Payment Schemes from Limited Information with Applications in Distributed Computing

NIKOLAJ IGNATIEFF SCHWARTZBACH, Aarhus University

We propose a generic mechanism for incentivizing behavior in an arbitrary finite game using payments. Doing so is trivial if the mechanism is allowed to observe all actions taken in the game, as this allows it to simply punish those agents who deviate from the intended strategy. Instead, we consider an abstraction where the mechanism probabilistically infers information about the outcome of the game. We show that payment schemes can be used to implement any set of utilities if and only if the mechanism can essentially infer completely what happened. We show that finding an optimal payment scheme for games of perfect information is P-complete, and conjecture it to be PPAD-hard for games of imperfect information. We prove a lower bound on the size of the payments, showing that the payments must be linear in the intended level of security. We demonstrate the applicability of our model to concrete problems in distributed computing, namely decentralized commerce and secure multiparty computation, for which the payments match the lower bound asymptotically.

CCS Concepts: • **Computing methodologies** → *Artificial intelligence*; • **Security and privacy** → **Security protocols**; • **Theory of computation** → *Complexity theory and logic*; **Algorithmic mechanism design**.

Additional Key Words and Phrases: payments, mechanism design, smart contracts, decentralized commerce, secure multiparty computation

ACM Reference Format:
Nikolaj I. Schwartzbach. 2022. Payment Schemes from Limited Information with Applications in Distributed Computing. In *Proceedings of the 23rd ACM Conference on Economics and Computation (EC '22), July 11–15, 2022, Boulder, CO, USA.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3490486.3538342

1 INTRODUCTION

Game theory is the study of strategic reasoning in rational agents, where rationality means the agents act to maximize their own utility. When all agents choose such a strategy, we call the resulting interaction an equilibrium. Unfortunately, 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. Here, 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 [27], 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 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 [25], 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 [12]. A weaker notion of security called 'covert security' was proposed by Aumann and Lindell in [6]. Here, agents are allowed to deviate but are caught with some constant non-zero probability. This was extended by Asharov and Orlandi in [4] to publicly verifiable covert (PVC) security where a certificate is output that can be verified by a third party 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 work, we consider using *payments* as a generic way to incentivize the participating agents to behave in a certain way. We ask the following question.

> Can we design a generic mechanism that uses payments to incentivize behavior in an arbitrary game involving rational agents?

Of course, this is trivial if we allow the mechanism to observe all actions taken by the parties, as this allows the mechanism to punish parties who deviate from the inteded strategy. Instead, we consider an abstraction where the mechanism is only allowed to probabilistically infer information about what happened in the game. To the best knowledge of the authors, this question has not been studied in full generality before, in a way that allows for 'plug-and-play' with an arbitrary game with some probabilistic release of information.

1.1 Related Work

1.1.1 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 [21, 36]. 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 [34]. This was extended to 'robust BTS' by Witkowski and Parkes [39] that instead uses the quadratic scoring rule. Scoring rules are also used in peer prediction methods [31], Bayesian markets [7] and choice matchings [13]. Payments are also used more generally in the generic Vickrey-Clarke-Groves (VCG) mechanism for obtaining a socially optimal outcome [11, 24, 37]. However, the VCG mechanism is fundamentally limited to games that involve distributing a set of 'items' among a set of players. It is not obvious how this would apply to an arbitrary extensive-form game. Another relevant line of work is that of environment design [40] 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 [41]) 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 players.

1.1.2 Distributed Computing. There are numerous works in the literature that 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; the protocol starts with each party 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 [20] who propose a framework for incentivizing honesty using 'adversarial level agreements' that specify damages parties must pay if found to act adversarially. We will show later that their model can be recovered as a special case of our model. Recently, Faust, Hazay, Kretzler and Schlosser propose 'financially backed covert security' [18] to punish parties 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 [42], Zhu, Ding and Huang propose a protocol for two-party computation that incentivizes honesty using a publicly verifiably covert secure protocol augmented with a deposit scheme. In [16], Dong et al. propose a protocol that uses deposit schemes to incentivize honesty in outsourcing cloud computations. BitHalo [43] implements an escrow using deposits and multisigs that was analyzed in [8] by Bigi et al. In [3], Asgaonkar and Krishnamachari propose a smart contract for decentralized commerce of digital goods using dual deposits. This was extended by Schwartzbach to non-digital goods in [35] in a way that uses deposits optimistically. Deposits have also been used for 'truth-telling mechanisms': in [2], Adler et al. propose a system, Astraea, that uses deposits and rewards to incentivize a group of voters to decide to validity of a proposition. Kleros [30] uses a similar mechanism to implement a decentralized court system.

1.1.3 Economics. In the economics literature, the payment schemes that we study are known as 'deposit-refund systems' [19]. They are often studied in the context of environmental issues for incentivizing compliance with laws and regulations. In [23], 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. [38] for an overview. Such systems can also be

used at a higher level of governance: in [32], McEvoy studies deposit-refund systems as a means of enforcing nations to comply with international environmental agreements.

