Privacy-Enhancing Auctions Using Rational Cryptography*

Peter Bro Miltersen[1], Jesper Buus Nielsen[1], and Nikos Triandopoulos[2]

[1] Dept. of Computer Science, Aarhus University, Denmark
[2] Dept. of Computer Science, Boston University, USA

Abstract. We consider enhancing with privacy concerns a large class of auctions, which include sealed-bid single-item auctions but also general multi-item multi-winner auctions, our assumption being that bidders primarily care about monetary payoff and secondarily worry about exposing information about their type to other players and learning information about other players' types, that is, bidders are *greedy then paranoid*. To treat privacy explicitly within the game theoretic context, we put forward a novel *hybrid utility* model that considers both monetary and privacy components in players' payoffs.

We show how to use rational cryptography to approximately implement any given *ex interim* individually strictly rational equilibrium of such an auction without a trusted mediator through a cryptographic protocol that uses only point-to-point authenticated channels between the players. By "ex interim individually strictly rational" we mean that, given its type and before making its move, each player has a strictly positive expected utility. By "approximately implement" we mean that, under cryptographic assumptions, running the protocol is a computational Nash equilibrium with a payoff profile negligibly close to the original equilibrium.

1 Introduction

1.1 The Problem: Realizing Privacy-Enhanced Auctions

Consider the following scenario: A *seller* S wants to sell some items to a subset of n *bidders* $P_1, P_2, \ldots, P_n$ using a sealed bid auction, e.g., a first-price or a second-price (Vickrey) auction if there is just one item. To optimize their expected payoff in these settings, the bidders P_i are to submit their true valuation of the items (e.g., in a Vickrey auction) or more generally a function of their true valuation (e.g., the Bayesian equilibrium strategy in a first-price auction) as their bid. However, in the scenario we suggest, matters are complicated by the following issues: First, bidders are not happy revealing any information related to their true valuation to the seller. Second, bidders would also be unhappy if other buyers gain information about their valuation. On the other hand, they would appreciate learning something about the valuations of the other players if they get the chance.

Some of these concerns can be handled by assuming the availability of a trusted *mediator* M. Such a trusted party can collect the bids, determine the winners, and ensure

* Supported by the *Center for Algorithmic Game Theory*, funded by The Carlsberg Foundation.

S. Halevi (Ed.): CRYPTO 2009, LNCS 5677, pp. 541–558, 2009.

that the seller and the winners get in touch with one another. Ideal mediation does not solve all problems though, as the outcome potentially depends on the type of all parties. Hence a player which is paranoid enough about leaking information about its type might abstain from reporting the true valuation simply for privacy reasons. In this paper we first investigate when we can expect to find mechanisms which the parties would be willing to participate in if executed by an ideal mediator M. We then investigate how to realize such a mechanism in a world without an ideal mediator. The first problem forces us to assume that the parties are more interested in winning the good than worried about privacy. To solve the second problem we propose to replace M by a secure multiparty computation (MPC), as follows:

1. The seller commits in advance to sell the items to the bidders that can present a document digitally signed by all bidders, stating that P_i is the buyer of some given items. The document should also specify at which price P_i is to get each item.
2. The bidders perform a secure multiparty computation that simulates the mediator of the mediated auction and produces a set of such signed documents, i.e., one document per each winner associating the winner to the correct item-value pairs.

Indeed, previous papers concerned with secure cryptographic implementations of auctions have suggested schemes along these lines, e.g., [21, 18]. Also, at least in one instance such a scheme (for a double auction) has been implemented in practice [2].

There are issues that make this not quite solve our problem. As an example, the introduced privacy concerns of the bidders dictate the use of joint computations that eventually produce non-symmetric outputs for the bidders, where only the winners see their own contracts; then, nothing enforces the winners to send the contracts and complete the transaction with S. This, e.g., destroys the standard equilibrium analysis of a Vickrey auction which crucially depends on the winner being forced to buy, to make it costly to bid higher than ones valuation. This suggests using a first-price auction instead, but even then it is not obvious that rational parties with privacy concerns will carry out the protocol outlined above.

In general, we wish to extend classical equilibrium analysis of auctions of game theory to cryptographic auction protocols and make an argument that a rational party *has no incentive to deviate from following the protocol as specified.* A concrete problem is *protocol participation.* In realizations of games with non-symmetric final payoffs (like auctions), an agent has no incentive to continue and complete the protocol as soon as he realizes that he cannot be a winner. In contrast, the traditional analysis of multiparty computation assumes that at least *some* parties are "honest" and will carry out all steps of the protocol, no matter what (Bradford *et al.* [3] study the problem of protocol-completion incentives that exist in an auction when participants realize that they cannot win the auction, but in a model where privacy is not captured in players' rationality). Many works on rational cryptography have analyzed secret sharing and multiparty computation as a game [12, 11, 9, 1, 7, 16, 20] but, aiming at simultaneous information exchange and modeling rationality through pure information loss/gain, these works cannot precisely model auctions with non-symmetric outcomes/payoffs and a setting where utilities are a mix of monetary utilities and privacy concerns.

Matters are complicated by the fact that even the mediated auction *does* leak some information (e.g, the mere fact that a bidder did not win gives him information about the winning bid(s)). Hence, it is intuitively clear that if the privacy has high weight, existing equilibria in the classical case are disturbed (e.g., truth telling is no longer even a Nash equilibrium for Vickrey auctions), and for a high enough emphasis on privacy, not taking part in the auction (say, by submitting the bid 0, independently of the valuation) becomes a strictly dominant strategy. Whatever analysis one obtains will have to be consistent with this fact.

Perhaps the biggest challenge, finally, is to design a protocol as the above in a way that can be realized using *today's Internet computing and communication machinery*. While there are results that allow removing mediators in very general classes of games [10, 13, 14], these works use communication channels such as simultaneous broadcast (like most works on rational cryptography) or physical envelopes that are quite restrictive or even unobtainable when considering a practical Internet-based implementation.

1.2 Outline of Our Contribution

In this paper, we suggest a rational cryptographic protocol for replacing a trusted mediator in a large class of auctions. The protocol uses only point-to-point authenticated channels between the buyers, and can therefore be implemented on the Internet.

We propose a protocol where the seller does not participate. If we allowed the seller to be an active entity in the protocol execution some steps of the protocol could be significantly simplified, but a solution without seller participation has the potential to allow for more applications. As an example, a resource-limited device outsourcing computations might prefer the potential companies to execute the auction determining the winning company-price pair and just have them inform it of the outcome. As described above, the outcome of the protocol is determined by the winners getting contracts digitally signed by all other participants. How such a digitally signed contract is enforced is not our concern here. We simply assume that such bit strings have monetary value.

Besides such monetary concerns, we have to assign utilities to players so that the privacy concerns outlined in the previous subsection are adequately modeled. Because of the monetary value of the signed document, we deviate from previous works on secure auction implementation where privacy was treated at a second-phase technical level *outside* of the scope of game and parties' strategies, but also from previous works in rational cryptography where utilities were *solely* concerned with gain or exposure of information. Instead, we propose a *hybrid utility model* where agents are interested in both monetary gain from participating in the auction as well as in maintaining the privacy of their type (e.g., valuation). Their actual utility is a linear combination of a *monetary utility* and an *information utility*. For the information utility, rather than postulating one particular utility measure, we allow players to have *any* privacy concerns, under a few technical restrictions, like not positively valuing loss of information. We note that a different hybrid utility model is studied by Halpern and Pass [8].

