Cryptographic Securities Exchanges

Cryptographic Securities Exchanges

Christopher Thorpe, David C. Parkes

School of Engineering and Applied Sciences
Harvard University
Cambridge MA 02138
{cat,parkes}@eecs.harvard.edu

Abstract. While transparency in financial markets should enhance liquidity, its exploitation by unethical and parasitic traders discourages others from fully embracing disclosure of their own information. Traders exploit both the private information in upstairs markets used to trade large orders outside traditional exchanges and the public information present in exchanges' quoted limit order books. Using homomorphic cryptographic protocols, market designers can create "partially transparent" markets in which every matched trade is provably correct and only beneficial information is revealed. In a cryptographic securities exchange, market operators can hide information to prevent its exploitation, and still prove facts about the hidden information such as bid/ask spread or market depth.

Keywords: Cryptography, market microstructure, securities exchanges

1 Introduction

Market information plays a crucial role in modern securities exchanges. Published trades inform the public about the value of a particular security. Bid and ask quotations in limit books inform traders about other traders' interest in a security and at what prices orders are likely to be filled. Price change and trading volume information for equities track the (mis)fortunes and public awareness of corporations and equities markets. In theory, this market information should all benefit traders by forcing traders who have private information to disclose it via their trades. Unfortunately this information can also facilitate parasitic and unethical trading practices, and nondisclosure can itself lead to new exploits by market insiders who can then benefit their own accounts over investors' accounts. Balancing these forces is a significant challenge in market design, and homomorphic encryption techniques offer an attractive solution to this problem.

The application of homomorphic cryptography in other commercial protocols has been well studied in the academic literature (open and sealed-bid auctions [19, 14, 24], electronic cash [10], etc.) Yet, surprisingly little has been written about the contributions cryptography can make to securities markets, in particular the *open call auction* and *continuous double auction* protocols that underly most modern securities exchanges. Important prior work in this area includes Giovanni Di Crescenzo's pioneering work exploring privacy for stock markets [9], the secure double auction protocols Wang et al. propose in [23], which employs homomorphic ElGamal encryption, and a "Secure Protocol to Construct Electronic Trading" described by Matsuo and Morita in [16].

The work of Bogetoft et al. in [5], based on secure multiparty integer computation, proposes an application to securities exchanges, although in their protocol, "all trade is executed at the same market clearing price" and orders that do not clear are rejected. In our case, we wish to support both market orders and the limit order book that is an integral component of modern financial markets. Less directly but nonetheless related, Szydlo [22] has proposed the application of homomorphic cryptographic commitments

to the disclosure of stock portfolio holdings. Although some related work considers the privacy of trader identities, our work concerns only the revelation of quantitative information about trades not the anonymity of the traders, which we view as an orthogonal problem.

While the objective in most cryptographic work for auctions has been to hide information (secrecy), our objective is to enable a market designer to combine an appropriate level of *partial* transparency with provably correct behavior. We also do this in a setting informed by real-world demands, specifically, an exchange with both limit and market orders, and in which multi-party computation by all parties is infeasible.

Our design allows market designers to specify exactly what they wish to reveal, and reveal only that information while proving it, and the market operation, are correct. Immediate applications of our work can be seen in preventing unethical and parasitic trading practices in the major exchanges as well as providing for a means for trading large block orders without revealing information that can be exploited. Evidence for the need for information hiding in markets can be seen by recent SEC investigations and criminal convictions of unethical traders, and the development of new alternative trading systems (ATS's) that privately match large block trades. We detail how market information is misused and the securities industry's responses in Section 2.1.

In situating our work, we first discuss the role of information in securities markets from the perspective of *market microstructure*, a rich area of financial research that studies the role and exchange of information in markets and how market design principles serve to foster or inhibit information exchange. Market microstructure studies questions such as: What are the costs and benefits of transparency in financial markets? What determines the bid-ask spread for a particular stock? Do large orders really move the market? What is the effect of (not) publishing insider trades?