1.2 Our Contributions

We propose a generic mechanism to incentivize behavior in an arbitrary n-player finite games with the use of payments. This is a trivial task if the mechanism can observe all actions taken in the game; instead, our model assumes the mechanism probabilistically observes actions taken by the agents. 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 1). We show that our model generalizes similar models in the literature, such as 'adversarial level agreements' by George and Kamara [20] 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 payments for decentralized commerce to solve the 'eBay problem', such that neither buyer nor seller has an incentive to cheat.

We investigate the computational complexity of computing an optimal deposit 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 1 (INFORMAL). *Finding an optimal deposit scheme for a finite game of perfect information, or showing no suitable deposit scheme exists, is P-complete.*

For games of imperfect information, it is well-known that even computing an equilibrium is PPAD-complete [15], so it is unlikely there is an efficient algorithm for finding an optimal deposit scheme in these cases. As a consequence, we conjecture that finding an optimal deposit 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 multiparty computation. We show that payments can be used, together with what is known as an 'ε-deterrent publicly verifiable covert (PVC) secure protocol' [4, 6], 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 quasilinear, 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 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 party 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 2 since n, s, and ε are constant for any fixed PVC protocol.

THEOREM 3 (INFORMAL). *Any budget balanced payment scheme that achieves δ-strong game-theoretic security in a game of n players must have a largest punishment that is no smaller than $\Omega(1 + \delta\sqrt{n}/s)$, where s is the number of observable outcomes.*

The paper is organized as follows. We start in Section 3 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 4, we consider the computational complexity

of finding payment schemes and prove Theorem 1. Next, in Section 5, we apply the framework to secure MPC and prove Theorem 2. Finally, in Section 6, we show a lower bound on the size of the maximum deposits and prove Theorem 3.

2 PRELIMINARIES AND NOTATION

In this section we briefly state some preliminaries needed for the purpose of self-containment, as well as to establish notation. We use a boldface font to refer to vectors and matrices, and reserve capital symbols $\mathbf{A}$ for matrices, and lowercase symbols $\mathbf{u}$ for vectors. The $n \times n$ identity matrix is denoted $\mathbf{I}_n$. We may denote by $\mathbf{0}$, resp. $\mathbf{1}$ as either matrices or vectors containing only 0, resp. 1 and trust it is clear from the context what we mean. To emphasize the size, we may write e.g. $\mathbf{0}^n$ as the vector $[0, 0, \ldots, 0]^\top \in \mathbb{R}^n$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$, we denote by $\mathrm{vec}(\mathbf{A}) = [a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{mn}]^\top \in \mathbb{R}^{mn}$ the vectorized version of $\mathbf{A}$, resulting from 'flattening' the matrix to turn it into a vector in row-major order. Note that this is a linear operation. If $\mathbf{a} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^{n'}$ are two vectors, we denote by $\mathbf{a} \| \mathbf{b} \in \mathbb{R}^{n+n'}$ the 'concatenation' of $\mathbf{a}$ and $\mathbf{b}$.

2.1 Game Theory

We mostly assume familiarity with game theory and refer to [33] for more details. We give a brief recap to establish notation. An *extensive-form game* consists of a rooted tree T, the leaves of which are labeled with a utility for each player. We denote by $L \subseteq T$ the set of leaves in T, and suppose some arbitrary but fixed order on its elements, $\ell_1, \ell_2, \ldots \ell_m$. We assume the existence of an $n \times m$ matrix $\mathbf{U} = (u_{ij}) \in \mathbb{R}^{n \times m}$, called the utility matrix of G, that for each player P_i specifies how much utility u_{ij} they receive when the game terminates in the leaf $\ell_j \in L$. The remaining nodes $T \setminus L$ are partitioned into n sets, one belonging to each player. The game is played, starting at the root, by recursively letting the player who owns the current node choose a child to descend into. We stop when a leaf ℓ_j is reached, after which player P_i is given u_{ij} utility. A mapping s_i that dictates the moves a player P_i makes is called a strategy for that player, 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 player, 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 player P_i when playing the strategy profile s. If $C \subseteq \{1, 2, \ldots n\}$ is a set of indices of players, a coalition, we denote by $-C$ its complement so that we may write a strategy profile s as $s = (s_C, s_{-C})$. As solution concept, we will use a refinement of regular Nash equilibria that takes into account deviations by more than a single party, see [1] for details on this model. Formally, a strategy profile s^* is said to be a *t-robust (Nash) equilibrium* if for every strategy s_C with $|C| \leq t$, and every $i \in C$, it holds that:

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

A subgame of G is a subtree $G' \subseteq G$ such that whenever $u \in G'$ and $v \in G$ is a child of u, then $v \in G'$. A strategy profile that is a t-robust equilibrium for every subgame of G is said to be a *t-robust subgame perfect equilibrium* (SPE). These definitions suffice for so-called games of perfect information, where at each step, a player knows the actions taken by previous players, though, more generally, we may consider partitioning each set of nodes belonging to a player into *information sets*, the elements of which are sets of nodes that the player cannot tell apart. A game of perfect information is a special case where all information sets are singletons.

