Practical Secrecy-Preserving, Verifiably Correct and Trustworthy Auctions

David C. Parkes *

Harvard University SEAS, Cambridge, MA

Michael O. Rabin

Harvard University SEAS, Cambridge, MA

Stuart M. Shieber

Harvard University SEAS, Cambridge, MA

Christopher Thorpe

Harvard University SEAS, Cambridge, MA

Abstract

We present a practical protocol based on homomorphic cryptography for conducting provably fair sealed-bid auctions. The system preserves the secrecy of the bids, even after the announcement of auction results, while also providing for public verifiability of the correctness and trustworthiness of the outcome. No party, including the auctioneer, receives any information about bids before the auction closes, and no bidder is able to change or repudiate her [1] bid. The system is illustrated through application to first-price, uniform-price and second-price auctions, including multi-item auctions. Empirical results based on an analysis of a prototype demonstrate the practicality of our protocol for real-world applications.

Key words: Auctions, auction theory, cryptographic auctions, cryptography, e-commerce, electronic transactions, homomorphic cryptography, security.

[1] For clarity of reference, we use "she", "her", etc. to refer to the bidders and verifiers, and "he", "his", etc. to refer to the auctioneer (generally a prover), and other parties to the auction.

1 Introduction

In recent years, auctions and electronic marketplaces have been used to facilitate trillions of dollars in trade in the world economy [26]. Auctions, in particular, are often adopted to promote the ideal of competitive pricing and economic efficiency [45,9]. Previously used for rare goods, or for time-sensitive goods (e.g., flowers and fish), auctions can now be harnessed for all kinds of commercial transactions [49]. Auctions see especially wide use for the procurement of goods and services by firms and governments [24,32,77]. We also note that more and more auctions of all kinds are electronic, and operate over the Internet, which reduces the cost of participation and enables worldwide competition.

Individual procurement events in the private sector, for instance, the procurement of truckload services by Procter and Gamble, approach US $1 billion in transaction value [70]. To give a sense of the scale of procurement in the public sector, Asker and Cantillon [5] estimate public procurement in the European Union at about 16% of its GDP; by this estimate public procurement comprised $2 trillion of trade in 2006 [78]. Governments worldwide also use auctions to allocate property rights, such as auctions for wireless spectrum [46] (with worldwide proceeds exceeding US $100 billion by the end of 2001 [49]). In a typical week in February, 2006, the U.S. treasury sells more than US $25 billion in three-month treasury bills through a sealed-bid auction. [2] Sponsored search auctions drive over $1 billion in revenue to Google each quarter [40], and the eBay marketplace reported a record US $44.3 billion volume in the 2005 calendar year, representing a 30% increase over 2004.

Why are auctions so popular? Trepte [77] emphasizes the role auctions play in promoting competition. Competition, in turn, provides incentives for bidders to act as 'honest brokers' of information, so that in the context of procurement the winner is the most technically efficient firm. Yet, auctions are only effective in promoting competition if they are trustworthy, with all bids treated fairly and equally and all bids are *seen to be* treated in this way [77]. In discussing the role of regulation in the context of procurement auctions, Trepte emphasizes the importance of being able to commit to an objective process, so that

> "... the buyer binds himself in such a way that all bidders know that he will not, indeed cannot, change his procedures after observing the bids, even though it may be in his interest to do so."

* *Corresponding author.* Address: 33 Oxford St., Cambridge MA 02138, USA. Tel.: +1-617-384-8130. Fax: +1-617-495-9837.

Email addresses: parkes@seas.harvard.edu (David C. Parkes), rabin@seas.harvard.edu (Michael O. Rabin), shieber@seas.harvard.edu (Stuart M. Shieber), cat@seas.harvard.edu (Christopher Thorpe).

[2] Generally sold in uniform-price auctions. See http://www.publicdebt.treas.gov

Schelling [72] had already noted "... *it is a paradox that the power to constrain an adversary may depend on the power to bind oneself*." In the context of auctions the point is a simple one: the firm engaged in procurement would like to commit not to advantage one firm over another to promote fair competition.

1.1 The Problem of Corruption

Auctions are not immune to corruption and this commitment to a correct process can be hard to achieve. By *corruption*, we mean the auctioneer breaking the rules of the auction in favor of some bidder(s), typically in exchange for bribes [41]. The possibility of corruption exists in an auction whenever the auctioneer is not the owner of the goods for sale in the auction, or the owner of the firm that is seeking to procure goods [41]. For instance, there is a possible conflict of interest when the auction is operated by an individual within a large firm, or by a public servant within a government organization [39].

As evidence of the extent of concern about corruption in competitive processes, the main goal of governments and international bodies such as the World Bank, in regulating public procurement auctions, is to "*curb the discretion*" of the buyer [77]. The World Bank recently estimated the volume of bribes exchanging hands for public sector procurement alone to be roughly US$200 billion per year, with the annual volume of procurement projects 'tainted' by bribes close to US$1.5 trillion [7], and has made the fight against corruption a top priority [22].

When price is the only factor in determining the winner of an auction, then many authors argue that using an open and verifiable, *sealed-bid* auction should help to prevent corruption [77,67,42]. In a sealed bid auction, bids are committed during the bidding process and then opened simultaneously by the auctioneer and the rules correctly followed to determine the winner (and price). However, it seems difficult in practice to ensure a fully trustworthy sealed-bid auction. The kinds of manipulations that are possible in a first-price sealed-bid auction include the following:

- The auctioneer allows a favored bidder to improve on the bid of the winning bidder (possibly the same favored bidder) by revealing information about other submitted bids before the auction closes [42], or by inserting a bid for the favored bidder after reviewing the submitted bids. This allows the favored bidder to win at the best possible price.
- A favored winning or second-place bidder can be invited to change a bid after the auction has closed in order to obtain a better price or win the auction, respectively [48].
- Bribes can can be received before bids are made, in exchange for a promise to modify the bidder's bid to the bidder's advantage should that bidder be the winner [35].

Each of these manipulations relies on the ability to circumvent the intended sealed-bid auction process. The first method relies on learning information before the close of the auction, or being able to insert or modify a bid after some bidders have already bid. The second and third methods rely on being able to change, or cancel, bids after the close of the auction.

More than ethically troubling, corruption is undesirable because it can lead to both an efficiency loss (e.g., with the wrong supplier winning a contract) and also a distributional effect (e.g., with the government paying too much for a contract) [3,22,35,48,19,15,16]. Corruption is a widespread, real-world problem, as illustrated by the following examples:

- A 1988 U.S. investigation, *Operation Ill Wind*, into defense procurement fraud resulted in the conviction of 46 individuals and 6 defense corporations, with fines and penalties totaling US$190 million [15].
- Mafia families in New York City would sometimes pay bribes for an "undertaker's look" at the bids of other bidders before making their own bids when bidding for waste-disposal contracts [35].
- In 1996, Siemens was barred from bidding in public procurement auctions in Singapore for five years because they bribed the chief executive of Singapore's public utility corporation in exchange for information about rival bids [41].
- As many as 40–50 "information brokers" (buying information from oil companies and selling to suppliers) may be actively working at any given point of time in the North Sea oil industry, with corruption and bid rigging affecting upwards of 15% of contracts (an economic value of GB£1.75 billion per year in 1995) [2].

Driving home the difficulty of implementing truly sealed-bid auction processes, Ingraham [31] provides a remarkable account of corruption in New York City *School Construction Authority* (SCA) auctions, an approximately US$1 billion per year market. Two SCA employees and eleven individuals within seven contracting firms were implicated in the corruption. A dishonest contractor would submit a bid well below the projected price of the contract, and during the public announcements of the bids, the auctioneer would save the favored bid until the other bids were opened and announced. Knowing the current low bid, the dishonest auctioneer would then read alound a false bid just below the current low bid instead of the artificial bid actually submitted. The bid form would subsequently be corrected with correction fluid.

Second-price auctions are robust against all three of these manipulations [48]. In a second-price (Vickrey) auction the good is sold to the highest bidder for the second highest bid price [79] (respectively, bought from the lowest bidder for the second lowest bid price in a reverse auction such as a procurement auction.) In a second-price auction no single bidder can be given a special advantage because all bidders have the same opportunity to match other bids via the auction rules. However other (more complicated) manipulations are possible; e.g., the auctioneer can collude

with the two highest bidders, with the second highest bidder invited to *withdraw* her bid upon the auction closing so that the highest bidder wins the auction but has to pay only the third highest bid [42]. [3]

Moreover, without additional assurances, second-price auctions are vulnerable to a new kind of manipulation: when selling an item, an agent acting for the seller can insert a *shill* bid below the highest bid after the close of an auction and drive up revenue. [4]

An almost universal conclusion, across the many papers in the field, is that there is a need for *verifiably correct* and *trustworthy* first-price sealed-bid auctions [42,77,9], with emphasis placed on the need for the process to be *open and transparent*. It is apparent from the above examples that standard solutions, which rely on a well-defined and open process, with bids sealed until opened in public, and the use of regulations and penalties, often remain inadequate. Indeed, Andvig [2] makes the interesting point that even when an organization is *successful* in restricting access to information before an auction closes, then, paradoxically, there are fewer people that know enough to "police" the process and this can lead in turn to *more* opportunities for corruption.

1.2 Our Solution

Our solution ensures the correctness of a sealed-bid auction and allows verifiability of correctness by any third party and without revelation of the bids received. The solution extends to multi-item auctions and includes all popular variants of auction pricing rules, including first-price and second-price. Correctness is ensured by providing complete secrecy of bids until the close of the auction (including, even, from

[3] Moldovanu and Tietzel [50] provide a remarkable account of a failed attempt by the German author Goethe (1749–1832) to use a second-price auction to sell a manuscript. Goethe set a reservation price p and instructed his agent to collect a bid b from Vieweg (1761–1835), the propsective publisher, and to sell at p if and only if $b \geq p$. The story is relevant here because his agent, legal counsel Böttiger, deviated from the rules and revealed to Vieweg the exact amount p. Vieweg subsequently bid p, and Goethe accepted the offer but without realizing his desire, which was to learn about his true "worth" by running this truthful auction.

[4] Seeing problems with implementing truly sealed-bid auctions, one can also consider the role of open auctions in which bids are "broadcast" to all participants; traditionally, this would occur with all bidders in the same room but today an open auction can be conducted over the Internet. Although open auctions may provide transparency and reduce opportunities for manipulation, Lengwiler and Wolfstatter [42] conclude the open auctions may not be desirable for the fear of bidder collusion. Other authors argue that open auctions are often unsuitable for procurement, and other complex environments, because bidders need time to formulate technical proposals [5,41].

the auctioneer), assured revelation of bids to the auctioneer upon auction closing, and verification that the outcome (or the part of the outcome that the auctioneer promises to verify) is correct through the use of cryptographic methods. None of the aforementioned manipulations of first-price or second-price auctions is possible in our scheme.

An important factor in the practicality of cryptographic methods for providing trusted auctions is having a clearly understandable and convincing solution, that is accessible to knowledgable people who are nevertheless not experts on the intricacies of cryptography and general zero knowledge proofs. In this regard, we assume a public key infrastructure under which all parties possess public/secret key pairs for digital signatures and use Pascal Paillier's *homomorphic encryption* [56] scheme to provide verifiable correctness and trustworthiness without revealing information about the bids. The cryptographic proofs are based on universally accepted assumptions.

We focus on two additional aspects of practicality. First, the auction will clear in reasonable time and with reasonable communication requirements using commodity hardware, even for a large number of bidders. Second, the computational architecture must be consistent with practical business models. To achieve this we focus on *proofs of correctness* rather than secure computation. Unlike previous solutions, e.g., Naor et al. [53], we require neither the existence of multiple auctioneers nor that the auctioneers or bidders collaborate to conduct the auction. We believe that a model involving a single auctioneer that is solely responsible for conducting the auction and independent verification of the auction by third parties is more realistic from a business perspective.

We have carefully examined the role of all parties in a sealed-bid auction and formalized their role in cryptographically sound protocol. In addition to a seller, multiple bidders, and an auctioneer, our model employs two commercial entities: *notaries* protect bidders by acting as witnesses to the submission of bids–primarily to prevent the auctioneer from ignoring or modifying submitted bids, and a *Time-Lapse Cryptography Service* [63] provides a cryptographic commitment protocol that prevents bidders from refusing to reveal commitments they make during the auction protocol. The Time-Lapse Cryptography (TLC) Service is used to keep bids secret before the close of the auction. The TLC service publishes a public key before the auction begins, and delays the creation of the corresponding secret decryption key until after the close of the auction.

The auctioneer creates an appropriately certified Paillier public/secret key pair to be used for the security of the bids after the auction, and publishes both this public key and the time-lapse public key in the auction rules. Bidders first encrypt their bids using the auction's public key, then re-encrypt these encrypted bids using the TLC Service's public key, and finally submit the doubly encrypted bids to the auction.

This ensures that the auctioneer will be able to decrypt the bids, but only after the TLC Service's secret decryption key is revealed after the auction closes.

Whereas earlier methods required the auction to be distributed across the computers of multiple, independent auction operators, or required complex interactive protocols involving computation by bidders and the auctioneer, our solution has a simple, non-interactive, and familar computational architecture. Bidders prepare commitments to their bids and send the commitments to the auctioneer and any witnessing notaries. The auctioneer opens the commitments (but can do so only after the auction closes), determines the outcome of the auction and publishes proofs of its correctness. In return for this simplicity, we do not achieve all of the same privacy guarantees as earlier solutions [38,53,30,44].

We choose not to protect against the revelation of bid information by the auctioneer *after* the close of the auction. In our view, while an important area of research, the algorithmic and software methods currently available for solving this problem are too cumbersome and challenging to understand to find wide business applicability. Moreover, we consider this kind of manipulation to be less insiduous because it does not facilitate corruption *during* the auction. No information can be leaked by any party before the auction closes, and after the auction closes no new bids can be introduced and no bids can be altered. We also note that even when bid values stay concealed from the auctioneer at great process complexity cost, a determined adversary can try to spy and obtain information on a rival's bid using corrupt insiders. Thus, an absolute guarantee of secrecy is never attainable in real life.

Complete post auction-closing secrecy can be enforced, in cases where it is deemed essential, by appeal to specialized hardware and monitoring software. A *Trusted Computing* infrastructure, based on secure hardware and digitally signed software (audited by third parties for correctness), installed in physically secure locations with ongoing monitoring and auditing, can prevent the leaking of information with high assurance [73]. In fact, with such deliberately opaque servers it is of the utmost import that an auction participant can independently verify the correctness of the outcome of an auction and be assured that there is no fraud. Thus, such technological methods to eliminate secrecy leaks are very well complemented by our methods for verifiable correctness.

While providing the secrecy of bid information is our primary focus, privacy of bidder identities can be accomplished by other business or cryptographic protocols. For example, bidders may use legal proxies to place bids on their behalf to hide their identity, or the auctioneer may employ a cut-and-choose blinding technique (as described in Section 2.4.5) so that the mapping of winners to bidders is revealed only where necessary by revealing the random blinding factors. [5]

[5] This point becomes important when proving the outcome of the auction; in the protocols we describe, we do not attempt to keep secret that, say, bidder B_3 was the winner, because we assume B_3's true identity is already private if that is necessary.