For simplicity, we will consider a single, *electronic clearing network* in which specialists, broker/dealers, or retail traders may place limit or market orders for shares of a particular equity (e.g. IBM stock) in a continuous double auction. We will explain the roles of each of these parties in the market, the types of transactions they may participate in, and why they do. We consider the various forms of information that these parties reveal through their actions (or inactions) and what information the markets reveal to them, and how they can profit from that information.

After considering the role of participants and information in our market, we construct a cryptographic framework that enables this information to be finely controlled and disseminated according to the specific rules established by a market operator. We observe that presently these types of information control are not achievable in financial markets because of a lack of trust: it is this transparency that proves *correctness* of the market transactions. Yet requiring full *transparency* to achieve correctness is a blunt method that can be exploited. We decouple these considerations.

Our proposed system proves correctness and provides for any level of transparency; being able to prove facts without directly revealing the numbers behind them offers market designers a more expressive set of possibilities for reporting market status. Our construction and protocols use homomorphic cryptography [18, 8, 19, 23, 22] to prove the correct operation of the market according to its published rules and also to credibly reveal the required market information to the participants.

It is not our intention to advocate particular kinds of transparency but rather to offer a finer level of control to market designers. Indeed, this application of cryptography seems to us to open up interesting new questions for the field of finance. We conclude with worked examples and report the result of an initial analysis of the cost to support a realistic order flow on current hardware.

2 Introduction to Financial Markets

In this section we provide an overview of how equities are traded in order to motivate our contributions to those without a background in finance. The study of market microstructure in finance is most applicable to our work; Larry Harris' book *Trading & Exchanges* [12] is a well respected textbook on the field; we also found three recent survey papers of market microstructure [4, 15, 21] helpful in framing our contributions.

In many cases, we will simplify the complex workings of modern financial markets in order to illustrate the core principles that are relevant to our work. We clearly indicate these simplifying assumptions in our exposition. We use as our model a market for a single equity for a single company and assume that all trades in that market take place on an electronic clearing network (ECN) running a continuous double auction with an open limit order book. We assume that the market operates at fixed daily opening and closing times and trading does not take place anywhere else when the market is closed. For simplicity, we do not consider short sales or buying to cover, which are equivalent to selling and buying long positions for our purposes.

The market maintains an *order book* in which all outstanding limit orders are recorded. Depending on the transparency rules of the market, all, some, or none of the limit orders on the order book may be available to the public. Real-world exchanges (NYSE, NASDAQ, Chicago Board of Trade) offer various degrees of transparency for their order books.

For the purposes of the cryptographic properties of our exchange, there is no important difference between dealers, brokers, specialists, or investors. In our simplified model, everyone may post limit orders and has access to the same information.[1] Therefore we only consider two classes of participants: the (market) *operator*, i.e. the exchange or its agent, and *traders*, in which we include specialists, broker/dealers, and institutional and retail investors.

As we present our model, we introduce formal definitions that we use later in our protocol construction. We model the market state as the current state of the limit order book B and trade history H. The order book B is private, but the trade history H is public. (H can be public because any values logged therein are maintained in encrypted form.) The market operator also maintains a public, encrypted order book $\hat{B}$ that is equivalent to B except that all bids and quantities are encrypted. Each order placed receives a unique identifier i regardless of whether it is a bid or ask order which is associated with the order and its components. Ask and bid orders, a_i and b_i respectively, enter the market when placed and exit the market when withdrawn or executed. An order is the tuple $(p_i, q_i, t_i, s_i \in \{\mathtt{a}, \mathtt{b}\})$ representing the price, number of shares, the time

[1] A simplified way to look at it is that dealers, brokers and specialists provide liquidity to the market to support the trades investors want to make. Possibly the most important function of liquidity providers' use of limit orders is enabling investors to place market orders.

the order was placed on the market, and the side of the market: whether the order is an ask (sell) or a bid (buy). When an order is taken off the order book, it is removed from the state of the order books B and $\hat{B}$ and the history H is updated with the execution or cancellation that resulted in its removal.