3 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 games of perfect information as every such game allows for backward induction to efficiently determine an SPE. In general,

we should not hope to efficiently determine an optimal payment scheme for games of imperfect information, as it is well-known that finding an equilibrium in these games is PPAD-complete as famously shown by Daskalakis, Goldberg and Papadimitriou [15].

We consider a set of n parties $P_1, P_2, \ldots, P_n$ playing a fixed finite extensive-form game G of perfect information. The parties are assumed to be risk-neutral such that they aim to maximize their expected utility. We assume the parties have quasilinear 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 parties 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 a player plays a mixed strategy. This has the effect that s^* defines, at each branch in the game, a unique 'honest move' that the corresponding party 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. In order 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 parties 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 1 (INFORMATION STRUCTURE). *An* information structure *for G is a pair $\langle \Sigma, \Phi \rangle$ 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 $\Phi = (\phi_{kj}) \in \mathbb{R}^{s \times m}$ is a matrix of emissions probabilities such that every column of Φ 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 parties based on what was emitted by the information structure. We assume *quasilinear* utility functions, i.e. if for a player 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 is able to process. A party 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, quasilinearity allows the payment schemes to alter the order by punishing or rewarding parties with money.

DEFINITION 2 (PAYMENT SCHEME). *A* payment scheme *for $\langle G, \mathcal{I} \rangle$ is a matrix $\Lambda = \{\lambda_{ik}\} \in \mathbb{R}^{n \times s}$, where $\mathcal{I} = \langle \Sigma, \Phi \rangle$ is an information structure for G, and λ_{ik} is the utility lost by P_i when observing the symbol $\sigma_k \in \Sigma$.*

Λ is a matrix that explicitly defines how much utility λ_{ik} party P_i loses when the payment scheme observes the symbol $\sigma_k \in \Sigma$. Note that Λ is allowed to contain negative entries which means parties receive back more funds from the payment scheme than they initially deposited. When the game is played reaching the leaf ℓ_j, by quasilinearity the expected utility of party P_i is the utility they

would have received in a normal execution, minus their expected loss from engaging with the payment scheme:

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

Correspondingly, we say $\mathbf{\Lambda}$ *implement 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 1. $\mathcal{S}(\mathbf{E})$ *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 $\mathbf{U} - \mathbf{E} = \mathbf{\Lambda_E}\mathbf{\Phi}$. Assume for the sake of contradiction that there are fewer symbols than leaves. Thus, there is a leaf for which the payments 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 $\mathbf{F}\mathbf{\Lambda_E}\mathbf{\Phi} = \mathbf{I}_m$, 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 1. *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 $\mathbf{1}^{\top}(\mathbf{U} - \mathbf{E}) = \mathbf{1}^{\top}\mathbf{\Lambda\Phi} = (\mathbf{1}^{\top}\mathbf{\Lambda})\mathbf{\Phi} = \mathbf{0}^{\top}\mathbf{\Phi} = \mathbf{0}$. □

3.1 A Special Case: Adversarial Level Agreements

We now show how the model of 'adversarial level agreements' (ALAs) by George and Kamara [20] can be recovered as a special case of our model. An ALA for a game with n players consists of 1) a description of the intended strategy for each player, and 2) a vector of damages $\mathbf{d} \in \mathbb{R}^n$ that specifies how much utility $\mathbf{d}_i$ party P_i should lose when found to deviate from the intended strategy. Their model does not explicitly consider deviations by more than a single party, 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 parties 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 $t \le n$ parties by including more symbols, e.g. $\perp_{12}$ or $\perp_{456}$.

3.2 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. [18, 26] for more formal cryptographic modeling. At a high level, we want to ensure party P_i loses λ_{ik} utility when the symbol σ_k is observed. We can implement this by defining $\lambda_i^* := \max_{k \in \{1,2,\dots s\}} \lambda_{ik}$ and letting each party P_i make a deposit of size λ_i^* to a smart contract before playing the game. Afterwards, the parties 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 party 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 parties are anonymous. The important thing is that party P_i loses λ_{ik} utility when symbol σ_k is observed. When parties 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.

3.3 Game-Theoretic Security

We now define what it means for a game to be secure in a game-theoretic sense. Intuitively, 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 parties 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 parties lose by deviating from the honest strategy profile, in effect measuring the cost of dishonesty. We introduce a parameter $\delta \ge 0$ such that being dishonest results in the deviating parties losing at least δ utility. A game with this property is considered secure against δ-deviating rational parties. We give the definition by Schwartzbach [35] that generalizes t-robust subgame perfect equilibria for finite games of perfect information. Let G be a fixed finite game with n players and m leaves, and let $\mathbf{U} \in \mathbb{R}^{n \times m}$ be the corresponding utility matrix, and let $\langle \Sigma, \Phi \rangle$ be some fixed information structure on G. 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 labelled by $\mathbf{u}$ with non-zero probability.