To demonstrate the scalability of our technology, we have conducted empirical timing tests (Section 5). We show that for acceptable strength of the cryptographic security key, single or multi-item auctions with 100 bidders can be prepared in around two hours of computation and verified in less than half an hour, all on a modest (2.8 GHz Pentium 4) PC. We also show that the computations scale linearly with the number of bidders. Because our method is easily parallelizable, it is possible to accommodate auctions with even tens of thousands of bidders in at most a day of computation on a 64-node network of commodity PC's. Over a decade ago, Franklin and Reiter [27] also found that conducting cryptographic sealed-bid auctions was possible on commodity computing hardware of the day, although their protocol differs substantially from our own.

1.3 Additional Benefits: Better Robustness to Collusion

Providing for verifiable and trustworthy auctions *without* revealing information about bids brings another indirect benefit. A major concern in the use of auctions in practice is that of *bidder collusion* [69]. By collusion we mean bidders coordinating in a *bidding ring*, with the intention to manipulate the final price. The basic idea is to bid jointly in order to limit competition, with the proceeds being shared among members of the ring. [6]

Collusion between bidders is an especially difficult problem to address because it necessarily exploits information asymmetries between the auctioneer and the bidders, and is therefore hard to prevent and detect [77]. Unlike the recommendations of the World Bank and other national and international agencies, our technology allows for auction verification without revealing information about bids, and this provides further robustness against bidding rings.

As evidence of the problems caused by bidder collusion, consider the following examples in first-price sealed-bid auctions:

- Multiple firms were convicted of participating in bidding rings in auctions for school milk contracts in Florida and Texas in the 1970s and 1980s [57].
- Following allegations of bidder collusion at Forest Service timber sales in the Pacific Northwest in the 1970s, an empirical study finds evidence for collusion in auctions conducted between 1975 and 1981 [6].
- In 1984, one of the five biggest highway construction firms in New York state was convicted in federal court of rigging bids in auctions for state highway contracts on Long Island in the early 1980s. Four other firms were listed as unindicted co-conspirators [59].

[6] Porter and Zona [59] note that joint bidding is typically illegal unless the specified work could not be performed without the combined capabilities of the participating firms or if the bidders could not be competitive individually.

First-price auctions are preferred over second-price auctions because they are less susceptible to collusion. [7] In first-price auctions, bidding rings are only sustained by the threat of punishment because members have to submit bids lower than their true value. Bidding rings are unstable without the ability to identify a bidder that deviates and without repeated interaction [66,29,31]. Indeed, Ashenfelter [4] suggests that auction houses such as Sotheby's and Christie's keep the identity of buyers secret to combat rings so that buyers can break from a ring and buy anonymously.

Yet, a common feature in every one of the aforementioned real-world auctions was that the auction was concluded with the *public opening* of bids. As discussed by Porter and Zona [59], this has an unfortunate side effect:

> "The... policy of publicly announcing the bids and the identity of bidders allows cartel members to detect deviations from cartel agreements. Undercutting or cheating would not go unnoticed."

Indeed, the World Bank's own official *Procurement Guidelines* [7] state that,

> "Bids shall be opened in public; bidders or their representatives shall be allowed to be present... The name of the bidder and total amount of each bid, and of any alternative bids if they have been requested or permitted, shall be read aloud (and posted online when electronic bidding is used)..."

Why, one might ask, is bid information made public when it can enable bidding rings to sustain themselves through credible threats of punishment? Trepte [77] makes the reason very clear. While noting the value of "*restricting the detail and content of post-award information*," he adds that "*the existence of such information is essential if disappointed buyers are to be able to challenge unfair or unlawful procurement procedures*." For this reason, we argue that our solution may have important ramifications in terms of reducing opportunities for bidder collusion while addressing corruption. Our auction protocol provides a balance of transparency, trustworthiness and secrecy that reduces the potential for corruption while improving market efficiency.

A related point can be made in the context of using our techniques to verify the correctness of second-price auctions. The main effect, of course, is that we enable a trustworthy and verifiably-correct auction process. This prevents, in particular, any concern about the manipulation through shill bidding discussed earlier. But there is also a second benefit, that comes from not needing to reveal bid values in establishing that the auction process was correctly conducted. Second-price auctions support truthful bidding in a dominant strategy equilibrium, usefully simplifying

[7] In a second-price auction the collusive strategy is for the member of the ring with the highest value to bid high and the rest of the members to bid low, or not at all. This is stable because no member of the ring can do better through a unilateral deviation from the collusive agreement [66].

the bidding process for participants. On the other hand, this bidding strategy can have a number of unpleasant side effects when bids are revealed after the auction closes. In the context of procurement, a supplier will be reluctant to reveal her true cost basis to a competitor [68]. Similarly, when purchasing government assets such as wireless spectrum, a bidder will be recluctant to reveal her true value for acquiring assets to competitors. Governments may also be reluctant to reveal to the public that the value of the highest bid was significantly more than the revenue collected. [8]

1.4 Related Work in Cryptography

Much of the previous work on the use of cryptography for conducting verifiably correct and trustworthy auctions has focused on the goal of *complete privacy*, where not even the auctioneer learns information about bids after the close of the auction [38,53,30]; see Brandt [12] for a recent discussion. This is typically achieved through assuming two or more trusted third parties, either through numerous auctioneers [30] or with asymmetric models in which the commerical entity of an *auction issuer* is assumed in addition to the auctioneer [53,44]. Some protocols achieve complete privacy through bidder-resolved multi-party computation [12]. In comparison, we settle for verifiable correctness and trustworthiness in combination with complete secrecy to all parties except the auctioneer; see also Franklin and Reiter [27], which employs "verifiable signature sharing", requires an electronic cash infrastructure, and distributes this trust in the auctioneer among a set of servers. As discussed above, the auctioneer in our solution cannot learn any information about bids until the auction has closed. In return we achieve a non-interactive [9] protocol that is especially simple from a bidder's perspective.

In justifying the focus on computationally secure methods to provide correct and verifiable auctions, it is interesting to note that achieving information-theoretic guarantees on complete privacy is impossible in a single-item Vickrey auction [13], at least when it is desired that the payment is only revealed to the winning agent. (One cannot prove to another party that the winner's payment was correct without revealing information beyond that implied by the fact that this bidder had the highest bid.)

[8] For example, when the New Zealand government conducted a Vickrey auction for telecommunications licenses, it was revealed after the fact that the winner had been willing to pay much more [46].

[9] Interactive cryptographic auction protocols require the active participation of bidders throughout the auction process in order to obtain the auction results, generally via multi-party computation or related methods. Non-interactive protocols such as ours require no such bidder participation; submission of bids is the only required bidder activity, and bidders' verifications of auction correctness can be performed with no additional interaction with the auctioneer.

For trusted third parties we require only notaries, who provide a lightweight "witness" service and are independent business entities that already exist in practice [74]. The level of trust in them is quite low, as they never possess any non-public information. The Time-Lapse Cryptography Service functions as a trusted third party, although Rabin and Thorpe [63] describe a TLC service that distributes trust among many parties using secret sharing, so that there is no single completely trusted party.

In addition to providing business realism (also see Lipmaa et al. [44] for a critique of earlier methods), we choose to adopt standard methods from homomorphic encryption combined with "cut and choose" test sets and eschew more complex cryptographic ideas such as secure multi-party computation, obfuscation of circuits, and oblivious transfer. As Bradford et al. [11] argue, many such complex protocols, particularly those requiring the ongoing participation of bidders, suffer from "protocol completion incentive problems", in which bidders who know they have lost or change their minds can disrupt the protocol and prevent the completion of an auction. We intentionally avoid such problems by having a single partially trusted auctioneer compute the outcome.

We share with Lipmaa et al. [44] (see also [1,8,12,75,21]) the use of homomorphic encryption, but seek a simpler solution through the use of a single auctioneer in place of the two server model adopted in their work. In their protocol, the seller and an auction authority, who are trusted not to collude, work interactively to generate zero-knowledge proofs of correctness. Cachin [18] proposes a technique based on homomorphic encryption in which a semi-trusted single auctioneer provides a means for two bidders to determine whose bid is higher in zero knowledge (in fact, not even the auctioneer learns the bids). However, his extended protocol for cryptographic auction similarly requires two auction servers which are assumed not to collude. Nakanishi et al. [52] describe a similar protocol based on additively homomorphic encryption and a set of auction servers who conduct a multi-party computation. Such methods result in stronger privacy and secrecy properties at the cost of this additional process complexity.

Rabin, Servedio and Thorpe [62] have recently proposed a somewhat different cryptographic architecture suitable for conducting sealed-bid auctions with similar properties that does not employ homomorphic cryptography. Instead, the system uses a statistically secure encryption scheme based on cryptographic commitments and proves all computations correct to an arbitrarily low probability of error.

Earlier work on multi-item auctions either assumes distributed trust [34,21,1], or adopts multi-party computation techniques [12], and the current state of the art for secure combinatorial auctions is still not very scalable [80,75]. In comparison, our approach can be extended to secrecy-preserving multi-item auctions (presented here) and combinatorial auctions (reserved for future work). Specifically, our trusted auctioneer can apply fast algorithms to the combinatorial optimization

problem in determining winners. The auctioneer must simply construct a *proof* that the outcome is correct and need not involve multiple parties in *computing* the outcome.

Whereas previous architectures use cryptography for anonymity, we note that existing real-world business entities (e.g., notaries as proxy bidders) also meet this need. We therefore do not complicate our protocol with maintaining bidder anonymity and consider it outside the scope of this work. Another practical issue, addressed in previous work but not here, is that of *noncoercibility* [17,74] of an auction. Noncoercibility prevents a bidder from being able to credibly claim to a third party that it bid in a particular way after the close of an auction. Auctions with this property are more resistant to bidding rings, since the stability of bidding rings in first-price auctions depends on being able to detect (and punish) deviations from agreed upon rules.

2 Preliminaries

The standard auction model considers an auctoneer AU, bidders $B = \{B_1, \ldots, B_k\}$, and a seller. This is a *forward* auction in that the goal is to allocate one or more items to some set of bidders. Reverse auctions, with a buyer rather than a seller, are suitable for procurement auctions and can be modeled in a similar way. In a single item auction, each bidder B_i is modeled with a private value v_i; she bids to maximize her net utility (which is v_i - p, her payment, in the event that she wins the auction.) In a first-price, sealed-bid auction, each bidder B_i makes a bid $\mathbf{Bid}_i$. This is a claim about its maximium willingness to pay. Bids are made without any information about the bids (or values) of other bidders, and the item is sold to the highest bidder, who pays the highest bid price. In a second-price sealed-bid auction, the item is sold to the highest bidder, who pays the second highest bid price. [10] See Krishna [37] for an introduction to auction theory.

2.1 Desired Auction Properties

Based on the analysis in the introduction, we list desiderata for any sealed-bid auction process. These go beyond the standard economic goals, for instance, efficiency or revenue maximization:

[10] As noted earlier, although more susceptible to collusive bidding behavior, second-price auctions have the useful property that it is a dominant strategy for a bidder to report her true value.

- Non-repudiation by bidders: Once a bidder submits a bid, her bid is provably unalterable. Moreover, a bidder is bound to reveal her bid to the auctioneer after the auction closing time.
- Non-repudiation by auctioneer: The auctioneer's exclusion of a properly submitted bid can be conclusively proven and thus becomes legally actionable.
- Trustworthiness: The auctioneer cannot know the bids until after the close of the bid submission phase. Thus the auctioneer cannot collude with bidders by sharing others' bids during the auction.
- Secrecy: The bids are hidden to everyone until all bids are committed. At the close of the auction, only the auctioneer knows any secret information. He may keep the outcome secret, notifying only winners of their allocations and payments, or make any part of the outcome public by revealing some or all of the allocations and payments and proving them correct. Revelation of these values does not reveal other secret information not implied by the values themselves.
- Verifiable correctness: All information revealed, whether private or public, is proven correct. Bidders receive a proof of the correctness of their own allocation and payments. The public, including all bidders, receives a proof of correctness for all public information about the outcome of the auction and also the validity of bids. The auction protocol enforces correctness; an auctioneer will not be able to present valid proofs for invalid winners or incorrect payments.

In achieving these properties we make standard cryptographic assumptions. Because the security of our encryption is related to the computational intractability of solving "hard" cryptographic problems, longer cryptographic keys can be adopted over time as computational hardware gets more powerful. This will maintain the same level of realized security at comparable computational running time.

2.2 Real-World Components

We recall that our auction system comprises an auctioneer AU, bidders $B = \{B_1, \ldots, B_k\}$, and a seller. Bidders can also be *proxies* to provide anonymity. In addition, we assume a universally accessible, tamper resistant clock (such as provided by the United States NIST time servers) and the following components.

2.2.1 Certified Bulletin Board

The auctioneer maintains a certified bulletin board. This can be a publicly known website maintained and updated by the auctioneer. The auctioneer uses the bulletin board to post all public information about the auction, including the initial auction announcement as well as (encrypted) information about bids that have been submitted and proofs that can be used to verify all publicly available information about the outcome. All posts to the the bulletin board will carry appropriate digital signatures identifying their originators.

2.2.2 Notaries

Notaries are reputable agents, such as law firms, accountants, or firms specializing in providing a *witness* for bidders. When preparing to participate in an auction, a bidder may select a set of notaries of her choosing from some set of notaries possibly authorized by the auctioneer. Use of the notaries is optional; their only purpose is to prevent a dishonest auctioneer from failing to post bid information from disfavored bidders. In using a notary, whenever a bidder sends concealed bid information to the auctioneer she also sends that concealed information to any notaries she has selected, most notably commitments to bids and random help values. These notaries also submit this information to the auctioneer, and act as witnesses in the case that a bidder complains that an auctioneer does not correctly post her information to the bulletin board. We require that a majority of the notaries is not corruptible. Note that our process is structured so that no information about the actual bids is revealed to the notaries, and, again, their only role is to serve as witnesses to the communications in the auction in case of a dispute between a bidder and the auctioneer.

2.2.3 Time-Lapse Cryptographic Service

A bidder B_i, possibly in collusion with the auctioneer, might refuse to open her commitment and reveal her encrypted bid $E(\mathbf{Bid}_i)$. [11] One way to prevent this practice of *bid repudiation* is to employ a "Time-Lapse Cryptography Service" named and described by Rabin and Thorpe in [63].

The Service will at regular intervals post a new cryptographic public encryption key *TPK* (Time-lapse Public Key), and after a fixed period of time post the associated secret decryption key *TSK* (Time-lapse Secret Key). For our purposes, it suffices that the public key be available before the bids are to be submitted, and that the secret key be released soon after the auction closes. We envision that the Service will publish a constant stream of keys with appropriate lifespans for an auction, and the Auctioneer selects and specifies a key to be used that expires soon after the closing time of the auction. [12] For example, the Service might publish a set of public encryption keys each hour, each with a different lifespan, e.g. three hours, one day,

[11] The notation $E(m)$ designates an *encryption* of a message m; see Section 2.5 for details of the cryptographic notation we employ.