We consider a general class of auctions in the standard Bayesian setup of auction theory and *without* privacy concerns. We formally define the corresponding mediated game *with* privacy concerns, as modeled using our hybrid utilities. In general, as we indicated in an intuitive way in the previous subsection, if high weight is put on the information part of the hybrid utilities, then the equilibria of the privacy-aware game may be very different from the equilibria of the original game. However, for many interesting cases of auctions, for instance in a variant of the first-price auction with discrete valuations and bids, we observe that when the weight put on the information concern is "smaller" than the weight on the monetary concern, then the original auction mechanism (with a small twist) is an equilibrium of the mediated game.

To study auctions with privacy concerns for the Internet, where the seller does not participate, we introduce *mediation with reject*, a slightly relaxed mediated setting where the winners are given the choice to reject their contracts. This captures the following issue: at some specific point in the computation, the winners (and only those) will locally compute their contracts (similar to the *revelation point* of [12]); nothing prevents them from not sending the contract to the seller. As we will see, the reject option can drastically affect the equilibria.

Our main result is the following. We can relate a given equilibrium (suggested behavior) π of the mediated game to a corresponding suggested behavior π' of our unmediated cryptographic protocol so that π' has the same payoff profile as π, up to a negligible amount, and for computationally bounded agents following the protocol π' is an ϵ-Nash equilibrium where ϵ is negligible. Here, "negligible" is defined relative to the strength of the cryptography used. The assumption we need is the following: The equilibrium π should have an *ex interim* expected monetary utility for all players which is large compared to the players' privacy concerns. That is, after a player learns his type, but before he makes his move, his expected conditional monetary utility is large compared to how concerned he is about privacy—parties are "greedy-then-paranoid".

As an example, our protocol enables a variant of the first-price auction and the corresponding Bayesian bidding equilibrium to be conducted by computationally bounded, rational but *not* necessarily honest buyers over the Internet in a realistic way, without a trusted mediator and without participation of the seller. In this regard, our results can be viewed as a more realistic step towards privacy-aware extensions of computational and distributed mechanism design (e.g., Ch. 14 of [19]).

We remark that while Kol and Naor [11] identify ϵ-Nash equilibrium as a minimum rationality requirement for rational cryptography, a body of works [9, 1, 7, 16, 11, 12, 17], suggest using stronger solution concepts, most notably *iterated admissibility*, and equilibria that are *not susceptible to backward inductions* [11]. However, at the time of writing, there is no clear consensus about which equilibrium refinement is the "right one" for rational cryptography. This is especially true for the computational setting where one must refine computational Nash (i.e., ϵ-Nash) equilibrium rather than Nash equilibrium: while there is a significant body of game theoretic literature about refining exact Nash equilibrium that one can draw upon, there is little or no help from the game theory community about refining approximate Nash equilibrium.[1] We note that Kol and Naor [12] strongly argue that iterated admissibility is not an appropriate concept to use. We want to add the following observation. Computational Nash equilibrium is a solution concept for games played by *software*, not conscious agents. Thus, when we ponder whether a given equilibrium is sufficiently stable or whether deviations will be made, it seems that we should focus on whether the software will be modified *before*

[1] There is a good reason for this: many or most standard equilibrium refinements are defined or motivated by players caring about *infinitely* small differences in payoff. This is inconsistent with the philosophy of ϵ-Nash in a fundamental way.

it is executed, e.g., at the moment when a player learns his type (i.e., *ex interim*) rather than whether deviations will take place *during play* at particular information sets. In other words, we propose the following thesis: Meaningful refinements of computational Nash equilibrium should be definable in the *normal form* of the game, rather than the *extensive form*. We note that the concerns about susceptibility to backward induction raised by Kol and Naor are in fact not consistent with the conjunction of this thesis and the basic assumption underlying ϵ-Nash: That players do not care pursuing advantages that are negligible. We expect much interesting work in the future about how to refine computational Nash appropriately, but in the meantime we take the standpoint that even ϵ-Nash is a meaningful property as a minimal requirement for stability, and in some cases, such as ours, it is not trivial to achieve even this.

Sketch of the protocol. The idea behind our protocol is intuitive and quite simple. Given individual signing keys and corresponding (publicly known) verification keys for some signature scheme, and also their private bids, the agents engage a randomized joint computation during which the winners obtain digital contracts signed by all agents. Conceptually, the protocol is divided in a fixed (and large) number E of stages, called *epochs*. Sequentially during each epoch e, each agent P_i receives a value $V_{e,i}$ and thus has the opportunity to obtain a contract. The contracts are released to the winners during one, randomly chosen epoch $e_0 \in [E]$ (with probability 2^{-e} in epoch $e = 1, \ldots, E-1$), whereas all other received values (by non-winners P_i in epoch e_0, or by any agent at all other epochs) are set to a special nil value $\perp$. This randomized functionality is implemented by first using secure multiparty computation, at the end of which each agent P_k obtains an *additive* share of each value $V_{e,i}$ (or $\perp$ if agents provide invalid inputs). From this point on, the E epochs of the protocol are realized sequentially, by simply asking in a round-robin fashion each agent to send its share of $V_{e,i}$ to P_i, and repeat for all $i = 1, \ldots, n$. Agent P_i is asked to refuse to send his shares in subsequent reconstructions, as soon as he experiences denial to reconstruct his own value $V_{e,i}$.

To see why several epochs are needed, consider a solution where the contracts are always handed out in epoch $e_0 = 1$. If P_1 does not get a contract in round 1, it knows that some other P_i is the winner, hence P_i will receive a contract in round i. This contract might contain information on P_1's type, which means that P_1 might have incentive to make the protocol abort, by not sending its share. We deal with this using the, by now, standard trick of not having a known epoch in which the outcomes are revealed, to ensure that with positive probability any agent deviating at epoch $e < E$ destroys his winning possibility in a later epoch. This does not hold in epoch E, but $e_0 = E$ occurs only with negligible probability, so the protocol is an ϵ-Nash for a negligible ϵ.

When there are several winners, the above protocol does not work: A winner P_i already having received his contract could have incentive to make the protocol abort before the other winners received their contracts, as these contracts could contain information related to P_i's type. To solve this issue we let the winners learn all the information in their contracts in epoch e_0, but in an unsigned form. Then in epoch $e_0 + 1$ we let them learn their signed contracts. Now, when P_i gets his contract, it is too late to prevent the other winners from learning the information in their contracts, and the contracts themselves contain no new information. Depriving other winners of their contracts would only change *their* monetary utility, and we do not model envy.

Inspired by early work on rational cryptography (e.g., [9, 1, 7, 16]) this epoch-based protocol design has been recently used along with sequential revealing of secrets to achieve complete fairness in joint computations and information exchanging (e.g., [6, 12]). The non-symmetric outcomes in auction games and the use of only point-to-point communication create a different setting where our protocol operates in. But what further distinguishes our work is how fairness is reached between the many "greedy-then-paranoid" winners: the *decoupling* of the revelation of their winning state from the (subsequent) release of their winning award in combination with bidders rationality can guarantee protocol termination.

Paper structure. In Section 2 we provide a brief description of the classical auctions model in the (pure) mediated setting. In Section 3 we introduce a definitional framework for protocol games. In Section 4 we present the mediated setting with reject and discuss the existence in this model of privacy-enhanced Nash equilibria for first-price auctions. In Section 5 we present our protocol for realizing auctions over the Internet. In Section 6 we introduce privacy-enhanced Nash realization, our core proof technique for designing and proving privacy-enhanced Nash equilibria in a modular manner.