DEFINITION 3 (GAME-THEORETIC SECURITY). *Let G be a game, and s^* an intended strategy profile. We say G has δ-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:*

$$\mathbf{u}_i^* \geq \mathbf{u}_i + \delta \tag{2}$$

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

3.4 Example: Decentralized Commerce

In this section, we demonstrate the applicability of our model by considering a simple model of problem of *decentralized commerce* (the 'eBay problem'), studied in further generality by Schwartzbach.

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. [3, 5, 17, 28, 29] 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. 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, 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 [14].

To do so, let us first assume the parties use a smart contract to process the trade, and let us assume we have some means of probabilistically inferring whether the item was actually shipped. In the blockchain literature, such a mechanism is called a *blockchain oracle* [9]. In particular, we assume the oracle can recover the ground truth with probability $1 - \varepsilon$ for some constant $\varepsilon < \frac{1}{2}$. One proposed blockchain oracle is the jury-based system Kleros [30], a simplified version of which was shown in [10] to yield $\varepsilon < \frac{1}{2}$ under reasonable assumptions on the jurors. To proceed, we need to define an information structure on the game. We first define an alphabet of outcomes as $\Sigma = \{\top, \perp_B, \perp_S\}$. Here $\top$ is a symbol emitted if the buyer accepts the trade, and $\perp_B, \perp_S$ are outcomes of the oracle if it is invoked, where $\perp_B$ (resp. $\perp_S$) means 'the buyer (resp. the seller) was dishonest'. We define the emissions matrix of the game as follows.

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 - \varepsilon & \varepsilon & 0 \\ 0 & \varepsilon & 1 - \varepsilon & 0 \end{pmatrix}$$
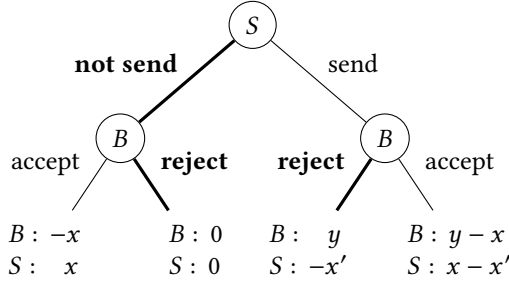
Fig. 1. Extensive-form representation of the decentralized commerce game. The dominating paths are shown in bold. We observe that the dominating strategy is for the seller not to send the item, and for the buyer to withhold their payment regardless of whether they received the item.

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. 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. (1) as follows.

$$\mathbf{\Lambda\Phi} = \mathbf{U} - \mathbf{E}$$

$$\iff \begin{pmatrix} \lambda_{B\top} & (1-\varepsilon)\lambda_{B\perp_B} + \varepsilon\lambda_{B\perp_S} & \varepsilon\lambda_{B\perp_B} + (1-\varepsilon)\lambda_{B\perp_S} & \lambda_{B\top} \\ \lambda_{S\top} & (1-\varepsilon)\lambda_{S\perp_B} + \varepsilon\lambda_{S\perp_S} & \varepsilon\lambda_{S\perp_B} + (1-\varepsilon)\lambda_{S\perp_S} & \lambda_{S\top} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2x & 0 \\ 0 & x' & 0 & 0 \end{pmatrix}$$

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.

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

In other words, the buyer must make a deposit of size $\frac{2(1-\varepsilon)}{1-2\varepsilon}x$ to the smart contract, while the seller must make a deposit of size $\frac{1-\varepsilon}{1-2\varepsilon}x'$. To make this more concrete, suppose we have $x = 100\$$, $x' = 50\$$, and $\varepsilon = 0.1$. 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 an 'optimistic' variant of this payment scheme proposed by Schwartzbach. Here, both buyer and seller submit a deposit of size $\lambda_B^* = \lambda_S^* = x$, and the resulting contract is shown to have $(1 - 2\varepsilon)x$-strong game-theoretic security. Their contract is optimistic in the sense that deposits are only used when the buyer disputes delivery of the item.

4 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.

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. $\Gamma^{\Lambda}(G)$ has δ-strong, t-robust game-theoretic security, for which $\mathbf{c}^\top \text{vec}(\mathbf{\Lambda})$ 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 1. *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 1. *PaymentScheme$_t^\delta$ is PPAD-hard for imperfect information games.*

4.1 Upper Bound: Reduction to LP

In this section, we show how to reduce 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 2. *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:

$$\mathbf{\Psi}^{(t)} \mathbf{v} \geq \boldsymbol{\delta}^{(t)} \tag{3}$$

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 a player 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 $\langle \Sigma, \mathbf{\Phi} \rangle$, we construct a matrix $\mathbf{R}$ equivalent to $\mathbf{\Phi}$ in the following way: for every index ij in $\mathbf{\Lambda}$, 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 $\text{vec}(\mathbf{L}^{ij}\Phi)$ as the product $\mathbf{L}^{ij}\Phi$ put in row-major order. It is not hard to see that the image of Φ is isomorphic to the column space of $\mathbf{R}$, and hence we say λ implements the utility vector $\mathbf{e}$ iff $\mathbf{e} = \mathbf{u} - \mathbf{R}\lambda$. We now substitute this in Eq. (3) to get $\Psi^{(t)}(\mathbf{u} - \mathbf{R}\lambda) \geq \delta^{(t)}$. Next, we move the constant terms to the right-hand side to yield the following:

$$-\Psi^{(t)}\mathbf{R}\lambda \geq \delta^{(t)} - \Psi^{(t)}\mathbf{u} \tag{4}$$

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 $k = 1 \ldots s$. Finally, note that our objective function is $\mathbf{c}^{\top}\lambda$, and since all constraints are linear we can produce the following linear program.