[12] Rivest et al. [65] propose similar methods for cryptography with forced time release where the user sends x to a time-released cryptography service, which returns $E_s(x)$ using a secret key s; s is released at some future date. They also suggest eliminating the trusted third party with threshold secret sharing as well as an "off-line" approach that does not require the involvement of the Service. Other work on identity-based encryption, token-controlled public key encryption, partial key escrow, and timed-release encryption solves this problem in similar ways; see [63] for a review.

one week, 90 days, etc. When the lifespan expires for a particular public encryption key, the Service reconstructs and publishes its associated secret decryption key.

For our purposes, the Service must not employ any single trusted third party who knows either the bid information or any secret key that could decrypt the encrypted bid information before the close of an auction. Additional cryptographic details about the TLC Service are provided in Section 2.4.4.

2.3 Overall Flow and Main Steps of Auction

Schematically, the auction process will proceed in three main stages (described in more detail in Section 3). In the first stage, the auctioneer posts the auction announcement on the bulletin board. The announcement, to be detailed later on, includes a deadline time T for submitting bids. In the second stage, the bidders commit to the encrypted forms of their bids and random data but post bid information in a form that is concealed even from the auctioneer. Notaries are engaged in this stage and witness these commitments posted to the auctioneer's bulletin board. In the final stage, the bidders must follow through and reveal the encrypted forms of their bids to the auctioneer and the public. They do *not* decrypt or reveal their unencrypted bids. The auctioneer and other bidders verify that these encryptions of their bids are consistent with the posted commitments. The auctioneer then decrypts the bids in secret, and computes the outcome of the auction according to the posted rules for that auction. He then posts the parts of the outcome to be verified on the bulletin board, along with public proofs that the selection of the winner(s) and their payments was done according to the auction rules. After the last posting, any party can verify the correctness of the publicly verifiable part of the outcome. A bidder can also privately verify the correctness of her individual outcome via a proof offered by the auctioneer if that outcome is to be kept secret.

2.4 Basic Cryptographic Tools

Our system relies on universally accepted cryptographic tools. We describe the tools we employ in our result, referring to other publications for established results and providing proofs for new uses of existing tools. We will sometimes refer to a "prover" $\mathcal{P}$ and a "verifier" $\mathcal{V}$ when discussing the secrecy-preserving proofs of mathematical facts relating to our auctions. See the *Handbook of Applied Cryptography* [47] for a general introduction to the applied cryptographic techniques and notation we employ.

2.4.1 Public Key Infrastructure

We assume cryptographically sound methods of establishing and exchanging public keys used for all the cryptographic tools we employ, including the auctioneer's public/secret key pair for Paillier encryption and the public and secret keys published by the time-lapse cryptography service. In addition, the auctioneer, notaries, and all bidders require public/secret key pairs for digital signatures. The public signature verification keys of all parties must be mutually known and certified. We notate digital signatures as follows: AU can sign message x, generating $\mathbf{Sign}_{AU}(x)$. A bidder B_i's signature of x is denoted $\mathbf{Sign}_i(x)$.

2.4.2 Sources of Randomness

Cryptographic key generation and probabilistic encryption require a good source of random data. We postulate bidders' and notaries' ability to create enough highly random data to create strong key pairs and encrypt or sign a small number of values. We further postulate that the auctioneer has a source of random data sufficient to encrypt large numbers of integers used in the secrecy-preserving proofs described below. Such a source might be hardware that extracts randomness from radio static or quantum noise in diodes. Such "hardware randomness generators" are already employed in important cryptographic applications.

2.4.3 Secure Random Data

In order to prevent any party (including the auctioneer) from cheating by infusing deliberately slanted random data into the cryptographic protocols, we require that all bidders commit to a random data string when they bid, and that the auctioneer post a commitment to a random data string in the auction rules. These strings are revealed only at the close of the auction, and then combined using exclusive OR so that even if just one of the strings is truly random, the combination thereof is also truly random. We denote this auction random data string by X. The resulting string X is used to "tie the hands" of the auctioneer: when proving the correctness of the auction, the auctioneer must reveal data exactly as specified by the bits in X.

The auctioneer publishes the algorithms to be used on data from the random data string in the auction rules, for example, the method for choosing a random permutation of integers in a specific range, which is employed in the course of proving the auction results correct.

2.4.4 Time-Lapse Cryptography

The Time-Lapse Cryptography Service (introduced in Section 2.2.3) provides for a binding and hiding commitment to bids (so that the bidder may not change her

bid and the auctioneer learns nothing about the bid from its commitment); it also enforces the nonrepudiation of bids, so that once a bidder has committed to her bid, she may not prevent the auctioneer from eventually decrypting it. We assume the TLC service wherever we employ cryptographic commitments in our protocol, and notate bidder B_i's commitment to a value x as the time-lapse encryption $E_{TPK}(x)$.

Each bidder B_i commits to her encrypted bid by encrypting $Z = E_{TPK}(E(\mathbf{Bid}_i))$ (where the bid is first encrypted with the public key of the auctioneer), using a time-lapse public encryption key *TPK*. The bidder then posts $\mathbf{Sign}_i(Z)$ on the bulletin board. After time $T+1$, the decryption key *TSK* associated with *TPK* will be posted by the TLC service. The release of the decryption key *TSK* will enable the auctioneer (and everybody else) to decrypt $Z = E_{TPK}(E(\mathbf{Bid}_i))$ after time $T+1$ and thus obtain $E(\mathbf{Bid}_i)$; this functions as the "decommit" operation – importantly, out of the hands of the bidder.

Where time-lapse encryption of long strings is required, a symmetric block cipher key is created and encrypted using the public TLC key, then published. Data are encrypted using the symmetric key; when the TLC secret decryption key is revealed, the symmetric key can be recovered and the data decrypted. Thus the magnitude of a time-lapse encrypted value x may be polynomial in the size of the TLC key given the assumptions underlying time-lapse cryptography and any suitably secure block cipher. We therefore assume any value in our protocol may be encrypted using a TLC public encryption key.

2.4.5 Cryptographic Blinding

If privacy is to be enforced by the auctioneer, or in cases where it is necessary to keep secret the number of parties allocated any items (Section 4.2.3), the Paillier encryption scheme we use permits the transformation of a known encryption of a value (ciphertext) into another ciphertext so that both decrypt to the same input value (plaintext). To blind a Paillier-encrypted value, say, $E(\mathbf{Bid}_i, r)$, the prover computes a random blinding factor $s \in \mathbb{Z}_n^*$ and computes $s^n \cdot E(\mathbf{Bid}_i, r) \equiv E(\mathbf{Bid}_i, r \cdot s) \pmod{n^2}$. This remains a valid encryption of $\mathbf{Bid}_i$, but only someone who knows s or the secret decryption key ϕ can prove that fact.

Blinding is well-complemented by a "cut-and-choose" protocol so that a prover $\mathcal{P}$ constructs $2v$ random blindings of a set of values, then the verifier $\mathcal{V}$ asks for v of the sets to be revealed by revealing the random blinding factors used to construct them. $\mathcal{V}$ then checks that each blinded set contains exactly the original set of elements. For example, once the posted bids are on the bulletin board, the auctioneer creates a number of blinded auctions, verifies half of them to be correct, and then proves the outcome of the auction on the other half. This keeps the original bidders' identities private.

The computational cost of blinding a ciphertext is almost equivalent to encrypting a plaintext, because the dominant computation is the modular exponentiation required by both operations.

2.5 Homomorphic Cryptography for Secrecy-Preserving Proofs

We employ Paillier's encryption scheme [56] to encrypt auction data. Paillier's is a *homomorphic encryption* system, in which the result of an operation applied to two ciphertexts is a valid encryption of an operation (possibly the same one) applied to their plaintexts. [13] In cryptography, a *plaintext* is the original form of a message, in our case the integer representing a bid or quantity; a *ciphertext* is the encryption of a plaintext.

Homomorphic encryption schemes enable computation with encrypted values without revealing any new information about the values themselves or the results of the computation. Paillier's system employs a public/secret key pair, n and ϕ respectively. The secret key n is the product of two large prime numbers p and q, and its size is determined by the security requirements of the application. The secret key ϕ is $(p-1)(q-1)$. A 1024-bit public encryption key is widely considered sufficient for security until 2010 [28]. Paillier encryption is also a *probabilistic* encryption scheme. In particular, encryptions are performed with a random "help value" r that is used to achieve *semantic security*: given two plaintexts and their encryptions, one cannot tell which ciphertext corresponds to which plaintext without being able to decrypt them. Semantic security is critical to preserve the secrecy of the bids both during their initial encryption and during the verification process, where both bids and the values in the test sets, whose plaintexts are well known, must still remain secret.

The security of this scheme is founded on the "Decisional Composite Residuosity Assumption" (DCRA) [56]. [14] The DCRA implies that if the public key n is difficult to factor, then it is also difficult to tell whether a particular number x is a number of the form $x = r^n \pmod{n^2}$ for some r. This assumption is related to the widely accepted assumptions underlying the security of RSA [64], ElGamal [25], and Rabin [61] encryption, and is believed to be of similar computational intractability.

The Paillier encryption of a message x will typically be denoted $E(x,r)$, where the public key n is implicit and the help value r is made explicit. In discussion below,

[13] More formally, in a homomorphic encryption scheme, there exist operations $\oplus$ and $\otimes$ such that given ciphertexts $C_1 = E(x_1)$ and $C_2 = E(x_2)$, $C_1 \otimes C_2 = E(x_1 \oplus x_2)$. Paillier's encryption scheme is homomorphic in that $E(x_1) \times E(x_2) = E(x_1 + x_2)$. See Appendix A for more details.

[14] A number $x = r^n \pmod{n^2}$ is known as an n^{th} *residue* mod n^2. Because n is a composite number — the product of two primes — x is called a composite residue.

the help value r will sometimes be omitted to simplify notation where it is implicit or irrelevant, for example, $C = E(x)$.

We present here a summary of the properties of, and extensions to, Paillier's scheme we use in this paper. First, given only the encryption $E(x_1)$ and either another encryption $E(x_2)$ or a public constant k, anyone can compute the encryptions $E(x_1 + x_2)$, $E(x_1 + k)$, and $E(x_1 \cdot k)$ *without learning anything about* x_1, x_2, *or the secret key* ϕ. Second, a prover $\mathcal{P}$ who knows the secret key ϕ can also prove a full set of *equality and inequality relations* for two encrypted values $E(x_1)$ and $E(x_2)$, e.g., $x_1 = x_2$, $x_1 > x_2$, etc., again, without revealing anything about x_1 or x_2. It is also possible to compare encrypted bids to constants in a similar way. The last two statements with respect to Paillier's encryption will be proved next. We employ the notation $E(x) \trianglelefteq E(y)$ to mean "$x \leq y$ can be proven using encrypted values $E(x)$ and $E(y)$" and the similar notation $\trianglerighteq$ ($\geq$), $\triangleleft$ ($<$), and $\triangleright$ ($>$). The verification of these comparisons is detailed in Appendix 2.5.1.

2.5.1 Secrecy-Preserving Equality and Inequality Proofs

Equality comparison. Given two ciphertexts $C_1 = E(x_1, r_1)$ and $C_2 = E(x_2, r_2)$, $\mathcal{P}$ can prove $x_1 = x_2$ without revealing any additional information—most importantly, the value of x_1 or x_2. Both $\mathcal{P}$ and $\mathcal{V}$ compute $C' = C_1 \cdot C_2^{-1} \pmod{n^2} = E(x_1 - x_2, r_1/r_2) = E(0, r_1/r_2)$. $\mathcal{P}$ then proves C' is an encryption of zero as above by revealing r_1/r_2.

Inequality comparison. Given two ciphertexts $C_x = E(x)$ and $C_y = E(y)$, $\mathcal{P}$ can show $x > y$ and $x \geq y$. Because our values x and y are integers mod n^2, we can prove $x > y$ by showing $x \geq y + 1$, provided $y \neq n - 1$. Due to the homomorphic properties of Paillier encryption, $E(x+1) = E(x) \cdot (n+1) \pmod{n^2}$, and so adding 1 to a value in its encrypted form is trivial. Thus, all ordering comparisons can be reduced to the ability to prove $x \geq y$. We first specify that x and y must be in the range $[0, 2^t)$ for $2^t < n/2$. This can be proven as described in Section 2.5.3. Then, to prove $x \geq y$, both $\mathcal{P}$ and $\mathcal{V}$ calculate $E(x-y) = E(x) \cdot E(y)^{-1} \pmod{n^2}$, and $\mathcal{P}$ proves $0 \leq (x-y) < 2^t < n/2$ from $E(x-y)$. If in fact $x < y$, then $(x-y)$ will wrap around mod n^2 so that $(x-y) \geq n/2$ and no such proof is possible. This principle is also detailed in Section 2.5.3.

We also mention that zero-knowledge proofs exist to prove, given only two Paillier-encrypted values $E(x)$ and $E(y)$, that $x \neq y$ without revealing whether $x > y$. We omit the construction of such a proof because we do not require it in our protocol.

2.5.2 Secrecy-Preserving Proof of Encrypted Products

Because Paillier encryption does not enable the secrecy-preserving multiplication of two encrypted values as it does addition, we require a method that allows a

prover $\mathcal{P}$ with three plaintexts u, v, and w such that $uv = w \pmod{n}$ to prove this fact to a verifier $\mathcal{V}$ who has Paillier encryptions $E(u)$, $E(v)$, and $E(w)$, respectively. Dåmgard et al. [23] propose another solution to this; the solution we present is in the spirit of our other cryptographic primitives.

Definition 1 *A* Multiplication Test Set *(MTS) for $E(u,r)$, $E(v,s)$, and $E(w,t)$ is a set of 8 elements:*

$$\{E(u_1,r_1), E(u_2,r_2), E(v_1,s_1), E(v_2,s_2),$$
$$E(w_{i,j}) = E(u_i v_j, p_{i,j}) \mid i,j \in \{1,2\}\}$$

where $u = u_1 + u_2 \pmod{n}$ and $v = v_1 + v_2 \pmod{n}$.

In each MTS, u_1 and v_1 are chosen uniformly at random from $\mathbb{Z}_n$; u_2 and v_2 are correspondingly defined, as above, so that $u = u_1 + u_2 \pmod{n}$ and likewise for v.