2 Classical Auctions

First, we recap the classical (i.e., privacy-oblivious) model of a sealed-bid auction as a Bayesian game with incomplete information. Such a game is played by parties (bidders) $P_1, P_2, \ldots, P_n$ competing for one or more items to be sold. The game starts with each bidder P_i receiving a private *type* $t_i \in T_i$ where T_i is the *type space* of the bidder. The vector $t = (t_1, t_2, \ldots, t_n)$ is drawn at random from a commonly known distribution on $T = T_1 \times T_2 \ldots \times T_n$. This distribution is known as the *common prior* and will also be denoted by T. Based on his type, bidder P_i strategically chooses and submits a *bid* b_i. That is, a *strategy* of party P_i is given by a map B_i mapping types to bids. Based on the bids $b = (b_1, b_2, \ldots, b_n)$ and possibly a random source, an *allocation mechanism* Mec now allocates the items to bidders and for each item computes a *price*. We write $(o_1, \ldots, o_n) = \mathrm{Mec}(b)$, where o_i is the *outcome* for P_i—i.e., o_i specifies which items P_i won and at which prices. The *monetary utility* of a winner P_j is $r_j = g(t, o)$ for some function g, while the payoff of a non-winner P_i is $r_i = 0$. As an example, in a single-item auction t_j could be the valuation of the item, o_j could specify the winning price p and r_j could be $t_j - p$ (this is the case for a risk neutral agent P_j as he gets the item at price p and values it t_j). For the case of the Vickrey auction, the winner P_j is the bidder with the highest bid, while the corresponding winning price p is the highest bid if the bid of the winner is removed. A Bayes-Nash (or simply Nash for brevity) equilibrium for the auction is a (possibly randomized) bidding strategy maximizing the expected payoff of each bidder, if other bidders follow their prescribed strategy.

3 Protocol Games

To enhance the classical auction with privacy concerns, we have to explicitly model privacy as part of the utility function and consider appropriate notions of equilibria. For this we in turn have to explicitly model the communication of the protocol, and the information collected by a party during the protocol execution.

3.1 Communication and Protocol Execution

We start with a formal communication and protocol execution model. It is convenient to use a unified model, which allows to capture both the mediated setting and the Internet-like setting using the same formalism, which we will call a *communication device*. To be able to use cryptography, we also want to model the fact that parties are computationally bounded to get the desired definitions; this we do by simply restricting the strategy space to poly-time strategies. The model we present in this section is not specific to auctions.

Communication devices. A protocol is of the form $\pi = (\pi_1, \ldots, \pi_n)$, where π_i is a program describing the strategy of party P_i. These programs communicate in rounds using a communication device $\mathcal{C}$. In each round, $\mathcal{C}$ takes an input $m_i \in \{0,1\}^d$ from each π_i and outputs a value $o_i \in \{0,1\}^d$ to each π_i. I.e., in each round, $\mathcal{C}$ is a function $(\{0,1\}^d)^n \to (\{0,1\}^d)^n, (m_1, \ldots, m_n) \mapsto (o_1, \ldots, o_n)$. Which function is computed might depend on the inputs and outputs of previous rounds and the randomness of $\mathcal{C}$.

Parties and strategies. We let the strategy π_i for each party P_i be an interactive circuit for R rounds. The circuit consists of $1+R$ circuits $\pi_i^{(0)}, \pi_i^{(1)}, \ldots, \pi_i^{(R)}$. The circuit $\pi_i^{(0)}$ takes $a+b$ bits as input and outputs $a+b$ bits, where a, b are integers specified by the circuit. In each round π_i takes as input a *state* $s \in \{0,1\}^a$, and a *message* $m \in \{0,1\}^b$ (from the communication device $\mathcal{C}$). The output of the circuit is parsed as an updated state $s' \in \{0,1\}^a$ and a message $m' \in \{0,1\}^b$ (for device $\mathcal{C}$). Initially, the state consists of a uniformly random bits and the message is P_i's type. In subsequent rounds, s is the updated state s' from the previous round and m is the value sent by $\mathcal{C}$ for that round.

Because we consider protocols using cryptography, we do not consider a single circuit π_i. Rather π_i specifies a family of circuits, namely a circuit $\pi_i(\kappa)$ for each value κ of the security parameter.[2] Each $\pi_i(\kappa)$ is allowed to have different state and message lengths $a(\kappa), b(\kappa)$. Similarly we let $\mathcal{C}$ specify a communication device $\mathcal{C}(\kappa)$ for each $\kappa \in \mathbb{N}$. Also, for technical reasons we adopt a non-uniform model, where the sequence of strategies $\pi_i(1), \pi_i(2), \ldots$ need not have a uniform description.[3] For a function $\tau : \mathbb{N} \to \mathbb{N}$ we use Π^τ to denote the *strategy space* consisting of all circuit families π_i where for all κ the size of $\pi_i(\kappa)$ is at most $\tau(\kappa)$. A strategy space Π^τ is always defined in context of some communication device $\mathcal{C}$ which for each κ expects (and produces) messages of some fixed size $d(\kappa) \in \mathbb{N}$. We require that Π^τ only contains circuit families where $b(\kappa) = d(\kappa)$ for all κ.

[2] The value of κ determines the key lengths of the underlying cryptographic primitives.

[3] Insisting on π_i having a uniform description might make it impossible to analyze the games for different values of κ independently, or would at least require an explicit argument that this can be done: Changing the strategies $\pi_i(\kappa)$ for some values of the security parameter κ might necessitate a change for other values to ensure that the sequence $\pi_1(1), \pi_1(2), \ldots$ still has a uniform description. The utility of changing strategy for one specific game (i.e., for a fixed κ) might therefore not be possible to define without considering the utility of changing strategy at other security levels, which seems unintuitive and might unnecessarily complicate analysis. Adopting a non-uniform model deals with such concerns in a straight-forward manner.

Executions. Let $\mathcal{C}$ be some communication device, let $\pi = (\pi_1, \ldots, \pi_n)$ be a protocol, where $\pi_i \in \Pi^\tau$, and let T be a distribution on types. An *execution* proceeds as in Fig. 1. We call $o = (o_1, \ldots, o_n) = (o_1^{(R)}, \ldots, o_n^{(R)})$ the *outcome* of the protocol. I.e., the outcome is the last round of outputs from $\mathcal{C}$. We call the output $w_i = (s_i^{(R+1)}, m_i^{(R+1)})$ of the last circuit $\pi_i^{(R)}$ of strategy π_i the *local output* of party P_i, and call $w = (w_1, \ldots, w_n)$ the *local outputs*. We use $(t, o, w) \leftarrow (\pi, \mathcal{C})(T)$ to denote the distribution of (t, o, w) on a *random execution*, i.e., for uniformly random initial states ρ, random $t \leftarrow T$ and uniform randomness of $\mathcal{C}$.

1. Sample $(t_1, \ldots, t_n) \leftarrow T$ and uniformly random $\rho_i \in \{0,1\}^a$ for $i = 1, \ldots, n$.
2. For $i = 1, \ldots, n$, run $\pi_i^{(0)}$ on (ρ_i, t_i) to produce $(s_i^{(1)}, m_i^{(1)})$. Then for each round $r = 1, 2, \ldots, R$: First run $\mathcal{C}$ on $(m_1^{(r)}, \ldots, m_n^{(r)})$ to produce $(o_1^{(r)}, \ldots, o_n^{(r)})$, and then, for $i = 1, \ldots, n$, run $\pi_i^{(r)}$ on $(s_i^{(r)}, o_i^{(r)})$ to produce $(s_i^{(r+1)}, m_i^{(r+1)})$.

Fig. 1. An execution

Utilities. The *utility* of P_i is a real valued function u_i. We assume that u_i is a function of the types, the outcomes and the local outputs. We use u to denote $(u_1, \ldots, u_n)$. We use $u_i(T, \pi, \mathcal{C})$ to denote the *expected utility* of P_i, i.e., the expected value of $u_i(t, o, w)$ for a random execution $(t, o, w) \leftarrow (\pi, \mathcal{C})(T)$.