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

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.

4.2 Lower Bound: Reduction from LP

We now show how to reduce LP to PaymentScheme$_1^0$ using logarithmic space. The resulting game is a two-player 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.

Proposition 3. *LP can be reduced to 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 players P_1, P_2. The root of the game consists of a move for player 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 players. Each gadget consists of a move for the other player 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. 2 for an illustration. The SPE of the game is for P_2 to move left in the ith gadget if $\mathbf{b}_i > 0$, and for P_1 to choose any convex combination of the gadgets for which the players 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 $\Sigma = \{\top, \perp_1, \perp_2, \ldots \perp_n\}$, where $\top$ means 'all inequalities are
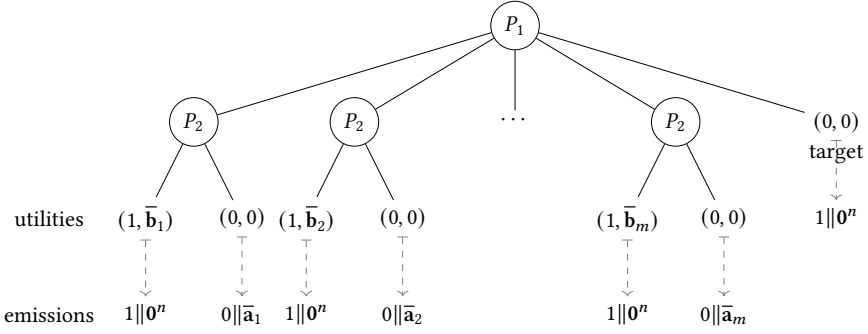
Fig. 2. Depiction of the reduction from LP to PaymentScheme$_1^0$. The dashed arrows depict the corresponding information structure (pdf for each leaf). The player P_1 wants to sabotage satisfiability of the circuit and gains 1 utility for doing so (0 otherwise). The player P_2 will sabotage the ith gadget (and hence allow P_1 to win) if and only if the ith 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.

satisfied', while $\perp_i$ 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 a player goes left, the corresponding pdf is $[1, 0, 0, \ldots, 0]^\top \in \mathbb{R}^{n+1}$, and when a player 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 ith *row* of $\mathbf{\Lambda}$ (as a column vector) corresponding to the leaf where P_i goes right, and let $\Phi_i = (0\|\overline{\mathbf{a}}_i)$ be the column of Φ corresponding to going right in the ith 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. (1), we must have,

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

where $\mathbf{x} := [\Lambda_{i2}]_{i=1}^m$ 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 player 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 Λ 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 CASE STUDY: SECURE RATIONAL MPC FROM PVC

In this section, we apply our framework to a more complicated scenario involving secure multiparty computation (MPC). Our work is similar to [18], 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 [12] for more details and formal definitions.

Secure Multiparty Computation (MPC).. In MPC, a set of n mutually distrusting parties $P_1, P_2, \ldots, P_n$ want to compute a public function f on their private data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The parties 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 parties, other than that which can be gathered from the function output itself. This should hold even if a coalition of t parties 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 [6] 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 parties 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 party 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 [4] where a proof of cheating is output that can be verified by a third party. 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 assumption in many cases, it is unclear that this works in e.g. an anonymous setting where the parties cannot be held accountable. Instead, we will use a payment scheme to prove it is rational for the parties not to cheat. Our plan is to analyze the information
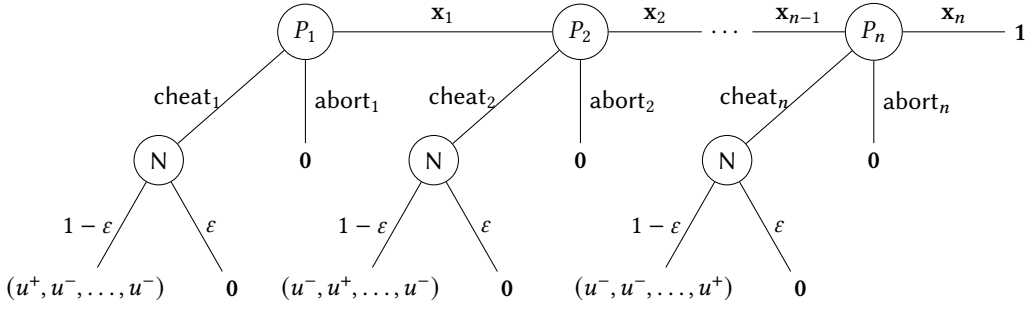
Fig. 3. The ideal functionality $\mathcal{F}_{\mathsf{PVC}}$ with the strong explicit cheat formulation represented as an extensive-form game G_{PVC} rooted at P_1. Our goal is to augment the functionality with a payment scheme such that it is rational to behave honestly. Note that in this representation, for clarity there are two distinct leaves when a player P_i attempts cheating, though in the following we 'merge' the two nodes belonging to nature for simplicity.