We will also refer to a function with access to a complete price ordering of the orders on the market $o(s \in \{\mathtt{a}, \mathtt{b}\}, rank)$ whose arguments are the side of the market (ask or bid) and the order's $rank$ where the most competitive price on either side has $rank = 0$. Its output is the unique identifier i for the ask or bid with the given $rank$. For example, we might write the current bid/ask spread as $p_{o(\mathtt{a},0)} - p_{o(\mathtt{b},0)}$, or the market depth (measured in shares) of the most competitive ten bid orders as $\sum_{r=0}^{9} q_{o(\mathtt{b},rank)}$. This ordering is maintained by the market operator; it is convenient for showing how the market operator proves correct operation of the market. Obviously, the ordering $o(s, r)$ changes whenever an order enters or exits the market. This function is also used to support the market invariant that all orders are maintained in strict priority order as described in Section 3.

In modern equities markets, orders fall into two basic categories: *market* orders are an instruction to buy or sell a specific quantity of a security, and are filled as soon as possible at the best available price on the market; *limit* orders are an instruction to buy or sell a specific quantity of a security at a specific price, and are filled only when another participant in the market is willing to make the opposite trade.

More complex orders that use real-time market information are possible, depending on broker support; for example, a *stop loss* order at a particular price instructs the broker to sell a position at the market when the market reports a trade at or below that price.[2] In practice, some traders also use orders based on real-time data as a substitute for limit orders because of the information revealed by limit orders or to get a better price; for example, an order such as "Buy 1,000 shares at the market if there are any trades below $20.00" might be used instead of a limit order to "buy 1,000 shares at $19.99" in order to keep the trader's intentions secret and potentially get a better price if the stock price dropped sharply. It might be that in a partially transparent market in which limit order prices are hidden, traders would be more inclined to use limit orders in these cases.

2.1 Market Information and Its Misuse

In this section we explore how transparency can be exploited, then examine at a high level the types of market information whose transparency may be regulated by cryptographic systems.

Misuse of Market Information. The information provided by transparency can be exploited by unethical or creative traders. To illustrate this hidden cost of transparency, we detail two common practices, one unethical and the other "parasitic": front-running and penny-jumping, respectively. Larry Harris' chapter "Order Anticipators" in *Trading & Exchanges* explores these and other related practices in depth [12]. We speculate that

[2] Limit orders cannot substitute for stop orders. Limit orders are persistent, and less competitive than the current equilibrium price; stop orders react to market movements and are at *more* competitive prices. Our framework can be extended to support stop orders with concealed prices; the operator would maintain a side list of stop orders and prove when their target prices are reached by executed trades, all without revealing either the trade price or stop order price.

these exploitations of transparency may be part of the cause for the conflict between published theoretical market microstructure results that show transparency should improve liquidity and other empirical results that are ambiguous with respect to this question [20].

"Front-running" is the unethical practice of a party with private information about an incoming large order to the market running in front of that order to take a position in the hope of making a quick profit when the large order arrives. For example, a trader knowing that a mutual fund is going to buy a $10M position in IBM stock might buy a smaller position beforehand with the expectation that the mutual fund's purchase will drive the price higher. Front-running and allegations of it are widespread. In 2001, Dreyfus agreed to pay $20.5 million to settle accusations that their fund manager Michael Schonberg engaged in front-running [1]. In 2003, the NYSE announced its Enforcement Division had investigated several specialist firms for rule violations including front-running and decided to bring disciplinary action against them. Seven specialist firms agreed to pay over $200 million to settle charges brought as a result of these allegations [2]. In July 2006 a Manhattan jury convicted former specialist firm Van der Moolen managers Michael Stern and Michael Hayward of fraud for trading stocks on the firm's account before filling clients' orders in order to boost Van der Moolen's profits and their own compensation [7]. In early 2007, *The New York Times* reported other allegations of front-running: "The [SEC] has begun a broad examination into whether Wall Street bank employees are leaking information about big trades to favored clients..." [3].

"Penny-jumping" is not illegal, but often described as "parasitic". The practice extracts value from the market without contributing information. Specifically, a trader identifies a large limit order on the order book (e.g. 10,000 shares at $25.00) and places a smaller limit order one tick above that order (e.g. 1,000 shares at $25.01). The penny-jumper's order will be filled first, and he expects that his upside is greater than his downside, because his downside is protected by a free trading option created by the large limit order. If the market is random, it is likely that the stock will trade at a higher price before the large order is filled. If the price happens to decline before it increases, the large order will be filled, and the penny jumper exits his position via the large order for a one-tick loss (e.g. after 8,000 of 10,000 shares have been filled).