3.2 The Mediator and the Internet as Communication Devices

For analyzing protocols for Internet-like networks we need a communication device $\mathcal{C}^{\texttt{int}}$ modeling communication on the Internet. Ideally we want $\mathcal{C}^{\texttt{int}}$ to closely reflect how messages are delivered on the Internet. Since our results are very robust with respect to the exact specification of $\mathcal{C}^{\texttt{int}}$ we will, however, use a rather idealized device.

A communication device $\mathcal{C}^{\texttt{int}}_{\text{gen},\text{Out}}$ parametrized by gen and Out works as follows:

set up PKI: In round 1, sample a key pair $(pk_i, sk_i) \leftarrow \text{gen}(1^\kappa)$ for each P_i and output $((pk_1, \ldots, pk_n), sk_j)$ to P_j for $j = 1, \ldots, n$.

protocol execution: In rounds $r = 2, \ldots, R-1$, the input from each party P_i is parsed as a message $m_i \in \{0,1\}^k$ for some fixed k. The output to $P_{r \bmod n}$ is $(m_1, \ldots, m_n)$. The output to all other parties is `silence`.

define outcome: In round $r = R$, compute $(o_1, \ldots, o_n) = \text{Out}(msg)$, where msg are all messages sent in the previous rounds, and output the outcome o_i to P_i.

Fig. 2. An Internet-Like Device $\mathcal{C}^{\texttt{int}}_{\text{gen},\text{Out}}$

We assume that the device can deliver secure messages directly between each pair of parties. This can be achieved using standard Internet technology, e.g., by establishing SSL connections between each pair of parties. Using such a model we avoid the introduction of unnecessary complications, like the exact structure of the network used to carry the messages. On the other hand, we do not want the simplification of $\mathcal{C}^{\mathtt{int}}$ to make the model unrealistic. One issue which we explicitly do not want $\mathcal{C}^{\mathtt{int}}$ to allow is simultaneous message exchange. We do this by saying that on $\mathcal{C}^{\mathtt{int}}$, in each round one predefined party receives messages from all other parties. Finally we assume the existence of a public-key infrastructure PKI. We model this in a simplistic manner by letting the device distribute the keys. In the last round the device will define an outcome, by the last set of messages output to the parties. We assume that this is a function Out of all the messages sent in previous rounds. Details are given in Fig. 2.

The communication device $\mathcal{C}^{\mathtt{med}}_{\mathrm{Mec}}$ for standard mediation is $\mathcal{C}^{\mathtt{rej}}_{\mathrm{Mec}}$ in Fig. 3 on page 552, but without **allow reject**. The recommend strategy $\pi^{\mathtt{med}}_j$ for each P_j is to input $b_j \leftarrow B_j(t_j)$ and to locally output $w_j = (t_j, o_j)$.

3.3 Information and Monetary Utilities

Information utilities. We now turn our attention to the valuation of the information collected and leaked during the protocol execution. For this we use the local outputs.

We let the local output w_i capture the type information collected by P_i. I.e., if P_i wants to take some type information with it from the execution, it outputs it as part of w_i. We assume that P_i valuates the type information collected using an *information utility* $q_i(t, w)$. Note that q_i can measure information collected by P_i as well as by other parties: maybe $q_i(t, w) = 1$ if $w_i = t_1$ but $q_i(t, w) = -1$ if $w_1 = t_i$, where $i \neq 1$.

We allow q_i to express *arbitrary* privacy concerns, except for two restrictions: To ensure that q_i is consistent with the view of knowledge from cryptography, where knowledge is the information which can be computed in poly-time, we require that q_i is poly-time computable. We also need that q_i does not positively valuate loss of information. Let $(w_1, \ldots, w_n)$ be any distribution and let $(w'_1, \ldots, w'_n)$ be the distribution where $w'_i = f(w_i)$ for a poly-time function f and $w'_{-i} = w_{-i}$. Then we require that $q_i(t, (w'_1, \ldots, w'_n)) \leq q_i(t, (w_1, \ldots, w_n)) + \epsilon$, where ϵ is negligible. In words: losing information about w_i (we think of $f(w_i)$ as throwing away information about w_i), and all other things being equal, cannot be valuated as significantly positive by P_i. We call q_i *admissible* if it has these two properties. Below we assume that all q_i are admissible.

Our protocols will work only for privacy concerns which are sufficiently small compared to the expected utility of playing the game. So it is convenient to have a measure of the privacy concerns: For an information utility $q_i(t, w)$ we call $\|q_i\| = \max_{t,w} q_i(t, w) - \min_{t,w} q_i(t, w)$ the *weight of the information utility* or *privacy weight*.

We will not be concerned about how the utility q_i measures privacy concerns, as we are going to develop protocols that are ϵ-Nash *for all admissible measures* $q = (q_1, \ldots, q_n)$ with sufficiently small weight compared to the expected monetary utility.

Monetary utilities. Complementing the information utility we have the notion of a *monetary utility*, which is just a utility function $r_i(t, o)$ that depends only on the types and the outcomes. For generality we allow r_i to change with κ. We do, however, assume that the absolute value of r_i is bounded by a polynomial in κ. The intuitive reason for this assumption is that we need to use cryptography, which withstands only poly-time attacks. In concrete terms, if you use a protocol where it would cost \$1000000 to buy enough computing power to break the cryptography, do not use it to play a game where anyone can win $\$1000001$. Bounding the monetary utility by a polynomial can be seen as an extremely crude way to deal with the price of computation in the utility function.

We design mechanisms which work only if the expected monetary utility of the parties is large compared to how they valuate information. We define a measure of this. For any t_i occurring with non-negligible probability as component i in $(t_1, \ldots, t_n) \leftarrow T$, let $(t, o, w) \leftarrow (\pi, \mathcal{C})(T)_{t_i}$ denote the conditional distribution of $(t, o, w) \leftarrow (\pi, \mathcal{C})(T)$ given that the i'th component of t is t_i, and let I_i denote the expected value of $u_i(t, o, w)$ for $(t, o, w) \leftarrow (\pi, \mathcal{C})(T)_{t_i}$. We call I_i the *ex interim* expected utility of P_i for t_i, i.e. its expected utility after seeing type t_i. For a given security level κ we let $\gamma(\kappa)$ be the minimum over all parties P_i and all t_i of the *ex interim* expected utility of P_i given t_i. We call $\gamma : \mathbb{N} \rightarrow \mathbb{R}$ the ex interim *rationality* of $(T, \pi, \mathcal{C})$.

3.4 Privacy-Enhanced Nash Equilibrium

When we design a mechanism, we can control the monetary utility $r_i(t, o, w) = r_i(t, o)$. In principle parties can have arbitrary utilities $u_i(t, o, w)$, even if running a protocol with the purpose of implementing some mechanism. However, we only consider settings where the part of the utility which cannot be explained as monetary utility from the designed mechanisms can be explained by an admissible measure of privacy. I.e., we assume that $q_i(t, o, w) = u_i(t, o, w) - r_i(t, o)$ is an admissible measure of privacy, s.t. $q_i(t, o, w) = q_i(t, w)$. Hence $u_i(t, o, w) = r_i(t, o) + q_i(t, w)$.

For the later schemes involving cryptography, we follow Kol and Naor [11] who argued that ϵ-Nash equilibrium for negligible ϵ is the appropriate minimum rationality requirement for "information games".