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 parties.

Secure Rational MPC from PVC.. We consider a set of n parties $P_1, P_2, \ldots, P_n$ 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 parties. In order to simplify the presentation, we assume the parties 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 parties. Note that we can always normalize the utilities in a game as this preserves the total order. As such, we assume a party receives 1 utility if they send their input and receive back the correct output. If instead a party cheats and is successful, they receive u^+ utility, while a party whose input is revealed receives u^- utility. As $\mathcal{F}_{\mathsf{PVC}}$ does not explicitly punish parties who are caught cheating, we assume a party who is caught cheating receives 0 utility. As in [4], we are using the strong explicit cheat formulation from [6], so a cheater who is caught does not learn the inputs of the honest parties, and as such earns 0 utility. We are not modeling the fact that parties can send incorrect inputs, for the simple reason that it is impossible for the payment scheme, in general, to detect this. We assume that parties 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 a party 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 parties are oblivious to the utility earned by other parties. This is in contrast to [25] who assume parties strictly prefer that as few other parties learn the output as possible. This is not to circumvent their impossibility result, as this is accomplished by quasilinearity 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 a party loses by other parties 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. 3. Observe that when $(1 - \varepsilon)\, u^+ > 1$, the only equilibrium in the game is for P_1 to cheat. Instead, we want all players 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 party can deviate in any given strategy profile. This means we can define the following alphabet of possible outcomes as, $\Sigma = \{\top, \mathrm{abort}_1, \mathrm{cheat}_1, \mathrm{abort}_2, \mathrm{cheat}_2, \ldots, \mathrm{abort}_n, \mathrm{cheat}_n\}$. 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. 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 1 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 parties are honest, we reach the outcome $\mathbf{1}$ and the symbol $\top$ is emitted. If some party P_i aborts, the output of the honest parties will always be abort_i. If instead, a party 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. 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.

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

By Lemma 1, 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 party 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 party as it is not possible for multiple parties 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 parties. In order to compute the deposits, we

again apply Lemma 1 and compute the appropriate payments:

$$\Lambda_{\text{PVC}} = (\mathbf{U} - \mathbf{E})\, \Phi_{\text{PVC}}^{-1} = \begin{pmatrix} 0 & \frac{u^+ + \delta}{\varepsilon} & \delta & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^+ + \delta}{\varepsilon} & \delta & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^-}{\varepsilon} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & \frac{u^-}{\varepsilon} & 0 & \frac{u^-}{\varepsilon} & 0 & \cdots & \frac{u^+ + \delta}{\varepsilon} & \delta \end{pmatrix}$$

We now briefly analyze the resulting payment scheme. We note that when $\top$ is emitted, all parties are repaid their deposits in full. When the symbol abort$_i$ is emitted, the party P_i loses part of their deposit, while all other parties are repaid their deposit in full. Finally, when cheat$_i$ is emitted, the party 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 parties, 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 party P_i when successful in cheating. This shows the following result.

THEOREM 2 (RATIONAL MPC). *Let f be a public function and let $P_1, P_2, \ldots, P_n$ be a set of rational parties 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 parties, while they gain u_i^- utility from another party 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).

6 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 [22] 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 $\|\mathbf{A}\|_p = \sup_{\mathbf{x} \neq 0, \|\mathbf{x}\|_2 = 1} \|\mathbf{A}\mathbf{x}\|_p$. If in addition, $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$, 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 $\alpha \|\mathbf{A}\|_a \le \|\mathbf{A}\|_b \le \beta \|\mathbf{A}\|_a$, for some constants $\alpha, \beta \in \mathbb{R}$. For our purposes, we need the following bounds:

$$\frac{\|\mathbf{A}\|_2}{\sqrt{mn}} \le \|\mathbf{A}\|_{\max} \le \|\mathbf{A}\|_2 \tag{5}$$

$$\frac{1}{\sqrt{m}} \|\mathbf{A}\|_1 \le \|\mathbf{A}\|_2 \le \sqrt{n} \|\mathbf{A}\|_1 \tag{6}$$

$$\frac{1}{\sqrt{n}} \|\mathbf{A}\|_\infty \le \|\mathbf{A}\|_2 \le \sqrt{m} \|\mathbf{A}\|_\infty \tag{7}$$

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

$$\Psi^{(t)} \boldsymbol{\Lambda} \Phi \le \Psi^{(t)} \mathbf{U} - \boldsymbol{\delta}^{(t)} \tag{8}$$

Applying Eqs. (5) and (8) and the properties of $\|\cdot\|_2$, we establish the following bound:

$$\|\boldsymbol{\Lambda}\|_{\max} \ge \frac{1}{\sqrt{n|\Sigma|}} \left(\frac{\left\|\Psi^{(t)}\mathbf{U}\right\|_2 + \left\|\boldsymbol{\delta}^{(t)}\right\|_2}{\left\|\Psi^{(t)}\right\|_2 \cdot \|\Phi\|_2} \right) \tag{9}$$