Clearly, if given encryptions as in *MTS* and

$$w_{1,1} + w_{1,2} + w_{2,1} + w_{2,2} = w \pmod{n} \tag{1}$$

then in fact $uv = w \pmod{n}$. But for $\mathcal{P}$ to prove and for $\mathcal{V}$ to verify all the relationships included in the *MTS*, $\mathcal{P}$ must reveal u_1, u_2, v_1, and v_2, which would consequently reveal u and v. Thus we adopt for an interactive proof the following challenge and partial revelation proof. $\mathcal{P}$ constructs and sends *MTS*. $\mathcal{V}$ randomly chooses a challenge pair (i,j), say, $(1,2)$, and sends it to $\mathcal{P}$. In this case, $\mathcal{P}$ reveals r_1, s_2, and $p_{1,2}$. This allows $\mathcal{V}$ to decrypt $E(u_1)$, $E(v_2)$, and $E(w_{1,2})$, and directly verify that $u_1 \cdot v_2 \equiv w_{1,2} \pmod{n}$. $\mathcal{P}$ further reveals:

$$R = r_1 \cdot r_2 \cdot r^{-1} \pmod{n}$$
$$S = s_1 \cdot s_2 \cdot s^{-1} \pmod{n}$$
$$p = p_{1,1} \cdot p_{1,2} \cdot p_{2,1} \cdot p_{2,2} \cdot t^{-1} \pmod{n}$$

$\mathcal{V}$ by use of R verifies $E(u_1) \cdot E(u_2) \cdot E(u)^{-1} \pmod{n^2} = E(0,R)$, i.e., verifies $u = u_1 + u_2 \pmod{n}$ and similarly $v = v_1 + v_2 \pmod{n}$ via S. Finally, $\mathcal{V}$ verifies $E(w_{1,1}) \cdot E(w_{1,2}) \cdot E(w_{2,1}) \cdot E(w_{2,2}) \cdot t^{-1} \pmod{n^2} = E(0,p)$, thereby verifying that (1) holds.

A moment's thought reveals that if *MTS* was not proper then the probability of $\mathcal{V}$ uncovering this by the random choice of (i,j) is at least $\frac{1}{4}$. Thus the probability of $\mathcal{P}$ meeting the challenge when $uv \neq w \pmod{n}$ is at most $\frac{3}{4}$. This implies that if m *MTS*'s are used and $\mathcal{P}$ meets all m random challenges then the probability of $\mathcal{P}$ cheating is smaller than $(\frac{3}{4})^m$. In practice, the auctioneer will act as $\mathcal{P}$ and verify the multiplications required to prove the validity of multi-item auction allocations by repeating these zero-knowledge proofs until the desired probability of error is achieved.

2.5.3 *Verifiable, Secrecy Preserving Range-of-Values Tests*

In order to prove for two values a and b that $a \geq b$, we can show that $a, b < n/2$ and then that $(a-b) \pmod{n} < n/2$ as described above. [15] This works because if a and b are less than $n/2$ and a is greater than b, then clearly $a - b < n/2$; if a is *less* than b, then $a - b$ will "wrap around" modulo n and must be a large number, that is, $a < b \Rightarrow a - b \pmod{n} > n/2$.

There are many approaches to prove in zero knowledge whether an encrypted value lies in a given range. We have formally specified and implemented a method that is easy to understand based on bit representations of encrypted values (below and Appendix 2.5.1); Damgård et al. [23] and Lipmaa et al. [44] present other similar solutions for Paillier encryption and auctions, respectively. Even more efficient techniques, based on clever number theoretic results (see [14,20,10,33,62]), may offer efficiency gains in future implementations of our protocol.

Thus, with a single additional primitive to prove that $x < n/2$ given only an encryption of x, we can prove inequalities of bids using only their encrypted forms. Given ciphertext $C = E(x,r)$ we want to prove that $x < 2^t$ for some t such that $2^t < n/2$. That is, we we want to be able to verify that a bid $\mathbf{Bid}_i$ is smaller than some agreed upon bound 2^t, without revealing any information about $\mathbf{Bid}_i$. The value of t determines the number of bits of resolution available to bidders in selecting their bids. For our purposes it suffices to take $t = 34$, so that if bids are in units of one thousand dollars, for example, then bids are limited to at most $16 trillion.

We perform the test as follows:

Definition 2 *A valid* test set *TS for the assertion "$C = E(x,r)$ is an encryption of a number $x < 2^t < n/2$" is a set of $2t$ encryptions:*

$$TS = \{G_1 = E(u_1, s_1), \ldots, G_{2t} = E(u_{2t}, s_{2t})\} \tag{2}$$

where each of the powers of 2: $1, 2, \ldots, 2^{t-1}$ appears among the u_i exactly once and the remaining t values u_j are all 0. Each test set's elements are randomly ordered.

By use of a test set TS, the prover $\mathcal{P}$ can prove that $x < 2^t < n/2$ as follows:

Range Protocol. Let $x = 2^{t_1} + \ldots + 2^{t_\ell}$ be the representation of x, a sum of distinct powers of 2. *AU* selects from TS the encryptions $G_{j_1}, \ldots, G_{j_\ell}$ of $2^{t_1}, \ldots, 2^{t_\ell}$, and further $t - \ell$ encryptions $G_{j_{\ell+1}}, \ldots, G_{j_t}$ of 0. Note that:

$$\left(E(x,r)^{-1} \cdot G_{j_1} \cdot \ldots \cdot G_{j_t}\right) \pmod{n^2} = E(0,s) \tag{3}$$

[15] Because our mathematical operations are over the integers modulo a large number, a small negative number is the same as a large positive number, and vice versa. For example, $13 \equiv -2 \pmod{15}$.

is an encryption of 0 with help value $s = \left(r^{-1} \cdot s_{j_1} \cdot \ldots \cdot s_{j_t}\right) \pmod{n}$ if and only if indeed $x = 2^{t_1} + \ldots + 2^{t_\ell}$ and the G_{j_h} were chosen as stated. Now since $\mathcal{P}$ has the decryption key ϕ and thus knows the help value r, then he can hand over to $\mathcal{V}$ the set $\{G_{j_1}, \ldots, G_{j_t}\}$ and the above help value s. $\mathcal{V}$ can now verify on her own that (3) holds and deduce that $x < 2^t < n/2$. $\square$

The above protocol reveals nothing to $\mathcal{V}$ beyond $x < 2^t < n/2$, because TS is a set, in actual implementation a randomly permuted array of the elements in question. Consequently $\mathcal{V}$ has no information about *which* encryptions of powers of 2 are included in $\{G_{j_1}, \ldots, G_{j_t}\}$ Furthermore, the inclusions of $t - \ell$ encryptions of 0 hides even the number of non-zero bits in the binary representation of x. Finally, the random factors $s_{j_1}, \ldots, s_{j_t}$ present in the test set's encryptions combine to a uniformly random s, which completely masks any information about the help value r in the encryption $E(x,r)$. Consequently no information about x is revealed.

There is, however, a problem with the above protocol in that $\mathcal{V}$ does not know that $\mathcal{P}$ has presented her with a true test set. This is overcome as follows. For ease of understanding we first describe an interactive verification protocol, then modify it for non-interactive use. The idea is to use a "cut and choose" procedure in which the prover commits to a number of test sets and allows the verifier to choose and inspect multiple test sets and make sure that they are each valid. Finally, the remaining test sets are all used to complete the proof. An early, possibly the first, use of this idea was presented by Rabin [60].

Tamper Proof Interactive Verification of $x < 2^t < n/2$. First, the prover $\mathcal{P}$ creates $2v$, say for $v = 20$, test sets $TS_1, \ldots, TS_{2v}$, and presents those to $\mathcal{V}$ claiming that they are all valid. Verifier $\mathcal{V}$ randomly selects v test sets $TS_{i_1}, \ldots, TS_{i_v}$ and requests that $\mathcal{P}$ reveal all the encryptions by revealing all the corresponding help values. $\mathcal{V}$ verifies all the encryptions and checks that every TS_{i_h} is valid. If any verification fails, the process is aborted. Otherwise, there now remain v unexamined test sets, call them $TS_{j_1}, \ldots, TS_{j_v}$. $\mathcal{P}$ now completes v repetitions of the above **Range Protocol**, and establishes that $x < 2^t < n/2$ by use of each of the above remaining v test sets. If all verifications succeed then $\mathcal{V}$ accepts that indeed $x < 2^t < n/2$.

The only way that $\mathcal{P}$ can cheat is if all the above remaining v test sets are invalid, which requires that initially the $2v$ test sets comprised v proper test sets and v improper ones and, furthermore, when examining the test sets, $\mathcal{V}$ randomly chose all the v proper ones. The probability of such an unfortuitous choice is $\binom{2v}{v}^{-1}$. In our example of $v = 20$, that probability is, by Sterling's Theorem, about $\sqrt{\frac{20\pi}{2^{40}}} < \frac{8}{10^{12}}$. Thus, we have a zero-knowledge protocol for $\mathcal{V}$ to verify interactively with AU that $x < 2^t < n/2$, when given a ciphertext $E(x,r)$ such that the inequality actually holds.

Tamper Proof Non-Interactive Verification of $x < 2^t < n/2$. We prefer to adopt the following non-interactive method to establish the validity of test sets in our

scheme. In what follows, we adopt the auctioneer AU as the prover. Suppose that there are (as in Section 3.2) $2k$ range-of-values tests to perform. On closing the auction but before receiving information about bids, AU posts $4kv$ test sets on the bulletin board. (For expository convenience, we proceed below with our assumption of $v = 20$.)

Prior to closing, each bidder, the seller (if desired), and the auctioneer are also asked to commit to a random string, which will be revealed after the auction closes and after the auctioneer commits to test sets. Given strings S_i from each bidder, S_S from the seller, and S_{AU} from the auctioneer, the strings are XORed together to generate $X = S_1 \oplus S_2 \oplus \cdots \oplus S_k \oplus S_S \oplus S_{AU}$. Note that even if only one of the participants chooses his string randomly and independently, then X is a truly random string.

The $80k$ test sets posted on the Bulletin Board are then segmented into $2k$ groups of 40 test sets each, i.e., the first 40 test sets, the next 40 test sets, etc. The random bit-string X is then used, in combination with a fixed rule available to all participants and posted at the start of the auction to the bulletin board, to select 20 test sets from each group. This random selection replaces the random selection by the verifier $\mathcal{V}$ employed in the interactive proof and allows the proof to work without interaction.

2.5.4 Bulk Verification of Test Sets

Because in practice an auction will require large numbers of test sets, we may accelerate the non-interactive verification process by verifying all the test sets to be used for an auction *en masse*, which requires a smaller percentage of the test sets be revealed and thereby made unusable.

We have already shown how AU can use a test set to prove both that for any encrypted bids $E(\mathbf{Bid}_1)$ and $E(\mathbf{Bid}_2)$, $\{\mathbf{Bid}_1, \mathbf{Bid}_2\} \leq 2^t$ and $\mathbf{Bid}_1 > \mathbf{Bid}_2$, provided $2^t < n/2$. However, the verifier $\mathcal{V}$ needs to know that the test set AU uses to prove this is correctly constructed in order to believe the proof.

In a traditional zero-knowledge proof (ZKP) setting, AU would present $\mathcal{V}$ with several test sets in a "cut-and-choose" protocol, and $\mathcal{V}$ would then select at $\mathcal{V}$'s own discretion some of the testsets for AU to reveal. In our setting, it is impractical for AU to perform real-time ZKP's of bid correctness to all of the verifiers. Therefore, we employ a technique where instead of the verifier choosing the test sets to reveal, we derive randomness from the test sets themselves and use that randomness to define both which test sets will be revealed, and the order in which other test sets will be used to verify bids. This means that AU can *publish* a ZKP of the correctness of the test sets that anyone can verify. This can even be done asynchronously, i.e. the test sets used to prove an auction correct can be verified correct before an auction closes.

In single-item auctions with B bidders, AU will verify B bids and $B-1$ comparisons to prove the correctness of the auction. These auctions require $2B-1$ proofs.

We observe that all of the test sets will be of identical form for such an auction. Each test set will contain t encryptions of powers of 2: $2^0,\ldots,2^{t-1}$, and t encryptions of 0. For visual comfort, we will use examples where $t = 32$, accommodating bids in a range of over 4 billion values. Because any bid or comparison of bids can be verified using such a test set, we will prepare a single very large collection of test sets that will be used for all comparisons in an auction.

We demonstrate with very high probability that for collections of sufficient size, after revealing 20% of the collection, no more than 10% of the remaining unrevealed test sets are improper. Assuming we draw from the remaining test sets uniformly at random, the probability of a correctness proof of s succeeding, i.e., all s sets are improper is $< 10^{-s}$.

If we select and reveal 500 test sets uniformly at random in a collection of 2500, the probability that all 500 will be correct and 200 (or more) of the remaining 2000 are incorrect is $< 7 \times 10^{-19}$. We can then prove correctness of each bid or comparison with probability of error $< 10^{-10}$ by drawing 10 of the remaining 2000 test sets uniformly at random and proving correctness on each of them. These numbers are appropriate for an auction with 100 bidders and moderate security requirements.

We can achieve a reasonable "random" ordering from the test sets using the random data string X constructed from the XOR of the values $S_1,\ldots,S_k$ from the bidders, S_{AU} from the auctioneer, and optionally S_S from the seller. We will call R some predefined substring of X of suitable length for this purpose.

Step 1. AU privately creates 2500 test sets $TS_i, i \in [0,2499]$, each of which is comprised of encryptions of 64 small values, $\{c_{i0},\ldots,c_{i63}\} = \{E(0) \times 32, E(2^0),\ldots,E(2^{31})\}$. AU creates a secret random permutation $\pi_i(0\ldots63) \in \{0\ldots63\}$ for each TS_i for each of the encrypted values in the test set and privately stores the plaintexts, random help values r and exponentiations thereof $r^n \pmod{n^2}$.

Step 2. AU creates a permutation $\rho(0\ldots2499) \in \{0\ldots2499\}$ of an ordering of the 2500 test sets using the random data in R according to the protocol published at the beginning of the auction.

Step 4. AU reveals the first 500 test sets defined by the ordering ρ. Verifiers will be given a reasonable specified time (depending on the size and complexity of the auction) to verify the correctness of these test sets, after which the test sets will be deemed correct if no objections are raised with AU or the notaries. If a test set is discovered to be invalid, the AU creates 2500 new test sets and the protocol is begun anew at Step 1.

Step 5. If all 500 test sets are correct, then ρ (excluding the revealed test sets) defines the random ordering of the unrevealed test sets that are used to prove each bid.

Once the bids have been published on the bulletin board by the auctioneer in a strict ordering where AU claims, w.l.o.g., $\forall i < j, \mathbf{Bid}_i \geq \mathbf{Bid}_j$, then each bid $\mathbf{Bid}_i, 1 \leq i \leq B$ is verified for correctness by the next 10 unused test sets in the collection in the order defined by ρ. Following that, the comparisons $\mathbf{Bid}_i \geq \mathbf{Bid}_{i+1}$ are proven, again by using each successive set of 10 unused test sets from the ordering defined by ρ.

3 Single-Item Auctions

Given the above cryptographic tools we can formulate a single-item auction succinctly. We assume that the bidders $B_1, \ldots, B_k$ are known entities with publicly known digital signatures $\mathbf{Sign}_i$. We further assume that the winner and her payment depend only on the ordering of the values of the bids and that the payment is one of the bids.