Definition 1. *For a single protocol π (i.e., for fixed κ), a strategy space Π^τ, a distribution T on types, and $\epsilon \in \mathbb{R}$, $\epsilon > 0$, we call π an ϵ-Nash equilibrium (for $T, \Pi^\tau, \mathcal{C}$) if it holds for all parties P_i and all $\pi_i^* \in \Pi^\tau$ that $u_i(T, (\pi_i^*, \pi_{-i}), \mathcal{C}) - u_i(T, \pi, \mathcal{C}) \leq \epsilon$. For a protocol π (specified for all κ), strategy space Π^τ, a distribution T on types, we call π a* computational Nash equilibrium *(for $T, \Pi^\tau, \mathcal{C}$) if for all polynomials τ there exists a negligible ϵ such that $\pi(\kappa)$ is an $\epsilon(\kappa)$-Nash equilibrium (for T, $\Pi^{\tau(\kappa)}, \mathcal{C}$) for all κ.*

Our notion of computational Nash is technically slightly different from the original notion introduced by Dodis *et al.* [4], in that we use a non-uniform model, as motivated before. The notion is, however, similar enough that we feel that we can soundly reuse the terminology of a computational Nash equilibrium.

As already mentioned, implementations of monetary mechanisms can only be expected to work if the weight of the privacy concerns is relatively small. We thus capture the size of the information utility in the definition of privacy-enhanced Nash equilibria.

Definition 2. *Fix a monetary utility r and a privacy weight α. We call a protocol a* privacy-enhanced Nash equilibrium *(for r and α) if it is a computational Nash equilibrium for $u = r + q$ for all admissible privacy measures q with $\|q\| \triangleq \max_i \|q_i\| \leq \alpha$.*

In words, a privacy-enhanced Nash equilibrium has the property that no matter how the parties valuate information (as long as it has weight at most α), there is no deviation which will allow any party to learn more valuable information, unless such a deviation would have it lose an equivalent amount of monetary utility. This implies that there is no way a party P_j can efficiently extract knowledge from its view of the protocol extra to that of its local output w_j. If there was, it could do so and output this extra knowledge, which would make some q_i prefer this. Therefore the recommended local outputs of a privacy-enhanced mechanism precisely specify what information each party can collect; not as an explicit requirement, but because we use computational Nash equilibrium as solution concept.

We extend the previously defined notions to cover also collusions of size t. In Definition 1 we consider $C \subset \{1, \ldots, n\}$ with $|C| \leq t$ and we consider deviations π^*_C consisting of π^*_i for $i \in C$. We call π *t-resilient* if $u_i(T, (\pi^*_C, \pi_{-C}), \mathcal{C}) - u_i(T, \pi, \mathcal{C}) \leq \epsilon$ for all $i \in C$. I.e., for all collusions of size t and all possible deviations, not even a single party in the collusion gets extra utility. This directly defines the notions of *t-resilient computational Nash equilibrium* and *t-resilient privacy-enhanced Nash equilibrium.*

As a concrete example of a privacy-enhanced Nash equilibrium for an auction mechanism with standard mediation, we consider a single-item sealed-bid first-price auction with three bidders and independent private valuations, each distributed uniformly in $\{1, 3\}$. The bidding space is the natural numbers, including 0. A general theory of equilibria of first-price auctions with integral valuations and bids is the topic of a recent paper by Escamocher *et al.* [5]. For the special case at hand, it is straightforward to check that the symmetric profile $\pi = (B_1, B_2, B_3)$, with $B_1 = B_2 = B_3$, $B_1(1) = 0$ and $B_1(3) = 1$, is a Nash equilibrium of the classical (privacy-oblivious) auction. The *ex interim* expected payoff of a bidder with valuation 1 is $1/12$ and the ex interim expected payoff of a bidder with valuation 3 is $7/6$; since payoffs are strictly bigger than 0, it is easy to check that for any privacy measure with sufficiently small weight, the equilibrium persists.

4 Mediation with Reject and Predictable Mechanisms

In what follows we consider a very general class of allocation mechanisms, but with some non-trivial restrictions. A first restriction we need is that if $(o_1, \ldots, o_n) = \mathrm{Mec}(b)$, then the utility of P_i is 0 if $o_i = \texttt{sorry}$, this outcome indicating that P_i got to buy no items. Instead, we call a party P_i with $o_i \neq \texttt{sorry}$ a *winner*. Our only use of `sorry` is to define mediation with reject below.

Towards designing a protocol that implements an auction on an Internet-like network without the participation of the seller and that is a privacy-enhanced Nash equilibrium, we first study privacy-enhanced Nash equilibria for a highly idealized setting that better fits the real-world setting. The idealized setting that we consider is called *mediation with reject*: here, the parties are allowed to reject the outcome of the auction and receive monetary utility 0 instead of the contract. Details are given in Fig. 3 on the next page.

It is easy to check that the standard truth telling equilibrium of a second-price auction is in general *not* a privacy-enhanced Nash equilibrium in the setting of mediation with reject: The fact that the winner is not forced to make the transaction makes bidding infinity (or the highest possible bid) a dominant strategy. For non-trivial privacy concerns, this dominant strategy is also a strictly better response than truth telling to a strategy profile where the other bidders bid truthfully. Thus, mediation with reject is a setting

Parameterized by a number of rounds R, the communication device $\mathcal{C}^{\texttt{rej}}_{\mathrm{Mec}}$ works as follows:

compute result: In round 1, take input b_i from each P_i, let $b = (b_1, \ldots, b_n)$, sample $(o_1, \ldots, o_n) \leftarrow \mathrm{Mec}(b)$, where $o_i \neq \texttt{sorry}$ iff P_i is a winner.

allow reject: For $i = 1, \ldots, n$: Output o_i to P_i. If P_i with $o_i \neq \texttt{sorry}$ does not input accept before round R, set $o_i \leftarrow \texttt{sorry}$.

define outcome: In the last round $r = R$, output the current value of o_i to each P_i.

side-channel: In rounds $r = 2, \ldots, R-1$, allow point-to-point communication as in $\mathcal{C}^{\texttt{int}}$.

The recommend strategy $\pi^{\texttt{rej}}_j$ for each P_j is to input $b_j \leftarrow B_j(t_j)$ and accept and locally output $w_j = (t_j, o_j)$.

Fig. 3. The Mediated Setting with Reject $(\pi^{\texttt{rej}}_B, \mathcal{C}^{\texttt{rej}}_{\mathrm{Mec}})$ for mechanism $(B_1, \ldots, B_n, \mathrm{Mec})$

where we observe a *separation* between first-price and second-price auctions with respect to the existence of reasonable privacy-enhanced Nash equilibria, fully justifying the importance of this abstraction.

It will, however, follow from our main result that a large class of privacy-enhanced Nash equilibria for the standard mediated setting are also privacy-enhanced Nash equilibria in the mediated setting with reject. We need a definition to phrase this result.

Definition 3. *A mechanism is called* predictable *if for each P_i, each type t_i for P_i and each bid b_i for P_i the expected monetary utility of P_i,* given that P_i bids b_i and gets $o_i \neq \texttt{sorry}$, *depends only on t_i and b_i. Furthermore, this number $m_i(t_i, b_i)$ can be computed from t_i and b_i in poly-time.*

Clearly a Vickrey auction is not predictable, as the expected utility depends on the second largest bid, but a first-price auction *is* predictable: given that a party wins, its utility only depends on its own type and bid.