One response to concerns of unethical and parasitic practices is to construct a market in which only partial information is reported. But it is unclear whether investors would trust that information's correctness or trust the market operators not to benefit from any private information. If prices are hidden, an unscrupulous market operator could simply fill a favored party's bid before higher bids. Indeed, regulators have begun to mandate transparency to protect investors, in light of specialists and broker/dealers exploiting private market information to their advantage [21, 15, 2, 1, 7].

There is also evidence that knowledge of large ("block") trades is similarly exploited. According to Stoll [21], large blocks of stock are not sent to the open market because "The risk of pre-trading portions of the block in this manner is that other traders will become aware of the block and will sell in anticipation, perhaps driving the price down..." and because other traders can exploit knowledge of large orders in other ways (as above). While Stoll further claims that "empirical evidence of block

trades is quite mild," Keim and Madhavan [13] (as cited in [15]) find in an empirical study that the average (one-way) price impact for a seller-initiated transaction is -10.2% from a benchmark three weeks before a large block trade, after adjustment for market movement.

Historically, block trades are performed in "upstairs markets" where brokers shop around for the best deal. Keim and Madhavan "attribute this large price impact to information 'leakage' arising from the process by which large blocks are 'shopped' in the upstairs market." [15] The reason for hiding information in block trades is mainly to protect the traders before the large transaction occurs.[3]

In September 2006, a number of major banks announced a response to the block trading problem with two new so-called ATS's (Alternative Trading Systems). Citigroup, Goldman Sachs, Lehman Bros., Merrill Lynch, Morgan Stanley and UBS also announced a "Block Interest Discovery Service" (BIDS) for automatically matching large block orders without revealing them to the primary markets. Another ECN, *Liquidnet*, specializes in institutional large block trades and has captured a small but significant share of order flow: as of 30 June 2006 they handled over $175 billion (notional) of trades with an average value of $1.42 million on US equities, according to their website. This is clear evidence that institutional investors are dissatisfied with traditional market support for large block trades.

While these approaches help to limit the exploitation of information they do not provide any correctness guarantees and moreover any published quotations can be exploited as before. These approaches also require that the block trades be separated from the primary securities exchanges. This could have a significant impact on liquidity and overall market efficiency. With our solution, quotations from the primary market can be integrated into our order book and matched against standing block trades; all transactions can be matched by a single efficient marketplace.

2.2 Developing a Cryptographic Securities Exchange

Our model is of a simple securities exchange in which a market operator keeps a private order book B and publishes its public analog $\hat{B}$ with encrypted prices and quantities, and (optionally encrypted) history of its actions H. Incoming limit orders are placed on the book or matched with existing limit orders; incoming market orders are matched with limit orders; the operator proves its actions correct.

Our primary goal is to prevent various adversaries from exploiting information present in limit order books to the detriment of traders who wish to place limit orders. These adversaries primarily include other traders and market insiders (market makers, specialists, exchange employees) who attempt to (unethically or parasitically) profit by exploiting limit order information.

Rindi [20] uses the term "partial transparency" in her examination of three regimes of pre-trade transparency in a market for a risky asset based on an open limit-order book: "under full transparency agents can observe the order flow and traders' personal identifiers; under partial transparency they can observe the order sizes and under anonymity they can only observe the market price."

[3] Gemmill [11] offers an empirical analysis consistent with this view of the effects of post-trade reporting of block trades on the London Stock Exchange. He finds *ex post* disclosure of block trades does not have a dramatic effect on liquidity.

We consider each of these information classes in turn. For existing orders, the type of the order is implied; if it is in the book, it is a limit order. Incoming orders may be market or limit orders; we assume that is disclosed. In call auctions, the transaction type (buy or sell) can be kept secret until the auction closes, but it is not meaningful to hide whether an order is to buy or sell in continuous double auctions. As noted before, timed expiration of orders is unimportant.