This class of auctions includes first-price and second-price auctions, and also allows for auctions with reservation prices by a simple extension in which the seller also submits a bid. [16] Thus, this class also includes *revenue-maximizing* auctions, as described in Myerson [51], in symmetric environments in which all bidders are assumed to have independent private values drawn at uniform from the same distribution.

For clarity, we focus here on an auction in which the complete outcome of the auction—the winner and the payment by the winner—is made public and then proved to be correct. The same techniques can be used to selectively prove part of the outcome to some party, for instance to prove the winner but not the winner's payment is correct.

[16] In a Vickrey auction with a reservation price, in addition to bids $\mathbf{Bid}_1, \ldots, \mathbf{Bid}_k$ there is a price rp_s from the seller which is handled just as any other bid. The item is sold to the highest bidder if the maximal bid is at least rp_i but goes unsold otherwise. (Think of this as "selling back to the seller".) When sold, the payment is the maximal value of the second highest bid and the reservation price. Note that because the seller must commit to her reservation price just like any other bidder there is no danger of shill bidding.

Step 1. *AU* posts the following information on the bulletin board: the terms of the auction specifying the item, the mechanism for selection of the winner, the deadline T, an identifier *ID* of the auction, and a Paillier encryption key n. *AU* knows the corresponding decryption key ϕ. The auctioneer also posts information about any notaries that are to be used for the auction. He posts the time-lapse encryption key *TPK* to be used by all participants in constructing their commitments. Finally, the auctioneer posts a commitment to his random string $\mathbf{Com}_{AU}(S_{AU})$ and a specification of the method that will be used to extract random permutations from the auction's random data string X.footnote We recall from Section 2.4.3 the random strings S_i XORed together to yield the auction random data X. *AU* must specify here the method used to extract a permutation of test sets from X before *AU* sees X so that everyone knows *AU* is revealing a truly random selection of test sets.

We emphasize that all of the above data D_{AU} is posted on the bulletin board, accompanied by *AU*'s signature $\mathbf{Sign}_{AU}(D_{AU})$.

Step 2. Every B_i chooses a bid $\mathbf{Bid}_i$. She encrypts it as $C_i = E(\mathbf{Bid}_i, r_i)$ using the public key n and a randomly chosen help value r_i. In order to create efficient test sets to prove bid sizes, we restrict the size of the bid so that $\mathbf{Bid}_i < 2^t < n/2$ for small t, say, $t = 34$. Every B_i also generates a random bit string S_i of appropriate length which will be used in the proof of correctness. Bidder B_i then commits to C_i and S_i by encrypting with E_{TPK} to form a single commitment string $\mathbf{Com}_i = E_{TPK}([C_i, S_i, ID])$, which also includes the auction identifier *ID*. Finally, the bidder signs this commitment, and sends $\mathbf{Sign}_i(\mathbf{Com}_i)$ to *AU* and her notaries, if used, before time T. *AU* returns a signed receipt $R_i = \mathbf{Sign}_{AU}([\mathbf{Com}_i, ID, T])$.

Note that hiding of the encrypted bids and of the random strings by use of the secondary encryption prevents anyone from gaining any knowledge of the data prior to time T. In particular, neither the notaries nor the auctioneer have any meaningful information.

Step 3. At time T, the *AU* posts all the received commitments $\mathbf{Com}_1, \ldots, \mathbf{Com}_k$ on the bulletin board, as well as a random bit string S_{AU}. *AU* also creates a number of test sets $TS_1, TS_2, \ldots, TS_K$, where K is a multiple of k, e.g., $K = 80k$. He signs and posts the test sets on the bulletin board.

Step 4. Between time T and $T+1$ any Bidder B_i who has a receipt R_i for a bid which is not posted, can appeal her non-inclusion, resorting to her notaries if she has used them.

Step 5. After time $T+1$, everyone, including the auctioneer *AU* and all bidders B_i, can recover all encrypted bids $C_i = E(\mathbf{Bid}_i, r_i)$ as well as all random strings S_i. This is done by employing the decryption key *TSK* posted by the TLC service to decrypt

all the commitments posted in Step 2. After time $T+1$, AU posts the encrypted bids, $C_1,\ldots,C_k$, and the random strings, $S_1,\ldots,S_k,S_{AU}$, on the bulletin board. Every bidder B_i can verify, for any bidder B_j, that the posted value $\mathbf{Com}_j$ corresponds to the ciphertext C_j and the random data string S_j. In case of discrepancies she protests. This check can be performed simply by decoding the commitments as above and verifying the digital signatures on these commitments. Every interested party constructs the auction's random data string X by combining the published strings: $X = S_1 \oplus \ldots \oplus S_k \oplus S_{AU}$.

Step 6. Using the decryption key ϕ, AU recovers the bids $\mathbf{Bid}_1,\ldots,\mathbf{Bid}_k$ for computing the auction results and associated random help values $r_1,\ldots,r_k$ for constructing the proofs of correctness. [17] The auctioneer then computes the winner of the auction and the payment according to the auction rules. The auctioneer posts the winner's identity B_i and information defining the payment to be made by the winner on the bulletin board. This information about payment can be posted in an encrypted form if the payment is to be kept secret from nonwinning bidders. Finally, and most importantly, the auctioneer also posts information that will enable any party to verify that the correct result was implemented. These include proofs of the correctness of the winner and payment, and proofs of the validity of each bid.

3.2 Verification

We now show how any verifier $\mathcal{V}$ (including any of the bidders) can verify on her own that the winner and payment of the auction were determined according to the rules of the auction. This will be done in a "zero knowledge" fashion, that is, without revealing anything about the value of any bid except that implied by the outcome of the auction. In addition, the auctioneer can choose how much of the outcome is revealed. For example, the proof can validate that an encrypted payment was correctly determined but without revealing any information about the value of the payment.

The class of single-item auctions under consideration (including first-price and second-price auctions) has the property that the winner and payment depend only on the *ordering of the bids*. Take as an example the Vickrey auction and assume, without loss of generality, that the prices posted by bidders $B_1,\ldots,B_k$ are monotonically decreasing (though there may be tied bids). AU announces that B_1 is the winning bidder, which is tantamount to the following set of claims:

$$\{\mathbf{Bid}_1 > \mathbf{Bid}_2;\ \mathbf{Bid}_2 \geq \mathbf{Bid}_3;\ \ldots;\mathbf{Bid}_2 \geq \mathbf{Bid}_k\} \tag{4}$$

[17] See Appendix A.3.1 for details of the decryption.

Note that the encrypted values

$$\{C_1,\ldots,C_k\} = \{E(\mathbf{Bid}_1, r_1),\ldots,E(\mathbf{Bid}_k, r_k)\}, \tag{5}$$

were posted in Step 5 of the protocol. To prove the claims, it suffices to show that each C_i is an encryption of a valid bid $0 \leq \mathbf{Bid}_i < 2^t < n/2$ for all i, and that

$$\{C_1 \rhd C_2, C_2 \trianglerighteq C_3, \ldots, C_2 \trianglerighteq C_k\} \tag{6}$$

Verifier $\mathcal{V}$ verifies these $2k-1$ claims in a zero knowledge fashion using the tools described above, which enables verification of the winner, item allocation, and payment as described in the following paragraphs.

Recall that the auctioneer had posted $2k$ groups of 40 test sets in Step 3. He creates proofs for each of the first k claims using k of these groups of 40 test sets, one for each claim. He reveals all encryptions for the subgroup of 20 test sets determined by the random string X and the random method posted in Step 1 of the auction. With each of the 20 other test sets AU performs the computation described in Section 15 (**Range Protocol**) and posts it on the bulletin board. $\mathcal{V}$ can verify that all the revealed test sets are valid, that their indices were chosen correctly, and that the k posted computations are of the form (3). This verifies the first k claims. In addition, AU posts proofs for the $k-1$ claims that $\mathbf{Bid}_1 > \mathbf{Bid}_2$ and $\mathbf{Bid}_2 \geq \mathbf{Bid}_i, 2 < i \leq k$ by using $k-1$ groups of 40 additional test sets for each inequality using the methods described in Section 2.5.1.

This ordering of bids is used to verify the winner as the bidder with identity corresponding to submitted bid $E(\mathbf{Bid}_1)$, and the item is allocated to this bidder. In a Vickrey auction, the payment to be made by the winner is $\mathbf{Bid}_2$ and this can be proved by sending a verifier $\mathcal{V}$ the random help value r_2 from B_2's encrypted bid $C_2 = E(\mathbf{Bid}_2, r_2)$. $\mathcal{V}$ can then verify the correctness of its payment by re-encrypting $\mathbf{Bid}_2$ with r_2 and checking the result is C_2.

In the case of a tie, where $\mathbf{Bid}_1 = \mathbf{Bid}_2$, this can also be proven using a zero-knowledge equality proof. (Indeed, the auctioneer would not be able to prove $E(\mathbf{Bid}_1) \rhd E(\mathbf{Bid}_2)$.) Tiebreaking in the single item case is done according to the auction rules, either by conducting another auction or randomly selecting a winner using the auction random data string X according to rules defined at the beginning of the auction.

3.3 Verifying Partial Infomation about Outcomes

As mentioned in the introduction, there exist many examples in which the public disclosure of the bids or outcome of an auction is undesirable. Because there are a number of factors that play a role in determining which data are to be revealed

at the close of the auction, our system provides the flexibility for the auctioneer to prove specific facts about the bids or outcome of the auction to only the individuals who need to know, without revealing anything more.

Many real-world auctions reveal such partial information, perhaps most notably the U.S. Treasury auctions for U.S. public debt, where only partial information about the bids is revealed. In that case the reputation of the Treasury provides the trust necessary for them not to disclose complete auction information, but where such a reputable auctioneer or seller is not involved, our correctness proofs provide the trust necessary to conduct such an opaque auction.

The flexibility of our system comes from the architecture of our correctness proofs; a verifier computes mathematical operations on public values posted to the bulletin board (the bidders' encrypted bids, random strings, and auction rules); the auctioneer then reveals a small amount of special data to the verifier that they compare to their calculations to verify the proof. This allows the auctioneer to control exactly who gets a correctness proof of any fact by private revelation of that special data. We illustrate below the power of our approach by examples of various partial information the auctioneer might reveal about bids and payments.

Bids. At one extreme, the auctioneer can reveal all bids to the public by revealing the random help values used to encrypt the bids. At the other, the auctioneer need not reveal any bid to any bidder to prove the payments correct. Yet there may be legal or auction theoretic reasons to provide "partial transparency" of bids. Due to the nature of the homomorphic cryptosystem employed, the auctioneer can reveal interesting partial information about the bids that can be computed using linear functions of the bid values. For example, the auctioneer might wish to reveal only the mean bid – equivalent to the sum of all bids, assuming the number of bids is public. He does this by revealing the random help value required to decrypt the product of all encrypted bids (which is an encryption of the sum of all bids). The auctioneer could also reveal other interesting statistics, such as the maximum and minimum bid, the median bid, the mean of the bids excluding the highest and lowest bid, or even the standard deviation of bids.

Payments. The auctioneer may also prove winners' payments correct in a public or private fashion. For example, instead of revealing winners' payments to everyone, each bidder can act as her own verifier. She computes an encryption of her payment on her own, and then decrypts it with the auctioneer's help. The auctioneer can privately reveal just the payments – without the bids – to both the sellers and the winners, and prove to all bidders who did not win that their bid was not high enough to win. Thus the seller and every bidder are satisfied that the auction was conducted fairly, yet no information about the outcome of the auction needs to be publicized. Further transparency can be provided by requesting bidders "sign off" on their proven outcomes with a digital signature, so that the auctioneer can

show that every bidder accepted the outcome. If a bidder refuses, the auctioneer can prove the outcome he provided was indeed correct by publicly revealing it.

4 Multi-Item Auctions

Consider now auctions for multiple identical items. In these auctions, the auctioneer has some number l of available identical items for sale. Real-life examples include large lots of refurbished items on eBay, or U.S. Treasury bills. We consider auctions in which bids are *flexibile* and each bidder is willing to accept *any number of items up to a maximal limit and bid a price per item*. However, there is nothing about the framework that is limited in this way, and we will describe extensions to "all-or-nothing" bids and "bid curves" [71,36] in future work.

As before, we can implement a general class of auctions that includes the first-price, uniform-price, and second-price (generalized Vickrey) auctions [37]. These are auctions in which the allocation depends only on the order of the bids and payments are defined as linear functions of the values of bids. For illustrative purposes we again focus on the case in which the complete outcome of the auction, i.e. the allocation and all payments, is made public and then proved to be correct. Easy variants are available in which the correctness is selectively proved, either publicly for some restricted information about the outcome or privately to individual bidders.

4.1 Protocol

Step 1. *AU* posts the auction information on the bulletin board as in Section 3.1. In addition, *AU* posts the total number of items available, l, and the maximum allocation to any one bidder (if any), $l_{\max}$.

Step 2. Each participating bidder B_i prepares two integer values $(\mathbf{Bid}_i, \mathbf{Qty}_i)$ for each bid she wishes to submit to the auction, where $\mathbf{Bid}_i$ is the amount that she will pay per item and $\mathbf{Qty}_i$ is the maximum number of items desired by B_i.

As above, B_i also generates a random bit string S_i and sends it to AU. B_i then encrypts $\mathbf{Bid}_i$ and $\mathbf{Qty}_i$, using AU's public Paillier key n, as $E(\mathbf{Bid}_i)$ and $E(\mathbf{Qty}_i)$ and commits by sending AU and her notaries, if used, the commitment

$$\mathbf{Com}_i = [E_{TPK}(E(\mathbf{Bid}_i)), E_{TPK}(E(\mathbf{Qty}_i)), E_{TPK}(S_i), ID], \tag{7}$$

and digital signature $\mathbf{Sign}_i(\mathbf{Com}_i)$. AU issues a receipt for these commitments and publishes them on the bulletin board in accordance with our standard protocol.

Step 3. As above, at time T, the auctioneer AU posts received commitments, his random string S_{AU}, and test sets on the bulletin board. The number of test sets will depend on the type of the auction and the payment calculation; these numbers are detailed in Section 5.

Step 4. As above, bidders have between time T and $T+1$ to appeal non-inclusion, which may involve resorting to the commitments sent to any notaries.

Step 5. As above, bidders' encrypted bids and quantities $E(\mathbf{Bid}_i)$ and $E(\mathbf{Qty}_i)$, as well as their strings S_i, are revealed between time T and $T+1$. AU publishes these values on the bulletin board. All bidders can check that the revealed values correspond with earlier commitments.