We can show that if Mec is predictable and $\gamma \geq 2\alpha$ (where α is the weight of the information utility and γ is the *ex interim* rationality) and $\pi^{\texttt{med}}_{\mathrm{Mec}}$ is a privacy-enhanced Nash equilibrium for $(T, u, \mathcal{C}^{\texttt{med}}_{\mathrm{Mec}})$, then $\pi^{\texttt{rej}}_{\mathrm{Mec}}$ is a privacy-enhanced Nash equilibrium for $(T, u, \mathcal{C}^{\texttt{rej}}_{\mathrm{Mec}})$. This shows that one can construct interesting equilibria for a mediated setting with reject. The intuition why "predictable equilibria" do not have a problem with reject, follows from the proof sketch we give in Section 5.

Privacy-enhanced Nash equilibria for first-price auctions with standard mediation exist for certain settings of the parameters, as exemplified in Section 3, and these are predictable. We therefore have interesting Nash mechanisms for the mediated setting with reject. Other examples of mechanisms for which one can design mechanisms for the setting with reject include auctions where a number ℓ of uniform items are sold to bidders with unit demand, selling to the highest ℓ bidders at their bidding price—such an auction is predictable.

5 Rational Auctions for Internet-Like Networks

We now present our Internet-based and privacy-enhanced Nash-equilibrium protocol for realizing auctions.

Assigning value to signed contracts. We want an unmediated protocol for the device $\mathcal{C}^{\texttt{umed}} = \mathcal{C}^{\texttt{int}}_{\text{gen},\text{Out}}$ for gen and Out described below. For this to be meaningful we need to make explicit how the Internet protocol allocates monetary utility. This is a fundamentally problematic issue as we are, after all, considering a pure communication protocol which anyone can set up and run without money being exchanged. As indicated in the introduction, we assign monetary value to a document if it is a possible winners' outcome for Mec and is signed by all parties.

Taking uniform items, unit-demand, first-prices auctions as an example, we can make the assumption that the seller is willing to sell to the first ℓ parties presenting a document including the party's name and a price (over the reservation price), if it is signed by all parties. This immediately assigns monetary value to commonly signed contracts. One could also use society to enforce signed contracts (cf. [15]).

In more detail, we assume that the key pair generated by gen for each party P_i consists of a verification key vk_i for an existentially unforgeable digital signature scheme and the signing key sk_i. We call σ_i a *contract* on (i, o_i) if $\sigma_i = (\sigma^1, \ldots, \sigma^n)$ and each σ^j is a valid signature of (i, o_i) under vk_j. We use $\text{Contract}((i, o_i), sk)$ to denote the computing of such σ_i. We define $(o_1, \ldots, o_n) = \text{Out}(msg)$ by letting $o_i = O_i$ if P_i at some point sent a valid contract on (i, O_i) to itself. We let $o_j = \texttt{sorry}$ for all other parties. For a specific mechanism, we need a way to resolve what happens if a party inputs several, different signed contracts or the parties input signed contracts not consistent with an outcome of Mec. All we need for our proof to go through is that the defined outcome only depends on the contents (i, o_i) of the signed contracts and the global order in which the device received them, like for the uniform items, unit-demand, first-prices auction above.

Mediation via a secure protocol. We show how to implement a privacy-enhanced Nash $\pi^{\texttt{rej}}_{\text{Mec}}$ in the Internet setting described in the above section. The idea is to compute the outcomes $(o_1, \ldots, o_n) = \text{Mec}(b)$ as in the mediated setting with reject, using a secure MPC protocol, but then release the signed outcomes in a particular manner. The release phase will consist of E so-called epochs indexed $e = 1, \ldots, E$, each consisting of n tries indexed $i = 1, \ldots, n$. We index a try i within an epoch e by (e, i). In try (e, i) party P_i is given a value $V_{i,e}$, if the other parties allow it. The recommended strategy is to allow all deliveries, but as soon as a party has been denied a delivery, it will deny all parties their deliveries in all following tries. There is a special epoch $e_0 \in \{1, \ldots, E-1\}$. The epoch e_0 is chosen using a probabilistic function $e_0 \leftarrow \text{Epoch}(E)$, where $e_0 \in \{1, \ldots, E-1\}$ and $\Pr[e_0 = e] = 2^{-e}$ for $e = 1, \ldots, E-2$. If P_i is not a winner, then $V_{e,i} = \texttt{sorry}$ for all epochs e. If P_i is a winner, then $V_{e,i} = \texttt{sorry}$ for $e \notin \{e_0, e_0 + 1\}$, and $V_{e_0,i} = o_i$ and $V_{e_0+1,i} = \text{Contract}((i, o_i), sk)$. When P_i receives $\text{Contract}((i, o_i), sk)$, it sends it to the seller (formally it sends it to itself and the device defines P_i to be a winner, by letting o_i be P_i's final output).

We use some notation for the $V_{e,i}$ values: For any $((o_1, \sigma_1), \ldots, (o_n, \sigma_n))$ and epoch $e_0 \in \{1, \ldots, E-1\}$ we define $V = (V_{1,1}, \ldots, V_{1,n}, V_{2,1}, \ldots, V_{E,n}) = \text{Values}(((o_1, \sigma_1), \ldots, (o_n, \sigma_n)), e_o, E)$, where for all P_i, $V_{e_0,i} = o_i$, $V_{e_0+1,i} = \sigma_i$ and $V_{e,i} = \texttt{sorry}$ for $e \notin \{e_0, e_0 + 1\}$.

We use a secure MPC to compute sharings of the values $V_{e,i}$. Given inputs $(b_1, \ldots, b_n)$, the protocol securely samples $V = (V_{1,1}, \ldots, V_{1,n}, V_{2,1}, \ldots, V_{E,n})$ and generates sharings $(S_{1,1}, \ldots, S_{E,n}) \leftarrow \text{Sharings}(V)$, where $S_{e,i} = (S_{e,i}^{(1)}, \ldots, S_{e,i}^{(n)})$ is an n-out-of-n sharing of $V_{e,i}$, where the shares are authenticated such that P_i can validate their correctness. Then the protocol gives all $S_{e,i}^{(j)}$ to P_j. The MPC protocol is chosen to tolerate the active corruption of up to $t = n - 1$ parties. With this threshold termination cannot be guaranteed. The protocol should, however, guarantee that all parties P_j which received an output $y_j \neq \perp$, where $\perp$ is some designated error symbol, received a correct output. Furthermore, the protocol should guarantee that $y_j \neq \perp$ for all parties if all parties followed the protocol. After the secure MPC protocol terminates, the parties reconstruct the sharings. The details of the complete protocol $\pi_{\mathrm{Mec}}^{\mathtt{umed}}$ are given in Fig. 4.

The unmediated protocol for communication device $\mathcal{C}^{\mathtt{umed}}$. The recommend strategy $\pi_j^{\mathtt{umed}}$ for P_j is as follows:

1. Receive (pk, sk_j) from the communication device.
2. In the rounds with point-to-point communication, run the code of P_j in a secure MPC for the following probabilistic function f:
   - Each P_i inputs some b_i and some (pk', sk_i'), and receives output y_i, computed as:
     - If all P_i input the same pk', and sk_i' is a signature key for pk_i', then sample $(o_1, \ldots, o_n) \leftarrow \text{Mec}(b)$ and $e_0 \leftarrow \text{Epoch}(E)$. If $o_i \neq \mathtt{sorry}$, then let $\sigma_i = \text{Contract}((i, o_i), sk')$. If $o_i = \mathtt{sorry}$, then let $\sigma_i = \mathtt{sorry}$. Let $V = (V_{1,1}, \ldots, V_{E,n}) \leftarrow \text{Values}(((o_1, \sigma_1), \ldots, (o_n, \sigma_n)), e_0, E)$, sample $(S_{1,1}, \ldots, S_{E,n}) \leftarrow \text{Sharings}(V)$, and let $y_i = (S_{1,1}^{(i)}, \ldots, S_{E,n}^{(i)})$.
     - Otherwise, let all $y_i = \perp$.

