeman, Jennifer Vaughan, David Pennock, and Andreas Krause. Incentive-compatible forecasting competitions. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 57

[257] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. 4, 13, 122

[258] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014. 13, 95

[259] Shoufang Xu, Changlin Mei, and Yu Miao. Limit theorems for ratios of order statistics from uniform distributions. *Journal of Inequalities and Applications*, 2019(1):303, Nov 2019. ISSN 1029-242X. doi: 10.1186/s13660-019-2256-7. URL `https://doi.org/10.1186/s13660-019-2256-7`. 133

[260] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. IEEE, November 1982. doi: 10.1109/sfcs.1982.38. URL `https://doi.org/10.1109/sfcs.1982.38`. 16

[261] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, 1986. doi: 10.1109/SFCS.1986.25. 16

[262] H. Peyton Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988. 7, 8, 57

[263] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 270–282, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978326. URL `https://doi.org/10.1145/2976749.2978326`. 7

[264] Haoqi Zhang and David Parkes. Value-based policy teaching with active indirect elicitation. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, page 208–214. AAAI Press, 2008. ISBN 9781577353683. 76

[265] Haoqi Zhang, Yiling Chen, and David Parkes. A general approach to environment design with one agent. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, page 2002–2008, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. 76

[266] Shijie Zhang and Jong-Hyouk Lee. Analysis of the main consensus protocols of blockchain. *ICT express*, 6(2):93–97, 2020. 11

[267] Zhili Zhou, Meimin Wang, Ching-Nung Yang, Zhangjie Fu, Xingming Sun, and Q.M. Jonathan Wu. Blockchain-based decentralized reputation system in e-commerce environment. *Future Gener. Comput. Syst.*, 124(C): 155–167, nov 2021. ISSN 0167-739X. doi: 10.1016/j.future.2021.05.035. URL `https://doi.org/10.1016/j.future.2021.05.035`. 7

[268] Ruiyu Zhu, Changchang Ding, and Yan Huang. Efficient publicly verifiable 2pc over a blockchain with applications to financially-secure

computations. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 633–650, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363215. URL https://doi.org/10.1145/3319535.3363215. 77

[269] David Zimbeck. Two Party double deposit trustless escrow in cryptographic networks and Bitcoin. Technical report, BitHalo, 2015. 77