Step 6. AU privately recovers bids $\mathbf{Bid}_1,\ldots,\mathbf{Bid}_k$ and quantities $\mathbf{Qty}_1,\ldots,\mathbf{Qty}_k$ using secret key ϕ, and uses the information to compute the correct outcome of the auction. We again assume, without loss of generality, that the prices bid by bidders $B_1,\ldots,B_k$ are monotonically decreasing, though consecutive bids may be tied. We then choose the *threshold bid index*, α, which is new in our multi-item setting, such that bidders $\alpha,\ldots,B_k$ do not receive any items. The sum of the quantities associated with winning bids $\mathbf{Bid}_1,\ldots,\mathbf{Bid}_{\alpha-1}$ is greater than or equal to the number of available items l, and this is not true for a smaller threshold index. Thus all bidders B_i, such that $i<\alpha$, are winners. The threshold winner $\alpha-1$ may receive some subset of her total demand. Formally, threshold index α is defined so that:

$$\left[\sum_{i=1}^{\alpha-2}\mathbf{Qty}_i < l\right] \wedge \left[\sum_{i=1}^{\alpha-1}\mathbf{Qty}_i \geq l\right] \tag{8}$$

Note that we have assumed here that there are enough bidders to cover all of the supply. This can be handled without loss of generality, by also introducing a single dummy bid at zero price for all supply, l. In addition to determining α, and thus the winners in the auction, AU also posts proofs of which bidders won and their allocations on the bulletin board, as well as proofs of the validity of each bidder's bid and quantity. He also computes proofs of correctness of each winner B_i's payment. If public verification of payments is required, AU posts these correctness proofs on the bulletin board, along with the random help values needed to decrypt the payments. If the payments are to remain secret, he privately sends the proof for B_i's payment and any associated random help values to each winner B_i.

4.2 Verification

The verification step in a multi-item auction is more complex than for the single item auction, but relies largely on the same cryptographic primitives used in the

simpler single-item case. Each verification can be done in a zero knowledge fashion, revealing no information beyond that implied by the outcome of the auction.

As before, *AU* first publicly proves the minimum *bid-ordering information*, that all winning bids are strictly greater than the threshold bid $\mathbf{Bid}_\alpha$, i.e., $\mathbf{Bid}_i > \mathbf{Bid}_{\alpha-1}$ for all $i < \alpha - 1$ and $\mathbf{Bid}_{\alpha-1} > \mathbf{Bid}_j$ for all $j \geq \alpha$. This reveals only minimum public information about the value of the bids; the same information that is implied by the outcome. *AU* will also prove that the bid values are valid and without wraparound. (See Section 2.5.3 for an explanation of wraparound.)

In addition, *AU* must also prove that the *quantities* of the items were encrypted correctly, i.e., without wraparound. We assume that $l < 2^t < n/2$ for number of available items l and test set size parameter t. *AU* first proves that no bidder has submitted a quantity greater than a specified maximum allowed allocation $l_{\max} \leq l$. To do this, *AU* first encrypts $E(l,1)$ and $E(l_{\max},1)$; a help value 1 is used so that anyone can verify those encryptions. *AU* then proves $E(\mathbf{Qty}_i) \trianglelefteq E(l_{\max},1)$ for all $1 \leq i \leq k$. Next, *AU* can use encryptions of various sums of quantities to prove the correctness of the threshold bid index α. Paillier's homomorphic encryption system allows for a zero-knowledge proof that a ciphertext represents the encrypted value of the sum of two encrypted values; in particular, $\prod_{i=1}^{\alpha-2} E(\mathbf{Qty}_i) = E(\sum_{i=1}^{\alpha-2} \mathbf{Qty}_i)$. Given this, *AU* can establish Eq. 8 over the encrypted quantities:

$$\left[E(\sum_{i=1}^{\alpha-2} \mathbf{Qty}_i) \lhd E(l)\right] \wedge \left[E(\sum_{i=1}^{\alpha-1} \mathbf{Qty}_i) \trianglerighteq E(l)\right] \tag{9}$$

4.2.1 Tiebreaking

In the event of a tie, with multiple bids equal in value to $\mathbf{Bid}_{\alpha-1}$, the auctioneer must also prove equality of these bid values and then establish correctness in allocating to these tied threshold bidders. Various algorithms exist for allocating the items among winners with equal bids at the threshold. One possibility is to randomly order the threshold bidders and divide the items among them in "round robin" fashion until the items are exhausted, with the condition that no bidder B_i is entitled to more than the $\mathbf{Qty}_i$ items bid for.

In this case, we require additional proofs that the allocation is fair. In summary, we use the random data X jointly constructed by all auction participants to define a publicly verifiable ordering π of w equal bidders, [18] $\pi(1 \ldots w) \in \{1 \ldots k\}$ such that $B_{\pi(1)}$ is the first to be allocated an item, and so forth, and prove the round robin allocation as follows. We notate l_i as the allocation to bidder B_i.

Step 1. Prove that the allocations to all bidders add to l, i.e. $\sum_{i=1}^{k} l_i = l$.

[18] Generating such a random ordering is described in Section 2.4.3.

Step 2. Given ordering π of threshold bidders, compute j such that $B_{\pi(j)}$ is the first bidder in the ordering to receive a partial allocation. Compute h such that $B_{\pi(h)}$ is the first bidder in the ordering to receive $l_{\pi(j)} - 1$ items, i.e. the next bidder in line when the items ran out. If no such h exists, set $h = w + 1$.

Step 3. Prove that all allocations were fair as follows:
3a. For $1 \leq i < j$, prove $l_{\pi(i)} = \mathbf{Qty}_{\pi(i)}$ and $l_{\pi(i)} < l_{\pi(j)}$.
3b. For $j < i < h$, prove either that $l_{\pi(i)} = l_{\pi(j)}$, or both $l_{\pi(i)} = \mathbf{Qty}_{\pi(i)}$ and $l_{\pi(i)} < l_{\pi(j)}$.
3c. For $h \leq i \leq w$, prove that $l_{\pi(i)} = (l_{\pi(j)} - 1)$, or both $l_{\pi(i)} = \mathbf{Qty}_{\pi(i)}$ and $l_{\pi(i)} < l_{\pi(j)}$.

In words, we show that bidders either received their entire allocation or at most one fewer than the first bidder in line to receive a partial allocation, and that the ordering of the partial allocations is proper.

4.2.2 Payment

In a *first-price* auction, the auctioneer can prove a payment to a third party by revealing the random help value used to encrypt winner B_1's bid. A verifier can use this to recover $\mathbf{Bid}_1$ from the now public encrypted value $E(\mathbf{Bid}_1)$ submitted by the bidder. Similarly, in a *uniform-price* auction, whereby every bidder pays the bid price of the losing threshold bidder $B_{\alpha-1}$, *AU* can provide a public proof by revealing $\mathbf{Bid}_{\alpha-1}$ via the help value used by $B_{\alpha-1}$. The uniform price auction is an approximation to a Vickrey auction in this setting. It generates the same payment as in the Vickrey auction to winning bidders $i < \alpha - 1$, as long as the threshold bidder has enough spare demand to cover the allocated capacity of any winner. The payment by the threshold winner $B_{\alpha-1}$ is always larger than in the Vickrey scheme.

We turn our attention to proving the correctness of prices in a generalized Vickrey auction (GVA) for this multi-item setting [37]. As in the single item setting, the GVA provides the useful property of truthfulness so that each bidder's dominant strategy is to bid her true value per unit and true quantity demanded. In a GVA mechanism the number of items are allocated according to the price bid but the actual payment for each winner depends on others' bids. The Vickrey payment for bidder B_i is defined as:

$$p_{\text{vcg},i} = \mathbf{Qty}_i^* \cdot \mathbf{Bid}_i - [V(B) - V(B_{-i})], \tag{10}$$

where $V(B)$ is the total revenue in the auction with all bidders, $V(B_{-i})$ is the total revenue in the marginal economy with bidder B_i removed, and $\mathbf{Qty}_i^*$ denotes the quantity allocated to bidder i in the auction. This has a simple interpretation: a bidder's payment is determined as *the greatest amount other (displaced) bidders would have paid for the same items had B_i not been participating in the auction.*

We require a proof to establish the correctness of this payment. Let $\mathbf{Qty}_j^{-i}$ denote the quantity awarded to bidder B_j in the marginal auction without bidder B_i. For a non-marginal winner, i.e., $i < \alpha - 1$, her GVA payment is:

$$\mathbf{Qty}_i^* \cdot \mathbf{Bid}_i - \left[\mathbf{Qty}_i^* \cdot \mathbf{Bid}_i + \sum_{j \neq i, j \leq \alpha-1} \mathbf{Qty}_j^* \cdot \mathbf{Bid}_j\right] + \sum_{j \neq i, j \leq \beta_{-i}-1} \mathbf{Qty}_j^{-i} \cdot \mathbf{Bid}_j$$
$$= \left[\sum_{\alpha-1 < j \leq \beta_{-i}-1} \mathbf{Qty}_j^{-i} \cdot \mathbf{Bid}_j\right] + [\mathbf{Qty}_{\alpha-1}^{-i} \cdot \mathbf{Bid}_{\alpha-1} - \mathbf{Qty}_{\alpha-1}^* \cdot \mathbf{Bid}_{\alpha-1}] \quad (11)$$

For the marginal winner, $i = \alpha - 1$, her GVA payment is:

$$\mathbf{Qty}_i^* \cdot \mathbf{Bid}_i - [\mathbf{Qty}_i^* \cdot \mathbf{Bid}_i + \sum_{j \neq i, j < \alpha-1} \mathbf{Qty}_j^* \cdot \mathbf{Bid}_j] \quad + \sum_{j \neq i, j \leq \beta_{-i}-1} \mathbf{Qty}_j^{-i} \cdot \mathbf{Bid}_j$$
$$= \sum_{\alpha-1 < j \leq \beta_{-i}-1} \mathbf{Qty}_j^{-i} \cdot \mathbf{Bid}_j \quad (12)$$

Thus, the GVA payment by bidder B_i is a linear combination of the product of the bid price and allocated quantity to bidders displaced by bidder B_i from the winning allocation. In the case of a non-marginal bidder, this computation also accounts for the effect on the allocation to bidder $\alpha - 1$.

Consider the following verifiable proof structure for the term $\sum_{\alpha-1 < j \leq \beta_{-i}-1} \mathbf{Qty}_j^{-i} \cdot \mathbf{Bid}_j$ that is common to both kinds of winners:

Step 1. In generating the proof, *AU* must first establish a bid ordering for the marginal auction without B_i, i.e., prove that β_{-i} is the correct threshold bid index by showing $\mathbf{Bid}_j > \mathbf{Bid}_{\beta_{-i}-1}$ for $j \neq i, j < \beta_{-i} - 1$ and $\mathbf{Bid}_{\beta_{-i}-1} > \mathbf{Bid}_j$ for $j \geq \beta_{-i}$; this can be done as in the main auction. Second, *AU* must prove that bidder $\beta_{-i} - 1$ is the threshold winner in this auction, by proving the analog to Eq. 8. Third, *AU* must publish encrypted values $\mathbf{Pay}_j = \mathbf{Qty}_j \cdot \mathbf{Bid}_j$ for all $j > \alpha_i, j < \beta_{-i} - 1$ (and similarly for the new marginal bidder, $\mathbf{Pay}_{\beta_{-i}-1} = \mathbf{Qty}_{\beta_{-i}-1}^{-i} \cdot \mathbf{Bid}_{\beta_{-i}-1}$), and prove the correctness of all of these ciphertexts. This requires proofs of *correct multiplication*, as described in Appendix A. The proof of $\mathbf{Pay}_{\beta_{-i}-1}$ in turn requires a proof of the quantity allocated $\mathbf{Qty}_{\beta_{-i}-1}^{-i}$ to this bidder, via a proof that a published ciphertext is the encrypted value of $l - \sum_{j \neq i, j < \beta_{-i}-1} \mathbf{Qty}_j$. Fourth, *AU* must publish the encrypted value of the sum of these payments and a proof of its correctness.

Step 2. A verifier $\mathcal{V}$ can independently compute the encrypted Vickrey payment as above and check the correctness of the proof.

Step 3. *AU* reveals the random help value in the resulting encrypted Vickrey payment to $\mathcal{V}$, who decrypts using that value and verifies it is correct by re-encryption.

The verifier $\mathcal{V}$ now knows that B_i's Vickrey payment is correct while knowing (almost) nothing more about any bidder's bid value than can be derived from the definition of Vickrey payments. In fact, the verifier $\mathcal{V}$ learns the *number* of bids required to compute a Vickrey payment in the marginal economy $\mathbf{E}(B_{-i})$. We can get around this through padding the input using dummy bids as described in the next section.

The additional term, $[\mathbf{Qty}^{-i}_{\alpha-1} \cdot \mathbf{Bid}_{\alpha-1} - \mathbf{Qty}^{*}_{\alpha-1} \cdot \mathbf{Bid}_{\alpha-1}]$ can be determined in the case that bidder i is the threshold winner and $i = \alpha - 1$ in an analogous fashion. Encrypted values of the allocation quantities received by bidder i in the main auction and in the marginal auction, i.e., $\mathbf{Qty}^{*}_{\alpha-1}$ and $\mathbf{Qty}^{-i}_{\alpha-1}$, can be established via subtraction from total items l of the total allocation to other bidders. Then, a ciphertext for the difference, $\mathbf{Qty}^{-i}_{\alpha-1} - \mathbf{Qty}^{*}_{\alpha-1}$, and then the product $(\mathbf{Qty}^{-i}_{\alpha-1} - \mathbf{Qty}^{*}_{\alpha-1})\mathbf{Bid}_{\alpha-1}$ can be published and proved.

4.2.3 Secrecy-Preserving Payment Proofs

While our above methods are correct, secure, and efficient in practice, they reveal a slight amount of additional information than that implied solely by the GVA payments. In particular, the method described to prove a GVA payment reveals the number of bidders whose bids in the marginal economy determine a bidder's price. This section outlines a more involved solution that eliminates the revelation of that information at some increased cost in complexity and computation.

We recall that the GVA payment for bidder B_i is defined as:

$$p_{\text{vcg},i} = \mathbf{Qty}^{*}_{i} \cdot \mathbf{Bid}_i - [V(B) - V(B_{-i})], \tag{13}$$

where $V(B)$ is the total revenue in the auction with all bidders, $V(B_{-i})$ as the total revenue in the marginal economy with bidder B_i removed, and $\mathbf{Qty}^{*}_{i}$ denotes the quantity allocated to bidder i in the auction.

In order to prove the correctness of term $[V(B) - V(B_{-i})]$ we currently determine the threshold bidder β_{-i} in the marginal economy $(B \setminus i)$. Recall that the threshold bidder β_{-i} is defined so that all bids $< \beta_{-i} - 1$ receive a full allocation, bid $\beta_{-i} - 1$ may receive a partial allocation, and bids $\geq \beta_{-i}$ receive no allocation. But establishing the index of threshold bidder β_{-i} reveals information beyond that implied either by knowledge of the outcome of the auction or by the amount of an agent's GVA payment, specifically information about the number of bidders that were displaced by the presence of bid B_i.

To solve the problem we introduce a technique to prove the correctness of an encrpyted term $V(B_{-i})$ without revealing any information about the number of winners in that marginal economy. This term can be used in combination with a proof of the

correctness of term $V(B)$ and $\mathbf{Qty}_i^* \cdot \mathbf{Bid}_i$ to prove correctness for the GVA payment to bidder i.