The price per share p_i associated with an order a_i or b_i on the book may be fully transparent ($p_i = \$20.06$), partially transparent ($\$20.00 \leq p_i \leq \$20.25$), or kept completely private ($p_i = ?$). Similarly, the quantity q_i and time posted t_i may be fully, partially, or not transparent.

The parameters of multiple orders may be related by inequalities. Two orders may be related by price (e.g. $p_i \geq p_j$), quantity (e.g. $q_i = q_j$) or time posted (e.g. $t_i < t_j$). Partial or complete orderings for price and time of all orders in a limit book can be constructed using these methods, as will become important for more expressive partially transparent revelations. Quantity becomes important when proving order flow and correct execution of trades.

Finally, one might wish to prove information about linear functions on the parameters of multiple orders, or compute linear functions without revealing unnecessary additional information about the orders themselves. Examples of these functions include:

- Bid/ask spread between the two most competitive orders
- Market depth within p cents of the mean between the outstanding bid and ask (measured in number of shares)
- Bid-ask spread between the two least competitive orders comprising a market depth of q shares
- Prices (if any) at a market depth of q shares
- Average number of hours outstanding orders above price p have been on the market

Using recent advances in homomorphic encryption, market designers can construct markets in which this kind of information can be revealed and proved correct without revealing additional information about underlying orders.

Information, and related proofs, need not be issued in real-time, and in fact in many cases market designers may prefer delayed revelation. In our system, market designers can decide exactly when to reveal market activity, and even construct different disclosure rules for different trade sizes. For example, the market might disclose small trades within 30 seconds and large trades within 1 day.

3 The Cryptographic Securities Exchange

We have described the model of our market as a limit order book with a history. We consider the state of the order book B, the encrypted public order book $\hat{B}$, and the history H to be the core state of our market. Various actions by the participants in the markets update this state. We formally define these actions, who may perform them, and how the update the state of the market depending on its state. The order book and history begin as empty states.

In our present model we maintain an important invariant in B and $\hat{B}$: all orders are maintained in a strict priority ordering as defined by the ordering function $o(s, rank)$.

Despite regulations that prescribe order routing priority, the priority of trades within active markets is a complicated process beyond the scope of the present work. For example, smaller orders at slightly less competitive prices or more recently submitted might be filled instead of a large order that is the longest standing at the most competitive price.

We model these priority rules as follows, from highest to lowest:

1) Most competitive price (p_i is maximal)

2) Longest standing (t_i is minimal)

3) Best "fill", measured by the percentage of shares filled of the larger of the two orders ($\frac{|q_i - q_j|}{\max(q_i, q_j)}$ is maximal).

We do not consider a formal mechanism for proving the time priority of an order correct, in part because we see no benefit in encrypting the timestamp of an order: orders are posted when they arrive, and that reveals the time they were posted. Further, this information is not readily exploitable.

We assume a bulletin board that orders are posted to; the market operator is required to accept new orders by adding them to the history H as soon as they arrive. We also assume that at the beginning of each new trading session the public, encrypted order book $\hat{B}$ has been verified by tracing through the previous day's history in H.

3.1 Assumptions

Our protocol rests on certain realistic assumptions. The operator and all traders possess the means for generating secure digital signatures. A universal, tamper-resistant clock must be accessible by all parties, such as that maintained by the US NIST, to preserve the integrity of timestamps. To prevent the operator from improperly failing to disclose instructions, there is a universally accessible bulletin board—not maintained by the operator—that records all activities of all parties and publishes them for anyone to see.[4] (All private data remain secure by encryption.) We assume the hardness of the composite residuosity problem supporting Paillier's homomorphic encryption scheme [18]. We assume that a computer network may be monitored for activity, and that even large amounts of activity can be examined for any information "leakage".

3.2 Encryption Method

We employ the homomorphic encryption scheme described by Pascal Paillier [18] and extensions published by Damgård and Jurik [8], Parkes et al. [19], and a use of Boudot's efficient range proofs [6]. We write the encryption of a value m with the market operator's public key and random help value r as $E(m, r)$. The properties of this cryptosystem allow construction of mathematical proofs of certain facts over the ciphertexts. For example, given only $E(m_1, r_1)$ and $E(m_2, r_2)$, one can prove a value is within a constant range, e.g. $m_1 < n/2$; inequalities, e.g. $m_1 > m_2$; or generate new ciphertexts that are the sum of others, e.g. $E(m_1 + m_2, r_1 \cdot r_2) = E(m_1, r_1) \cdot E(m_2, r_2)$. We require these primitives for proving the correct operation of the market.