Each row of $\boldsymbol{\delta}^{(t)}$ is filled with δ, so the resulting absolute row sum is δn. Similarly, each row of $\Psi^{(t)}$ contains exactly one 1 and one -1, so each absolute row sum is 2. Finally, each column of Φ is a pdf, so its absolute row sum is 1. Combining these insights with Eqs. (6) and (7) and substituting in Eq. (9) gives the following bound:

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

We note that in general, there is not much to say about $\left\|\Psi^{(t)}\mathbf{U}\right\|_2$, as $\mathbf{U}$ can lie in the kernel of $\Psi^{(t)}$. 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, $\|\boldsymbol{\Lambda}\|_{\max} \ne \boldsymbol{\Lambda}^*_{\max}$ iff we can pay more to a party to misbehave *and still retain security* than what we have to pay another party 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 $\Delta_G^{(t)}(s^*)$ the minmax deposit required to obtain 0-strong t-robust game-theoretic security. We note that $\Delta_G^{(t)} > 0$ iff the game is not secure for any $\delta \ge 0$, while $\Delta_G^{(t)} \le 0$ iff the game is already secure for $\delta = 0$. We note that by definition, $\Delta_G^{(t)}$ 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 3. *Let G be a game on n players with an information structure $\langle \Sigma, \Phi \rangle$, and let s^* be the intended strategy profile. If Λ is budget balanced and ensures δ-strong t-robust game-theoretic security, then the maximum deposit must satisfy $\Lambda^*_{max} \geq \Delta_G^{(t)}(s^*) + \Omega\left(\frac{\delta\sqrt{n}}{|\Sigma|}\right)$.*

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a generic mechanism for incentivizing behavior in games by the use of payments. We analyzed the complexity of finding an optimal payment scheme and found it to be equivalent to linear programming. We demonstrated the applicability of our framework to concrete problems in distributed computing, namely decentralized commerce and secure multiparty computation. Finally, we proved a lower bound on the payments, showing that the largest payment must be linear in the security parameter for any budget balanced payment scheme.

We hope that our framework will find applications in distributed computing, giving a simple, yet expressive model for instantiating payments in a variety of interactive protocols. However, more attention is needed for games of imperfect information; we conjecture it is unlikely there is a generic and efficient way to find payments for such games, though a formal reduction is needed.

8 ACKNOWLEDGEMENTS

REFERENCES

[1] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. 2006. 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* (Denver, Colorado, USA) *(PODC '06)*. Association for Computing Machinery, New York, NY, USA, 53–62.

[2] John Adler, Ryan Berryhill, Andreas Veneris, Zissis Poulos, Neil Veira, and Anastasia Kastania. 2018. 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)*. 1145–1152. https://doi.org/10.1109/Cybermatics_2018.2018.00207

[3] Aditya Asgaonkar and Bhaskar Krishnamachari. 2019. 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)*. 262–267. https://doi.org/10.1109/BLOC.2019.8751482

[4] Gilad Asharov and Claudio Orlandi. 2012. Calling Out Cheaters: Covert Security with Public Verifiability. In *Advances in Cryptology – ASIACRYPT 2012*, Xiaoyun Wang and Kazue Sako (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 681–698.

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

[6] Yonatan Aumann and Yehuda Lindell. 2007. Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries. In *Theory of Cryptography*, Salil P. Vadhan (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 137–156.

[7] Aurélien Baillon. 2017. Bayesian markets to elicit private information. *Proceedings of the National Academy of Sciences* 114, 30 (2017), 7958–7962. https://doi.org/10.1073/pnas.1703486114 arXiv:https://www.pnas.org/content/114/30/7958.full.pdf

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

[9] Giulio Caldarelli and Joshua Ellul. 2021. The Blockchain Oracle Problem in Decentralized Finance—A Multivocal Approach. *Applied Sciences* 11, 16 (2021). https://doi.org/10.3390/app11167572

[10] Ioannis Caragiannis and Nikolaj I. Schwartzbach. 2022. Adjudication with Rational Jurors. arXiv:2201.06597 [cs.GT]

[11] Edward H. Clarke. 1971. Multipart pricing of public goods. *Public Choice* 11, 1 (1971), 17–33. https://doi.org/10.1007/BF01726210

[12] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. 2015. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press. https://doi.org/10.1017/CBO9781107337756

[13] Jakša Cvitanić, Dražen Prelec, Blake Riley, and Benjamin Tereick. 2019. Honesty via Choice-Matching. *American Economic Review: Insights* 1, 2 (September 2019), 179–92. https://doi.org/10.1257/aeri.20180227

[14] Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. 2021. Balancing Privacy and Accountability in Blockchain Identity Management. In *Topics in Cryptology – CT-RSA 2021*, Kenneth G. Paterson (Ed.). Springer International Publishing, Cham, 552–576.