To illustrate the idea we consider the case of proving correctness of the encrypted value of $V(B)$ for the main economy without revealing the index of the threshold bidder. Note also that a dummy bidder is included with bid 0 and quantity demanded l (the supply of items) when the total demand is less than l. Let k denote the total number of bids in the input, including this dummy bidder when required.

In order to hide the true index of the threshold bidder the idea is to pad the input with an additional $k-1$ bids such that threshold index α given the padded input *is always defined so that* $\alpha - 1 = k$. Let γ denote the threshold index given the original input of k bids. The new bids are defined as follows: there are $k-\gamma+1$ bids defined with $\mathbf{Qty}_j = 0$ and $\mathbf{Bid}_j = V$ for a maximal value V (higher than any posted bid), and $\gamma - 2$ bids defined with $\mathbf{Qty}_j = 0$ and $\mathbf{Bid}_j = 0$.

For example, if $k = 5$ then when $\gamma = 2$ (and only the first bid receives an allocation) then all $k-1$ new bids have $\mathbf{Qty}_j = 0$ and $\mathbf{Bid}_j = V$. On the other hand, when $\gamma = 6$ (and all bids receive some allocation) then all $k-1$ new bids have $\mathbf{Qty}_j = 0$ and $\mathbf{Bid}_j = 0$.

Lemma 1 *The threshold index of the padded input is equal to* $k+1$ *and no information is learned about the threshold index in the initial index.*

Moreover, the introduction of this padded input does not change $V(B)$ because the new padded bids demand no quantity and thus contribute nothing to the revenue of the auctioneer.

One problem remains with this solution: how do we ensure that the auctioneer can be trusted to introduce dummy bids with this property without revealing to the verifier the mixture of high value and zero value bids introduced? The verifier must not be able to tell whether a bid in the padded input is a dummy bid or an original bid, but still be confident that the auctioneer has provided a set of bids that contains exactly the posted bids and quantities along with correct padding.

For this we can again use the idea of "cut and choose":

Step 1. The prover constructs $2v$ test sets $TS_1, \ldots, TS_{2v}$. Each test set contains several bid collections. [19] TS_i contains k collections of $2k-1$ bids, one collection for each of $m \in \{0, \ldots, k-1\}$ where there are m high value bids with quantity zero, the k original bids and quantities, and $k-1-m$ low value bids with quantity zero. Each element of the collection is encrypted using the semantically secure Paillier scheme used elsewhere. Instead of re-encrypting the original bids and quantities,

[19] For clarity, we use the two words "collection" and "set", though there is no technical meaning differentiating the two terms.

the auctioneer uses a blinding procedure (see Section 2.4.5) to yield an encryption of the same value that cannot be identified as such. The $2k-1$ elements of each collection are permuted randomly; the k collections within each test set are also permuted randomly. Now, each test set contains k collections of $2k-1$ bids, where each collection is the original auction's bids padded with dummy bids and zero quantities. These test sets are posted.

Step 2. The verifier randomly selects v test sets, requests that the prover identify each of the elements in every collection as either a high value, low value, or posted bid, and prove that fact by revealing the random help values (for dummy bids and their quantities) or the random blinding factor s and the original bid or quantity (for blinded posted values). If there is a problem then the procedure is aborted and the verifier requests a new list of $2v$ fresh test sets.

Step 3. There remain v unexamined test sets. The prover will use each on these to construct a proof as follows: for each test set, the prover identifies one of the collections of bids within that test set, and then completes the proof of the value for $V(B)$ using the padded input with that collection of bids. Not only must the payment be the same for each padded input but the threshold index, given the padded input, must always be $k+1$. As before the value v may be selected to provide a desired probability of error in the outcome.

Because we do not want to reveal which bid is in position α given marginal economy $(B \setminus i)$ and thus do not compute (and prove) the total revenue from bids $\{\alpha, \ldots, \alpha_{-i}-1\}$ we prove instead the total value of $V(B_{-i})$ in this new approach rather than establishing directly the loss in revenue as a result of bid $\mathbf{Bid}_i$ directly, as in the previous section.

This approach can also be used to prove to each bidder the correctness of her allocation without revealing the number of winners, and similarly to prove to any third party the correctness of any single bidder's allocation. As described in Section 3.3, this may have special importance in auctions in which it is desirable for partial information about the outcome to be privately proven to some parties; for example, it may be desirable for the outcome to be secret while the seller, each bidder, and perhaps a third-party auditor still receive a proof that the auction outcome is correct.

4.3 Extensions

We assume each bidder submits only one bid/quantity pair, but a single bidder could simply submit multiple bids in order to represent a more complex utility function. The auction will have the correct behavior when used with first-price or uniform-price payment schemes. For example, a bidder might wish to purchase 10 units if the price is \$50, but 30 units if the price is \$40. By placing two bids,

$(\$50, 10), (\$40, 20)$, the bidder will receive, for example, 30 units if the threshold for winning bids is less than \$40, 10 units if the threshold is between \$40 and \$50. While this "additive-or" bidding logic does not permit bidders to specify completely arbitrary utility functions, it does provide additional expressivity. Note, though, that if this language is used in an auction with GVA payments the bidder's payment could be too high. The logic of GVA requires removing *both* of its (\$50,10) and (\$40,20) bids when computing its payment, but this would not automatically happen when considering these as separate bids. Extensions to correctly handle GVA payments with more expressive languages [54], as well as methods to adopt more expressive languages in which bidders can submit a set of bids with explicit logical dependencies, are reserved for future work.

5 Empirical Results

We implemented Paillier encryption and test set verification in C++ using the LiDIA number theory package [43] on a commodity Linux workstation with a Pentium 4 2.8 GHz processor.

The greatest computational cost in our protocol is the construction and verification of test sets, and in particular the exponentiation of random help values (r^n) required to encrypt or (verifiably) decrypt a value. These calculations dominate all other computation; for example, to sort one million random 64-bit bids takes less than one second on our system. In a single-item auction, the auctioneer can prepare for an auction of 100 bidders in about two hours, and each verifier can independently verify the auctioneer's proofs of correctness in less than half an hour. Both preparation and verification scale linearly and are easily parallelized. Thus, with modest distributed computation, even a multi-item auction with ten thousand bidders can be prepared in a few hours and verified in reasonable time.

We present data for both 1024- and 2048-bit symmetric public encryption keys, which are considered safe until 2010 and 2030, respectively [28]. Because the lifetime of a security key is based on the difficulty of breaking it on available computing power, we claim that, for the most part, an auction with "5-year" security at any point in time will take about the same amount of time as it does today, as improvements in computing power for breaking keys are likely to be comparable to those in encryption. [20]

Table 1 shows the time it takes to compute various cryptographic operations on our test machine. We observe that the time required to prepare or verify a test set is

[20] Of course, if efficient algorithms to solve the composite residuosity problem or factor large composites are discovered, our claim does not hold.

essentially that required by the encryption and decryption. All test sets represent 2^{34} discrete values.

Table 1
Time to perform basic operations

Operation	**Time** (s.) (1024-bit)	**Time** (s.) (2048-bit)
Computation of r^n	0.045	0.287
Encryption	0.045	0.287
Decryption with r	0.045	0.287
Decryption with ϕ	0.014	0.089
Decryption with r^n	0.000	0.001
Constructing a TS	3.01	19.32
Verifying a TS	3.00	19.30
Proving $0 \leq x < 2^t$ given TS	0.001	0.001
Verifying proof of $0 \leq x < 2^t$	0.070	0.41

For a single item auction of k bidders, the auctioneer must produce k proofs of valid bids (i.e. $\mathbf{Bid}_i < 2^t$ for small t; we use 34), and $k-1$ proofs of comparisons to prove the ordering of the outcome. Using the bulk verification method suggested in Appendix 2.5.4, such an auction requires $10 \cdot (2k-1)$ test sets, plus 25% for the test sets that will be revealed to prove the test sets are valid. This gives us an upper bound of $25k$ test sets required to conduct a trustworthy single-item auction.

For a multi-item auction with payments based on one bid (e.g. first-price or second-price), we need only add to the above k proofs $\mathbf{Qty}_i < 2^t$, k comparisons $\mathbf{Qty}_i < l_{\max}$, and 2 comparisons to prove Equation 8. This means we need about double the number of test sets, $4k+1$, to conduct such a multi-item auction; about $50k$ test sets are needed for trustworthiness. We list the time taken to prepare these test sets and correctness proofs in Table 2.

For verified GVA payments in multi-item auctions (Section 4.2.2), we also require proofs of multiplications for at most $2k+1$ products, namely, $\leq k$ proofs of the products $\mathbf{Qty}_i \cdot \mathbf{Bid}_i$ and $k+1$ proofs of the products of the partial allocation to the threshold bidder for the main economy $\mathbf{E}(B)$ and up to k marginal economies (that is, excluding bidder B_i) $\mathbf{E}(B_{-i})$. Each proof of a product requires 4 exponentiations for creating the MTS ("multiplication test set") and 6 exponentiations to verify it. To achieve a reasonably small probability of error, we need to repeat the multiplication proof 80 times ($\frac{3}{4}^{80} \approx 10^{-10}$). Thus each proof requires 320 exponentiations to create and 480 to verify. Table 3 shows time required, again on a P4 2.8 GHz processor, to verify Vickrey payments in the worst case for various sizes of multi-item

Table 2
Time to prepare and verify auctions

Operation	**Number of Bids**		
	100	1000	10000
	Single-item Auctions		
Preparation (1024-bit)	2.1 hr	21 hr	8.7 days
Verification (1024-bit)	25 min	4.2 hr	42 hr
Preparation (2048-bit)	13.4 hr	5.6 days	56 days
Verification (2048-bit)	2.7 hr	27 hr	11 days
	Multi-item Auctions		
Preparation (1024-bit)	4.2 hr	42 hr	17.5 days
Verification (1024-bit)	52 min	8.7 hr	3.6 days
Preparation (2048-bit)	27 hr	11.2 days	112 days
Verification (2048-bit)	5.4 hr	54 hr	22 days

auctions. These computations are required *in addition* to the above computations for verifying prices and quantities.

Table 3
Verification of Vickrey payments for multi-item auctions

Operation	**Number of Bids**		
	100	1000	10000
Preparation (1024-bit)	48 min	8 hr	3.3 days
Verification (1024-bit)	72 min	12 hr	5 days
Preparation (2048-bit)	5.1 hr	51 hr	21 days
Verification (2048-bit)	7.7 hr	77 hr	32 days

6 Conclusions and Future Work

We have presented a new protocol for sealed-bid auctions that guarantees trust and preserves a high level of secrecy, yet is practical enough to run efficiently on commodity hardware and be accepted in the business community. Because we focus on proofs of correctness and secrecy during the auction, an auctioneer can still compute optimal results efficiently and publish efficiently verifiable proofs of those results. Our protocol rests on sound cryptographic foundations, and lends itself to interesting extensions to further types of auctions, including support for all-or-nothing bids, bid curves, and full combinatorial auctions; we intend to pursue

these extensions in later work. We believe that our practical, easily implemented approach can be extended to other areas of privacy, including electronic transactions, trading systems, privacy-preserving open outcry markets, and zero-knowledge public verification of private data. Along these lines, authors Thorpe and Parkes have recently extended our methods to a continuous double auction setting for information hiding in securities exchanges [76].

To further explore the practicality of our solution, David Austin has built a prototype of our protocol. His Python implementation comprises a fully functional, cross-platform web server and standalone client for creating, bidding on, and verifying sealed-bid auctions. Because it is implemented in Python, it runs at approximately half the speed of our empirical tests, which were conducted using optimized C++, but is still fast enough for practical use.

Other future work includes improving the efficiency of our protocols. Due to the dominance of range proofs in auctions, employing more efficient techniques to prove an encrypted value in a particular range are likely to reduce the computation required to prove and auction correct (see Section 2.5.3, and [14,20,10,33,62] cited there). Use of specialized cryptographic hardware for performing modular exponentiation of very large integers instead of standard 32- and 64-bit hardware may also yield significant time savings. Finally, it may be that for many auctions, the auction data need only be secure *during* the auction, and not for years later, and thus shorter cryptographic keys might be employed at a significant savings in computational cost.

While we focus in this paper on auctions in which price is the only consideration, non-price factors such as technical quality, terms of payment, and service agreements, are of course also important in auctions used for procurement. However the effect can be to make the rules of the auction "soft" and provide new opportunities for corruption, since the auctioneer has new flexibility to manipulate the outcome of the auction in return for a bribe [67,15]. Of course, the use of cryptographic methods to verify the correct outcome of an auction requires *objective* criteria for determining the outcome based on the bids. It is useful, then, that concerns about corruption have led the World Bank and other bodies to move towards requiring *quantifiable decision making*, with the revelant "scoring" criteria published as part of the rules of the auction [77,5,7]. This makes quality assessment objective and reduces the corruption concerns to those of bid rigging in price-based sealed-bid auctions. As such, it is of significant interest in future work to develop provably correct, and trustworthy auctions by appropriate extensions to our technology. We also plan to study the use of similar technology in cryptographic *open-bid* settings, such as ascending price and combinatorial clock [58] auctions.

7 Acknowledgments

This work was supported in part by an Alfred P. Sloan Foundation award to Parkes and NSF grant CCR-0205423 to Rabin. The authors thank Peter Coles, Michael Hamburg, Alex Healy, Adam Juda, Andrew Pimlott, Pai-Ling Yin and the anonymous reviewers for helpful comments and information.

References

[1] M. Abe and K. Suzuki. M+1-st price auction using homomorphic encryption. In *Proc. Public Key Cryptography*, 2002.

[2] J. C. Andvig. Corruption in the North Seal oil industry: Issues and assessments. *Crime, Law & Social Change*, 23:289–313, 1995.

[3] L. Arozamena and F. Weinschelbaum. The effect of corruption on bidding behavior in first-price auctions. Technical report, Universidad de San Andres, 2004.

[4] O. Ashenfelter. How auctions work for wine and art. *Journal of Economic Perspectives*, 3:23–36, 1989.

[5] J. Asker and E. Cantillon. Properties of scoring auctions. Technical report, Leonard N. Stern School of Business, 2006.

[6] L. Baldwin, R. C. Marshall, and J.-F. Richard. Bidder collusion at forest service timber sales. *The Journal of Political Economy*, 105:657–699, 1997.

[7] T. W. Bank. *Guidelines Procurement Under IBRD Loans and IDA Credits*. The International Bank for Reconstruction and Development, The World Bank, Washington, D.C., 2006.

[8] O. Baudron and J. Stern. Non-interactive private auctions. In *Proc. Financial Cryptography*, 2001.

[9] T. Börgers and E. van Damme. Auction theory for auction design. In M. C. W. Janssen, editor, *Auctioning Public Assets: Analysis and Alternatives*, chapter 1. Cambridge University Press, 2004.

[10] F. Boudot. Efficient proofs that a committed number lies in an interval. In *Proc. Eurocrypt 2000*, volume LNCS 1807, page 431. Springer, 2000.