[4] We assume a bulletin board strictly separate from the operator so that traders' orders may be presumed received and posted on time without respect to their content. Because the operator can decrypt incoming orders, it is important that all incoming orders be posted by a neutral third party to require the operator to prove its actions are correct; a corrupt operator could delay or ignore incoming orders to benefit favored traders.

3.3 Processing Incoming Orders

Before orders arrive in any trading session, we recall that we assume the operator has proven the public, encrypted order book $\hat{B}$ correct by reference to the orders posted on the bulletin board in previous sessions. This means that all transactions may be performed with respect to existing orders in the order book without need for further proofs of their correctness or rank in the order book.

Limit Orders. Any trader in our model may place a limit order according to the following protocol. Each limit ask order a_i is given a unique id i by the bulletin board and enters the market in the following manner. Note that the same method applies for bid orders b_i by interchanging "ask" and "bid" and reversing inequalities ($<$ becomes $>$).

Step 1. The trader encrypts the price p and quantity q and sends $(E(p, r_p), E(q, r_q), \mathtt{a})$ to the bulletin board. The bulletin board creates a unique identifier i, adds a timestamp t_i based on the current clock, publishes $\hat{a}_i = (E(p_i, r_{p_i}), E(q_i, r_{q_i}), t_i, \mathtt{a})$, computes the digital signature $SIGN_{BB}(\hat{a}_i)$ and both publishes it and sends it to the trader as a receipt. Only the operator can see what the p_i and q_i are.

Step 2. The trader privately sends the random help values r_{p_i}, r_{q_i} to the operator.[5]

Step 3. The operator privately decrypts the values in $\hat{a}_i$ to compute $a_i = (p_i, q_i, t_i, \mathtt{a})$, and verifies that the random help values correspond to the ciphertexts provided.

Step 4. The operator logs in H that order a_i was received at time t_i.

Step 5. The operator compares p_i to the best ask price, $p_{o(\mathtt{a},0)}$ and the best bid price, $p_{o(\mathtt{b},0)}$ and proceeds in one of four ways:

- If the incoming ask order is priced at less than or equal to the highest priority bid, i.e. $p_i \leq p_{o(\mathtt{b},0)}$, the operator matches a_i with all outstanding bid orders whose prices are $\geq p_i$ up to the quantity q_i in order of priority. If there are not enough to fill a_i, it becomes the most competitive ask order on the order book afterward.
- If the incoming ask order is priced between the highest bid and the lowest ask price, i.e. $p_{o(\mathtt{b},0)} < p_i < p_{o(\mathtt{a},0)}$, the operator adds it to the order book.
- If the incoming ask order is priced equal to the lowest ask price, i.e. $p_i = p_{o(\mathtt{a},0)}$, the operator adds it to the order book.
- If the incoming ask order is priced higher than the lowest ask price, i.e. $p_i > p_{o(\mathtt{a},0)}$, the operator adds it to the order book.

Step 6. The operator updates H on the bulletin board with the details of any trade that resulted from receiving a_i.

Step 7. The operator recomputes the ordering function $o(s, rank)$ such that the rank of all orders in B is defined and correct.

Step 8. The operator updates its private B and publishes $\hat{B}$ on the bulletin board with the new set of encrypted orders.

[5] This is required to prevent other traders from exploiting the malleability of the homomorphic encryption scheme to submit bids based on a function of another trader's bid, e.g. "his bid plus 10 cents." Knowing the random help value implies knowing the decryption, so provided the cryptosystem is secure and the random help values are secret, no trader can submit a correct random help value for a ciphertext based on another trader's encrypted values.

Step 9. The operator issues proofs of correctness of its actions on the bulletin board. Specifically, it proves the necessary inequalities to pigeonhole the incoming limit order a_i in its proper priority ordering, maintaining the invariant that the all outstanding orders in B and $\hat{B}$ are ordered according to priority.