   Use inputs $b_j \leftarrow B_j(t_j)$ and $(pk', sk_j') = (pk, sk_j)$ to the MPC.
3. Afterward, initialize a variable $d_j \in \{\mathtt{allegiance}, \mathtt{defection}\}$, where $d_j = \mathtt{defection}$ iff the secure MPC protocol outputs $y_j = \perp$. If $d_j \neq \mathtt{defection}$, then parse y_j as shares $(S_{1,1}^{(j)}, \ldots, S_{E,n}^{(j)})$.
4. Use En rounds of point-to-point communication to sequentially run E *epochs*, each consisting of *tries* $i = 1, \ldots, n$. In epoch e, try i send $s_j = S_{e,i}^{(j)}$ to P_i if $d_j = \mathtt{allegiance}$ and send $s_j = \perp$ to P_i otherwise. In epoch e, try j, let $(s_1, \ldots, s_n)$ be the shares just sent by $P_1, \ldots, P_n$. If any share is invalid, then let $V_{e,j} = \perp$ and $d_j = \mathtt{defection}$. Otherwise, let $V_{e,j}$ be the value reconstructed from $(s_1, \ldots, s_n)$. If $V_{e,j}$ is a valid contract, then input it to $\mathcal{C}^{\mathtt{umed}}$.
5. If in some round $V_{e,j} = o_j$ was reconstructed, then give the local output $w_j = (t_j, o_j)$. Otherwise, give the local output $w_j = (t_j, \mathtt{sorry})$.

Fig. 4. The Unmediated Protocol $\pi_{\mathrm{Mec}}^{\mathtt{umed}}$

Theorem 1. *Let* Mec *be any predictable mechanism. Assume that* $(\pi_{\mathrm{Mec}}^{\mathtt{med}}, \mathcal{C}_{\mathrm{Mec}}^{\mathtt{med}})$ *is a privacy-enhanced Nash equilibrium, let* γ *be the* ex interim *rationality and let* α *be the weight of the information utility. If* $\gamma \geq 2\alpha$*, then* $(\pi_{\mathrm{Mec}}^{\mathtt{umed}}, \mathcal{C}^{\mathtt{umed}})$ *is a privacy-enhanced Nash equilibrium with a utility profile negligibly close to that of* $(\pi_{\mathrm{Mec}}^{\mathtt{med}}, \mathcal{C}_{\mathrm{Mec}}^{\mathtt{med}})$.

Proof. (Sketch.) We want to argue that no P_i has an incentive to deviate. We look at two cases: Case I is the situation where P_i saw a reconstructed value of the form

$V_{e,i} \neq \texttt{sorry}$. Case II is the situation where a party P_i only saw reconstructed values of the form $V_{e,i} = \texttt{sorry}$.

We first argue that a party P_i in Case I has no incentive to deviate. We look at two sub-cases. First, assume that P_i received $V_{e,i} = \mathrm{Contract}((i, o_i), sk)$. Then it can no longer gain monetary utility: it has its contract and cannot receive another one, except by breaking the signature scheme (infeasible by assumption). It cannot gain information utility either, as all information has already been handed out: When P_i has received $V_{e,i} = \mathrm{Contract}((i, o_i), sk)$ the game is already in epoch $e_0 + 1$, and all winners P_j received o_j in epoch e_0 and $\mathrm{Contract}((j, o_j), sk)$ leaks no information on the types extra to o_j.[4] Second, assume that P_i received $V_{e,i} = o_i$ but did *not* yet receive $\mathrm{Contract}((i, o_i), sk)$. If P_i sends an incorrect share to any P_j, then P_j will punish back and P_i will not receive $\mathrm{Contract}((i, o_i), sk)$. It can essentially be argued that for any deviation there is a better deviation which never inputs a bid which will lead to a monetary utility less than $\gamma/2$ if the bid wins.[5] So, we can assume that the loss of the contract gives a loss of $\gamma/2 \geq \alpha$ in monetary utility. Aborting the protocol might gain information utility by withholding some (j, o_j), but at most utility α. So by sending an incorrect share, P_i gains utility at most $\alpha - \gamma/2 \leq 0$.

We then look at a party P_i in case II and, say, in epoch e, try j. Let S be the event that all values reconstructed by P_i until now were $\texttt{sorry}$, R the event that all values o_j with $o_j \neq \texttt{sorry}$ have been reconstructed at the corresponding winners P_j, W the event that P_i is a winner, $s = \Pr[S]$, and $w = \Pr[W]$.

We consider a party P_i which only saw $\texttt{sorry}$, which means that in the view of P_i, it is a winner with probability $\Pr[W|S] = \Pr[W \wedge S]/s$, and in the view of P_i the probability that all o_j with $o_j \neq \texttt{sorry}$ have not been reconstructed is $\Pr[\bar{R}|S] = \Pr[\bar{R} \wedge S]/s$. If P_i makes the protocol abort and P_i is a winner he loses γ' in utility, where γ' is the expected utility of P_i given that he is a winner. If P_i makes the protocol abort and $\bar{R}$, then he withholds the information o_j from at least one winner P_j and therefore gains up to α in privacy utility—if R, then no information is withheld and no privacy utility is gained. Therefore the maximal gain in utility is upper bounded by $-(\Pr[W \wedge S]/s)\gamma' + (\Pr[\bar{R} \wedge S]/s)\alpha$. To show that this is non-positive it is sufficient to show that $\Pr[\bar{R} \wedge S]\alpha - \Pr[W \wedge S]\gamma' \leq 0$. We have that $\Pr[W \wedge S] = \Pr[W \wedge (e_0 > e \vee (e = e_0 \wedge i > j))] \geq \Pr[W \wedge e_0 > e] = w2^{-e}$ and $\Pr[\bar{R} \wedge S] \leq \Pr[\bar{R}] \leq \Pr[e_0 \geq e] = 2^{-e+1}$. Since γ' is the expected monetary utility when P_i is a winner, it follows that $\gamma = w\gamma' + (1 - w)0$ and $\gamma' = \gamma/w$. So, $\Pr[\bar{R} \wedge S]\alpha - \Pr[W \wedge S]\gamma' \leq 2^{-e+1}\alpha - (w2^{-e})\gamma/w = 2^{-e}(2\alpha - \gamma) \leq 0$, as $\gamma \geq 2\alpha$.

6 Nash Implementation and Hybrid Proofs

The full proof of Theorem 1 is extensive, as handling the use of cryptography posses some challenges when fleshing out the above proof sketch. We do, however, have space to describe the general proof strategy.

[4] For this argument to work it is essential that all o_i are handled out *before* the contracts σ_i: if P_i received σ_i before a winner P_j with $j > i$ received the information o_j, P_i could find utility in aborting the protocol, thus withholding the information o_j from P_j.

[5] The full argument is slightly different: The argument uses the predictability to avoid playing such bad bids, replacing them by the recommended bid—which gains utility.

The idea is to start with an idealized version of the protocol, for a device much like the mediated setting with reject, and then introduce more and more of the details and cryptographic tools, and for each step prove that the new protocol is equivalent to the previous one. The value of such an approach when using cryptographic primitives is testified by the widespread use of hybrid proofs in the cryptographic literature.