[11] P. G. Bradford, S. Park, and M. H. Rothkopf. Protocol completion incentive problems in cryptographic Vickrey auctions. Technical Report RRR 3-2004, Rutgers Center for Operations Research (RUTCOR), 2004.

[12] F. Brandt. How to obtain full privacy in auctions. Technical report, Carnegie Mellon University, 2005.

[13] F. Brandt and T. Sandholm. (Im)possibility of unconditionally privacy-preserving auctions. In *Proc. 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems*, pages 810–817, 2004.

[14] E. Brickell, D. Chaum, I. Damgård, and J. Van de Graaf. Gradual and verifiable release of a secret. In *Proceedings of CRYPTO'87*, volume LNCS 293, pages 156–166, 1988.

[15] R. Burgeut and Y.-K. Che. Competitive procurement with corruption. *The RAND Journal of Economics*, pages 50–68, 2004.

[16] R. Burguet and M. Perry. Bribery and favoritism by auctioneers in sealed-bid auctions. Technical report, Institute of Economic Analysis, UAB, Barcelona, 2003.

[17] M. Burmester, E. Magkos, and V. Chrissikopoulos. Uncoercible e-bidding games. *Electronic Commerce Research*, 4:113–125, 2004.

[18] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proc. 6th ACM Conf. on Computer and Comm. Security*, pages 120–127, 1999.

[19] M. Celantani and J. J. Ganuza. Corruption and competition in procurement. *European Economic Review*, 46:1273–1303, 2002.

[20] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *Proceedings of EUROCRYPT'98*, volume LNCS 1403, pages 561–575, 1998.

[21] X. Chen, K. Kim, and B. Lee. Receipt-free electronic auction schemes using homomorphic encryption. In *ICISC*, 2003.

[22] O. Compte, A. Lambert-Mogiliansky, and T. Verdier. Corruption and competition in procurement auctions. *The RAND Journal of Economics*, 36:1–15, 2005.

[23] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Proc. Public Key Cryptography 01*, 2001.

[24] N. Dimitri, G. Piga, and G. Spagnolo, editors. *Handbook of Procurement – Theory and Practice for Managers*. Cambridge University Press, 2006.

[25] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, IT-31(4):469–472, 1985.

[26] W. J. Elmaghraby. Pricing and auctions in emarketplaces. In *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. Kluwer Academic Publishers, Norwell, MA, 2004.

[27] M. K. Franklin and M. K. Reiter. The design and implementation of a secure auction server. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1996.

[28] D. Giry and P. Bulens. Cryptographic key length recommendation. `http://www.keylength.com`, 2006.

[29] D. A. Graham and R. C. Marshall. Collusive bidder behavior at single-object second price and english auctions. *Journal of Political Economy*, 95:1217–1239, 1987.

[30] M. Harkavy, J. D. Tygar, and H. Kikuchi. Electronic auctions with private bids. In *Proc. Third USENIX Workshop on Electronic Commerce*, pages 61–74, 1998.

[31] A. T. Ingraham. A test for collusion between a bidder and an auctioneer in sealed-bid auctions. *Contributions to Economic Analysis and Policy*, 4:1–32, 2005.

[32] M. C. W. Janssen, editor. *Auctioning Public Assets: Analysis and Alternatives*. Cambridge University Press, 2004.

[33] M. J. Jurik. *Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols*. PhD thesis, University of Århus, 2003.

[34] H. Kikuchi. (m+1)st price auction protocol. In *Proc. Financial Cryptography*, 2001.

[35] S. A. Koc and W. S. Neilson. Bribing the auctioneer in first-price sealed-bid auctions. Technical report, Kocaeli University, 2006.

[36] A. Kothari, D. C. Parkes, and S. Suri. Approximately-strategyproof and tractable multi-unit auctions. *Decision Support Systems*, 39:105–121, 2005.

[37] V. Krishna. *Auction Theory*. Academic Press, 2002.

[38] M. Kumar and S. I. Feldman. Internet auctions. In *Proc. 3rd USENIX Workshop on Electronic Commerce*, 1998.

[39] J.-J. Laffont and J. Tirole. Auction design and favoritism. *International Journal of Industrial Organization*, 9:9–42, 1991.

[40] S. Lahaie. An analysis of alternative slot auction designs for sponsored search. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, 2006.

[41] Y. Lengwiler and E. Wolfstetter. Bid rigging. an analysis of corruption in auctions. Technical report, Humboldt-University at Berlin, 2005.

[42] Y. Lengwiler and E. Wolfstetter. Corruption in procurement auctions. In N. Dimitri, G. Piga, and G. Spagnolo, editors, *Handbook of Procurement – Theory and Practice for Managers*, chapter 16. Cambridge University Press, 2006.

[43] LiDIA-Group. LiDIA — a library for computational number theory. *TU Darmstadt*, 2001.

[44] H. Lipmaa, N. Asokan, and V. Niemi. Secure Vickrey auctions without threshold trust. In *Proc. 6th International Conference on Financial Cryptography (FC 2002)*, pages 87–101, 2002.

[45] P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

[46] J. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8:145–162, 1994.

[47] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001.

[48] F. M. Menezes and P. K. Monteiro. Corruption and auctions. Technical report, Getulio Vargas Foundation, Rio de Janeiro, Brazil, 2001.

[49] P. Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

[50] B. Moldovanu and M. Tietzel. Goethe's second-price auction. *Journal of Political Economy*, 106:854–859, 1998.

[51] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[52] T. Nakanishi, D. Yamamoto, and Y. Sugiyama. Sealed-bid auctions with efficient bids, 2003.

[53] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proc. First ACM Conf. on Elec. Commerce*, pages 129–139, 1999.

[54] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. Cambridge University Press, 2006.

[55] P. Paillier. *Cryptographie à Clé Publique Basée sur la Résiduosité de Degré Composite*. PhD thesis, École Nationale Supérieure des Télécommunications, 1999.

[56] P. Paillier. Public-key cryptosystems based on composite residuosity classes. In *Proc. EUROCRYPT '99*, pages 223–239, 1999.

[57] M. Pesendorfer. A study of colllusion in first-price auctions. *The Review of Economic Studies*, 67:381–411, 2000.

[58] D. Porter, S. Rassenti, A. Roopnarine, and V. Smith. Combinatorial auction design. *Proceedings of the National Academy of Sciences*, 100(19):11153–11157, 2003.

[59] R. H. Porter and D. J. Zona. Detection of bid-rigging in procurement auctions. *Journal of Political Economy*, pages 518–538, 1993.

[60] M. O. Rabin. Digitalized signatures. In *Foundations of Secure Computing*, pages 155–166. Academic Press, New York, 1978.

[61] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[62] M. O. Rabin, R. A. Servedio, and C. Thorpe. Highly efficient secrecy-preserving proofs of correctness of computations and applications. In *Proc. IEEE Symposium on Logic in Computer Science*, 2007.

[63] M. O. Rabin and C. Thorpe. Time-lapse cryptography. Technical Report TR-22-06, Harvard University School of Engineering and Computer Science, 2006.

[64] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Technical Report MIT/LCS/TM-82, MIT Laboratory for Computer Science, 1977.

[65] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed release crypto. Technical Report MIT/LCS/TR-684, MIT, 1996.

[66] M. S. Robinson. Collusion and the choice of auction. *Rand J. Econ.*, 16:141–145, 1985.

[67] S. Rose-Ackerman. The economics of corruption. *Journal of Public Economics*, 4:187–203, 1975.

[68] M. H. Rothkopf, T. J. Teisberg, and E. P. Kahn. Why are Vickrey auctions rare? *Journal of Political Economy*, 98:94–109, 1990.

[69] T. C. Salmon. Preventing collusion between firms in auctions. In M. C. W. Janssen, editor, *Auctioning Public Assets: Analysis and Alternatives*, chapter 3. Cambridge University Press, 2004.

[70] T. Sandholm, D. Levine, M. Concordia, P. Martyn, R. Hughes, J. Jacobes, and D. Begg. Changing the game in strategic sourcing at Procter & Gamble: Expressive competition enabled by optimization. *Interfaces*, 36(1):55–68, 2006.

[71] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2002.

[72] T. C. Schelling. *The Strategy of Conflict*. Harvard University Press, 1980.

[73] S. W. Smith. *Trusted Computing Platforms: Design and Applications*. Springer, New York, 2005.

[74] S. G. Stubblebine and P. F. Syverson. Fair on-line auctions without special trusted parties. In *Proc. of Financial Cryptography*, 1999.

[75] K. Suzuki and M. Yokoo. Secure generalized Vickrey auction using homomorphic encryption. In *Proc. Financial Cryptography*, 2003.

[76] C. Thorpe and D. C. Parkes. Cryptographic securities exchanges. In *Proc. Financial Cryptography and Data Security*, 2007.

[77] P. Trepte. *Regulating Procurement*. Oxford University Press, 2004.

[78] U.S. Central Intelligence Agency. The World Factbook: European Union. `https://www.cia.gov/library/publications/the-world-factbook/print/ee.html`, 2007.

[79] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[80] M. Yokoo and K. Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proc. First Int. Conf. on Autonomous Agents and Multiagent Systems*, 2002.

A Paillier Encryption

A.1 Public/Secret Keys

Paillier encryption uses an encryption key $n = p \cdot q$, where p and q are large primes. The decryption key is based on the factorization of n, $\phi = \varphi(n) = (p-1) \cdot (q-1)$. We recall that $\varphi(n)$ is Euler's totient function, the number of integers relatively prime to n. It is also required that n is relatively prime to ϕ.

A.2 Encryption

To encrypt a plaintext x, first compute a random value r from the range $[1, n-1]$ such that $\mathbf{gcd}(r,n) = 1$, then recall that $(1+n)^x \equiv (1+xn) \pmod{n^2}$ and encrypt as

$$E(x,r) = (1+xn) \cdot r^n \pmod{n^2} \tag{A.1}$$

A.3 Decryption

To decrypt $C = E(x,r)$, given decryption key $\phi = (p-1)(q-1)$, observe that $r^{n\cdot\phi} \equiv 1 \pmod{n^2}$ by Euler's Totient Theorem, and

$$\begin{aligned}
C^\phi &= (1+n)^{x\cdot\phi} r^{n\cdot\phi} \pmod{n^2} \\
&= \left(\tbinom{x\cdot\phi}{0} n^0 + \tbinom{x\cdot\phi}{1} n^1 + \tbinom{x\cdot\phi}{2} n^2 + \ldots\right) && \pmod{n^2} \\
&= 1 + x\phi n + \alpha n^2 + \ldots && \pmod{n^2} \\
&= 1 + x\phi n \pmod{n^2} \\
&\quad \text{implying} \\
x &= \frac{(C^\phi - 1)/\phi \mod n^2}{n}
\end{aligned} \tag{A.2}$$

We did not use this method when obtaining our results in Section 5. Instead, we used a more efficient algorithm involving precomputation and Chinese remaindering, as described in Paillier's Ph.D. thesis [55].

A.3.1 Decryption with random help value r

It is also possible for some $\mathcal{P}$ who knows the r used to encrypt $C = E(x,r)$ to show $\mathcal{V}$ that x is the unique decryption of C by revealing r. $\mathcal{P}$ may know r either by having encrypted all the values used to compute C or by computing it via the decryption key ϕ. To recover x, $\mathcal{V}$ computes

$$x = \frac{(C \cdot r^{-n} \mod n^2) - 1}{n} \tag{A.3}$$

$\mathcal{P}$ can also recover random help r from $C = E(x,r) = (1+xn) \cdot r^n \pmod{n^2}$ by use of the secret decryption key ϕ as follows. (Note that our computations are modulo n and not modulo n^2 because r was taken from $\mathbb{Z}_n^*$.)

$$\begin{aligned}
r &= C^{n^{-1} \pmod{\phi}} \pmod{n} \\
&= (1+xn)^{n^{-1} \pmod{\phi}} \cdot r^{n \cdot n^{-1} \pmod{\phi}} \pmod{n} \\
&= 1 \cdot r^1 \pmod{n}
\end{aligned} \tag{A.4}$$

A.3.2 Uniqueness of Encryptions

Paillier's encryption scheme uses a bijection from $(\mathbb{Z}_n \times \mathbb{Z}_n^*) \to \mathbb{Z}_{n^2}^*$ [56]. [21] Thus any integer in $\mathbb{Z}_{n^2}^*$ represents a single valid encryption of an integer $x \in \mathbb{Z}_n$ with random help value $r \in \mathbb{Z}_n^*$. Consequently, if $C = E(x,r)$, $C \neq E(x',r')$ for any $x' \in \mathbb{Z}_n$ and $r' \neq r$. (This requires, as stated above, that $\gcd(n, \varphi(n)) = 1$.)

$\mathcal{P}$ can attempt to cheat by providing a different random help value r'. Using r' instead of r in (A.3) will yield a different but invalid "decryption" x'. $\mathcal{V}$ must therefore verify the provided value r' is consistent with the known encryption C. This is done by re-encrypting the derived value x' as $C' = E(x',r')$ and rejecting r' unless $C' = C$.

A.4 Mathematical Operations on Encrypted Values

The following definitions apply to any values encrypted as above, such as bids, deposit amounts, or desired quantities. These properties are due to the homomorphic properties of Paillier's encryption scheme [56]. In these definitions we refer to a prover $\mathcal{P}$ who has the decryption key or all random help values for encrypted data, (generally the auctioneer), and a verifier $\mathcal{V}$ who does not.

Addition. Addition of two encrypted values:

$$E(x) \cdot E(y) = E(x+y) \pmod{n^2}$$

Adding a constant k to an encrypted value x is easily done by encrypting k with the random help value 1 and multiplying the two encryptions.

$$E(x) \cdot (1+kn) = E(x+k) \pmod{n^2}$$

Multiplication or division by a constant. Division is only possible when k is invertible [22] $\bmod n^2$.

$$(E(x))^k = E(x \cdot k) \pmod{n^2}$$
$$(E(x))^{1/k} = E(x/k) \pmod{n^2}$$

Negation. Implied by multiplication by a constant.

$$(E(x))^{-1} = E(-x) \pmod{n^2}$$

[21] $\mathbb{Z}_n$ is the set of integers $[0,n)$; $\mathbb{Z}_n^*$ is the subset of $\mathbb{Z}_n$ relatively prime to n.

[22] This is no impediment, as finding a noninvertible k is tantamount to breaking the security key.

Comparison to a constant k. $\mathcal{P}$ can prove any encryption $C = E(k,r)$ is an encryption of k by revealing the help value r used to encrypt C. $\mathcal{V}$ then verifies that $(1+nk)r^n = C \pmod{n^2}$, because

$$E(k,r) = (1+n)^k \cdot r^n \pmod{n^2} \tag{A.5}$$

This is of particular interest when $k = 0$. We remark that no encryption of a value other than zero is an n^{th} residue [23] $\bmod n^2$.

[23] To say that x is an n^{th} *residue* $\pmod{m}$ means that there exists some value g such that $x = g^n \pmod{m}$. See also Footnote 14.