Step 10. Anyone who wishes may verify the operator's public proofs.

Market Orders. A trader in our model may also place a market ask order a_i (or bid b_i). The protocol differs from the limit order protocol given above only in Step 5:

Step 6. The operator matches the incoming market ask order a_i with the k highest priority bid orders $b_{o(\mathtt{b},0,\ldots,k)}$ such that the $k-1$ highest bids do not fill a_i but k do, and executes the trade(s) on all matched orders.

Executing Trades on Matched Orders. The operator must prove that the quantity of the k multiple limit orders a large order is matched with is greater than or equal to the quantity of the market order, and that the sum of the quantities of the most competitive $k-1$ limit orders is strictly less than the quantity of the market order.

Two orders a_i and b_j are matched when the bid price meets or exceeds the ask price, i.e. $p_j \geq p_i$. If the quantities are equal, $q_i = q_j$, the trade is executed and both orders are removed from the order books B and $\hat{B}$ and the transaction is logged in the history H. Formally, to log the transaction the operator adds a journal entry to H $h_{i,j} = (\hat{a}_i, \hat{b}_j, t_{i,j})$ with its signature $SIGN_{MO}(h_{i,j})$. The time $t_{i,j}$ is the time reported by the universal clock at the time the order was executed. The operator also posts the following proofs on the bulletin board:

- A proof that $p_j \geq p_i$ given $E(p_i, r_{p_i})$ and $E(p_j, r_{p_j})$.
- A proof that $q_j = q_i$ given $E(q_i, r_{q_i})$ and $E(q_j, r_{q_j})$.

If the quantities differ, the order for fewer shares is fully filled and the order for more shares is partially filled. Then, the smaller order (w.l.o.g. a_i) is removed and the larger order (w.l.o.g.) b_j's quantity is updated in the order books B and $\hat{B}$. Formally, the entry b_j in B is replaced with $b_j = (p_j, (q_j - q_i), t_j, \mathtt{b})$, and in $\hat{B}$ with $\hat{b}_j = (E(p_j, r_{p_j}), E(q_j, r_{q_j})/E(q_i, r_{q_i}), t_j, \mathtt{b})$. Anyone can verify the correctness of the new published $\hat{b}_j$ by computing the quotient of the previously published encrypted values $E(q_j, r_{q_j})$ and $E(q_i, r_{q_i})$, which is known to be an encryption of their difference. The transaction is logged in the history H as above with a similar journal entry $h_{i,j} = (\hat{a}_i, \hat{b}_j, t_{i,j})$ and signature $SIGN_{MO}(h_{i,j})$. The operator also posts the following proofs on the bulletin board:

- A proof that $p_j \geq p_i$ given $E(p_i, r_{p_i})$ and $E(p_j, r_{p_j})$.
- A proof that $q_j > q_i$ given $E(q_i, r_{q_i})$ and $E(q_j, r_{q_j})$. This is done by showing that $(E(q_j, r_{p_j})/E(p_i, r_{p_i})) \cdot E(-1, 1)$ is the encryption of a value $(q_j - q_i - 1) < n/2$. (This proves that no wraparound occurred; we subtract 1 from $q_j - q_i$ to prove a strict inequality.)

One minor issue in a market without transparent prices is that a limit order may be submitted to the market that is more competitive than it needs to be to clear. For example, a trader might post a new limit order to sell at \$20.05 when there is a standing order

to buy at $20.09. In transparent markets, this would obviously never happen except in cases of error. Choosing the clearing price for such situations is a matter of market design. With the primitives we have described, it is possible to prove correct a clearing price based on the standing order's price, the incoming order's price, the mean of the two (within one tick), or indeed any linear function of the two prices, without revealing the price itself or any information not implied.

Once two orders are matched and the proofs posted, a clearing agent will be responsible for transferring the ownership of the shares at the correct settlement price. The market operator will send the clearing agent the random help values necessary to verify the correctness of the execution price and number of shares from the history posted on the bulletin board. The agent then verifies the trade and settles it.