[15] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. 2009. The Complexity of Computing a Nash Equilibrium. *SIAM J. Comput.* 39 (02 2009), 195–259. https://doi.org/10.1137/070699652

[16] Changyu Dong, Yilei Wang, Amjad Aldweesh, Patrick McCorry, and Aad van Moorsel. 2017. 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* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 211–227. https://doi.org/10.1145/3133956.3134032

[17] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. 2018. FairSwap: How To Fairly Exchange Digital Goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 967–984.

[18] Sebastian Faust, Carmit Hazay, David Kretzler, and Benjamin Schlosser. 2022. 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*. Springer-Verlag, Berlin, Heidelberg, 99–129. https://doi.org/10.1007/978-3-030-97131-1_4

[19] Don Fullerton and Ann Wolverton. 2000. Two generalizations of a deposit-refund systems. *American Economic Review* 90, 2 (2000), 238–242.

[20] Marilyn George and Seny Kamara. 2020. Adversarial Level Agreements for Two-Party Protocols. Cryptology ePrint Archive, Report 2020/1249. https://eprint.iacr.org/2020/1249.

[21] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Amer. Statist. Assoc.* 102, 477 (2007), 359–378. https://doi.org/10.1198/016214506000001437 arXiv:https://doi.org/10.1198/016214506000001437

[22] Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA.

[23] Hilary Grimes-Casey, Thomas Seager, Thomas Theis, and Susan Powers. 2007. A game theory framework for cooperative management of the bottle life cycle. *Journal of Cleaner Production* 15 (11 2007), 1618–1627. https://doi.org/10.1016/j.jclepro.2006.08.007

[24] Theodore Groves. 1973. Incentives in Teams. *Econometrica* 41, 4 (1973), 617–631. http://www.jstor.org/stable/1914085

[25] Joseph Halpern and Vanessa Teague. 2004. Rational Secret Sharing and Multiparty Computation: Extended Abstract. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing* (Chicago, IL, USA) *(STOC '04)*. Association for Computing Machinery, New York, NY, USA, 623–632.

[26] Thomas Kerber, Aggelos Kiayias, and Markulf Kohlweiss. 2021. KACHINA – Foundations of Private Smart Contracts. *2021 IEEE 34th Computer Security Foundations Symposium (CSF)* (2021), 1–16.

[27] Elias Koutsoupias and Christos Papadimitriou. 2009. Worst-case equilibria. *Computer Science Review* 3, 2 (2009), 65–69. https://doi.org/10.1016/j.cosrev.2009.04.003

[28] Alptekin Küpçü and Anna Lysyanskaya. 2010. Optimistic Fair Exchange with Multiple Arbiters. In *Computer Security – ESORICS 2010*, Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 488–507.

[29] Alptekin Küpçü and Anna Lysyanskaya. 2010. Usable Optimistic Fair Exchange. In *Topics in Cryptology - CT-RSA 2010*, Josef Pieprzyk (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 252–267.

[30] Clément Lesaege, Federico Ast, and William George. 2019. *Kleros Short Paper v1.0.7*. Technical Report. Kleros.

[31] Yang P. Liu and Yiling Chen. 2017. Sequential Peer Prediction: Learning to Elicit Effort using Posted Prices. In *AAAI*.

[32] David M. McEvoy. 2013. Enforcing compliance with international environmental agreements using a deposit-refund system. *International Environmental Agreements: Politics, Law and Economics* 13, 4 (2013), 481–496. https://doi.org/10.1007/s10784-013-9209-2

[33] Martin J. Osborne and Ariel Rubinstein. 1994. *A course in game theory*. The MIT Press, Cambridge, USA. electronic edition.

[34] Drazen Prelec. 2004. A Bayesian Truth Serum for Subjective Data. *Science (New York, N.Y.)* 306 (11 2004), 462–6. https://doi.org/10.1126/science.1102081

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

[36] Reinhard Selten. 1998. Axiomatic Characterization of the Quadratic Scoring Rule. *Experimental Economics* 1, 1 (01 Jun 1998), 43–61. https://doi.org/10.1023/A:1009957816843

[37] William Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance* 16, 1 (1961), 8–37. http://www.jstor.org/stable/2977633

[38] M. Walls. 2011. Deposit-Refund Systems in Practice and Theory. *Environmental Economics eJournal* (2011).

[39] Jens Witkowski and David C. Parkes. 2012. A Robust Bayesian Truth Serum for Small Populations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (Toronto, Ontario, Canada) *(AAAI'12)*. AAAI Press, 1492–1498.

[40] Haoqi Zhang, Yiling Chen, and David Parkes. 2009. A General Approach to Environment Design with One Agent. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (Pasadena, California, USA) *(IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002–2008.

[41] Haoqi Zhang and David Parkes. 2008. Value-Based Policy Teaching with Active Indirect Elicitation. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1* (Chicago, Illinois) *(AAAI'08)*. AAAI Press, 208–214.

[42] Ruiyu Zhu, Changchang Ding, and Yan Huang. 2019. 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* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 633–650. https://doi.org/10.1145/3319535.3363215

[43] David Zimbeck. 2015. *Two Party double deposit trustless escrow in cryptographic networks and Bitcoin.* Technical Report. BitHalo.