We introduce a notion of *Nash realization* which allows to structure such proofs. Consider an idealized communication device $\mathcal{C}^{\mathtt{ide}}$ (as e.g. $\mathcal{C}^{\mathtt{rej}}_{\mathrm{Mec}}$) and a recommended protocol $\pi^{\mathtt{ide}}$ for $\mathcal{C}^{\mathtt{ide}}$, as well as a closer to real-life communication device $\mathcal{C}^{\mathtt{imp}}$ (like $\mathcal{C}^{\mathrm{umed}}$) and a protocol $\pi^{\mathtt{imp}}$ for $\mathcal{C}^{\mathtt{imp}}$. We call $(\mathcal{C}^{\mathtt{imp}}, \pi^{\mathtt{imp}})$ a realization of $(\mathcal{C}^{\mathtt{ide}}, \pi^{\mathtt{ide}})$ if the parties do not have more incentives to deviate when they interact in $(\mathcal{C}^{\mathtt{imp}}, \pi^{\mathtt{imp}})$ than when they interact in $(\mathcal{C}^{\mathtt{ide}}, \pi^{\mathtt{ide}})$.

Definition 4. *Fix a distribution T on types and a monetary utility $r = (r_1, \ldots, r_n)$. Let $(\mathcal{C}^{imp}, \pi^{imp})$ and $(\mathcal{C}^{ide}, \pi^{ide})$ be two settings. We say that $(\mathcal{C}^{imp}, \pi^{imp})$ is a t-*resilient privacy-enhanced Nash realization *of $(\mathcal{C}^{ide}, \pi^{ide})$ if for all $u = r + q$, where $q = (q_1, \ldots, q_n)$ are admissible measures of privacy with weight at most α, there exists a negligible ϵ such that:*

No less utility: *For all P_l, $u_l(T, \pi^{imp}, \mathcal{C}^{imp}) \geq u_l(T, \pi^{ide}, \mathcal{C}^{ide}) - \epsilon$.*

No more incentive to deviate: *For all $C \subset \{1, \ldots, n\}$, $|C| \leq t$, all strategies $\pi_C^{imp^*}$ for $\mathcal{C}^{imp}$, there exists a strategy $\pi_C^{ide^*}$ for $\mathcal{C}^{ide}$ so that $u_l(T, (\pi_C^{ide^*}, \pi_{-C}^{ide}), \mathcal{C}^{ide}) \geq u_l(T, (\pi_C^{imp^*}, \pi_{-C}^{imp}), \mathcal{C}^{imp}) - \epsilon$ for all $l \in C$.*

Theorem 2. *For fixed T and r, it holds for all settings $(\mathcal{C}, \pi)$, $(\mathcal{D}, \gamma)$ and $(\mathcal{E}, \delta)$ that:*

Preservation: *If $(\mathcal{C}, \pi)$ is a t-resilient privacy-enhanced Nash realization of $(\mathcal{D}, \gamma)$ and γ is a t-resilient privacy-enhanced Nash equilibrium for $\mathcal{D}$, then π is a t-resilient privacy-enhanced Nash equilibrium for $\mathcal{C}$ with a utility profile negligibly close to that of $(\mathcal{C}, \gamma)$, i.e., $|u_l(T, \pi, \mathcal{C}) - u_l(T, \gamma, \mathcal{D})|$ is negligible for all P_l and for all considered $u = r + q$.*

Transitivity: *If $(\mathcal{C}, \pi)$ and $(\mathcal{D}, \gamma)$ are t-resilient privacy-enhanced Nash realizations of $(\mathcal{D}, \gamma)$ and $(\mathcal{E}, \delta)$ respectively, then $(\mathcal{C}, \pi)$ is a t-resilient privacy-enhanced Nash realization of $(\mathcal{E}, \delta)$.*

Though this theorem is fairly easy to verify, we find the notion of Nash realization an interesting conceptual contribution, as it allows to structure hybrid proofs in a game theoretic setting. The notion can also be used for other purposes. We can, e.g., show that our protocol in Fig. 4 is an $(n-1)$-resilient privacy-enhanced Nash realization of an information theoretic secure version of the protocol, where the $V_{e,i}$ values are computed by the device and leaked in the same epoch/try structure as in Fig. 4, depending on whether or not parties input `send` or `hold` in each try. Here the notion is used to analyze a property we could not have seen by only looking at equilibria in the unmediated protocol: The result shows that our use of cryptography does not give any further incentives for deviations, to *any* size of collusion, over what is present in this highly idealized setting, which gives an extra reassurance that the cryptography was used soundly.

We complete the proof by showing that the information theoretic idealization is a privacy-enhanced Nash equilibrium. By *preservation* this result carries over to the unmediated setting. In fact, designing any t-resistant privacy-enhanced Nash equilibrium for the information theoretic setting would directly give one for the Internet too.

References

[1] Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: PODC 2006, pp. 53–62. ACM, New York (2006)

[2] Bogetoft, P., Damgård, I., Jakobsen, T., Nielsen, K., Pagter, J., Toft, T.: A practical implementation of secure auctions based on multiparty integer computation. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 142–147. Springer, Heidelberg (2006)

[3] Bradford, P.G., Park, S., Rothkopf, M.H., Park, H.: Protocol completion incentive problems in cryptographic Vickrey auctions. Electronic Commerce Research 8(1-2), 57–77 (2008)

[4] Dodis, Y., Halevi, S., Rabin, T.: A cryptographic solution to a game theoretic problem. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 112–130. Springer, Heidelberg (2000)

[5] Escamocher, G., Miltersen, P.B., Santillan-Rodriguez, R.: Existence and computation of equilibria of first-price auctions with integral valuations and bids. In: AAMAS 2009 (2009)

[6] Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation. In: STOC 2008, pp. 413–422. ACM, New York (2008)

[7] Gordon, S.D., Katz, J.: Rational secret sharing, revisited. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 229–241. Springer, Heidelberg (2006)

[8] Halpern, J., Pass, R.: Game theory with costly computation (manuscript) (2008)

[9] Halpern, J., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: STOC 2004, pp. 623–632. ACM, New York (2004)

[10] Izmalkov, S., Lepinski, M., Micali, S.: Rational secure computation and ideal mechanism design. In: FOCS 2005, pp. 585–594. IEEE, Los Alamitos (2005)

[11] Kol, G., Naor, M.: Cryptography and game theory: Designing protocols for exchanging information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)

[12] Kol, G., Naor, M.: Games for exchanging information. In: STOC 2008, pp. 423–432. ACM, New York (2008)

[13] Lepinksi, M., Micali, S., shelat, a.: Collusion-free protocols. In: STOC 2005, pp. 543–552. ACM, New York (2005)

[14] Lepinski, M., Micali, S., Peikert, C., shelat, a.: Completely fair SFE and coalition-safe cheap talk. In: PODC 2004, pp. 1–10. ACM, New York (2004)

[15] Lindell, A.Y.: Legally-enforceable fairness in secure two-party computation. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 121–137. Springer, Heidelberg (2008)

[16] Lysyanskaya, A., Triandopoulos, N.: Rationality and adversarial behavior in multi-party computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)

[17] Micali, S., shelat, a.: Purely rational secret sharing (extended abstract). In: TCC 2009. LNCS, vol. 5444, pp. 54–71. Springer, Heidelberg (2009)

[18] Naor, M., Pinkas, B., Sumner, R.: Privacy preserving auctions and mechanism design. In: 1st International Conference on Electronic Commerce, pp. 129–139. ACM, New York (1999)

[19] Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)

[20] Ong, S.J., Parkes, D., Rosen, A., Vadhan, S.: Fairness with an honest minority and a rational majority. In: TCC 2009. LNCS, vol. 5444, pp. 36–53. Springer, Heidelberg (2009)

[21] Parkes, D.C., Rabin, M.O., Shieber, S.M., Thorpe, C.A.: Practical secrecy-preserving, verifiably correct and trustworthy auctions. In: 8th International Conference on Electronic Commerce, pp. 70–81. ACM, New York (2006)