In addition to sending information to the clearing agent, any information published about the state of the market is proven at this point on the bulletin board. For example, the auctioneer might reveal the random help values associated with the determined clearing price and matched quantity to provide "last trade" tick data, or update proofs of market depth, bid/ask prices, etc. Typically the "market price" of a security for any period is the price at which it was last traded during that period; thus, publishing provably correct market prices is straightforward.

3.4 Post-Trade Reporting

The market operator can report clearing prices by revealing the random help values of the encrypted orders in the history H after any specified delay. Immediate revelation may be a problem in the event a partial fill is revealed and the remainder is still on the market: is price is now public. Facts similar to those provable for limit orders may be proven about trades after the fact, for example, volume, average price, closing price, etc. Post-trade transparency is as easily controlled by market designers as transparency during other phases of market activity, and we leave the question of appropriate reporting rules open for this reason.

3.5 Adversaries and Attacks

The adversary we are most concerned about in this work is the unethical or parasitic trader who exploits (presently public) market information for profit in a way that discourages placement of limit orders. A secondary class of adversary is a dishonest market operator who may attempt to profit by exploiting the now private market information via trading or disclosure for compensation. We do not consider as adversaries parties with private information external to the market's operation, such as employees with proprietary information about traded companies.

Traders We first consider attacks by parties who do not possess any insider access to the market operator or its systems. These traders may either attempt to circumvent the cryptographic security of the system or exploit the information provided in new ways. Provided cryptographic keys of adequate security are chosen to prevent a brute-force attack, cracking the encryption scheme itself is believed to be intractable under the Decisional Composite Residuosity Assumption described in Paillier's work [18].

The semantic security of the probabilistic Paillier cryptosystem protects the encrypted values against chosen plaintext attack. (For example, using a deterministic encryption of prices would be insecure, because an adversary could try all realistic prices

and identify the values.) Paillier's scheme is not secure against an adaptive chosen ciphertext attack; indeed, the malleability of the scheme that enables the homomorphic properties we employ implies this insecurity.However, mounting a successful chosen ciphertext attack against our protocol does not seem a significant threat, as the only way a value can be decrypted is in the event someone is willing to trade it. Thus, any party attempting to gain information by submitting a chosen ciphertext as information must also be willing to execute any trades from that information.

We have not identified any additional parasitic trading practices that could be employed using a cryptographic securities exchange. Since we are not adding any information into the marketplace – only allowing designers to restrict information – we believe that there are no new exploits that would not be possible in an ordinary market with an open limit book.

This said, we reiterate that some parties may attempt to gain information from the marketplace by placing orders. For example, one could discover the price for the most competitive ask order by placing an order to buy one share at the market. Alternatively, a trader might place limit orders at various prices to see where they fit into the order book, in order to gain information about the price points, and then retract them. However, no trader may observe anything about the market without fundamentally changing the market: a "probe" share purchased revealed the price *for that share only*, and afterward, the number of shares at that price remains unknown and becomes smaller; probe limit orders enter the market and always bear the risk of being executed.

Several solutions to this problem come to mind. First, at a significant but tractable complexity cost, the marketplace could maintain not a strict ordering over all orders, but a partial ordering in which only the minimum information required to prove correctness is revealed. Thus incoming orders that were not competitive (and likely to be filled) would be proven only to be less competitive than the most competitive order. This would significantly limit the ability of a trader to count trades above a particular price by placing limit orders. Second, the market operator or market makers could place random numbers of zero-quantity limit orders on the marketplace so that there would be a large number of orders at every price point. Third, market designers could limit such exploitative practices by limiting order frequency, sizes, or specifying a minimum duration on the market.

The Market Operator A more insidious attack is if a dishonest market operator, possibly in collusion with another trader, exploits its valuable private information or gives preference to particular traders. We recall our assumption of a bulletin board operated by a third party to prevent the market operator from discarding dispreferred orders, or delaying their publication until after preferred orders are listed. With this, an unscrupulous market operator cannot issue valid proofs of correctness of matched trades, but he could still selectively reveal information to preferred traders. We reiterate that despite this implied trust in the market operator, our architecture provides for two improvements over existing markets: information can be specifically controlled and is possessed by only one party (instead of the entire market), and the market operator may not manipulate the market by front-running or matching orders on any basis other than the published rules.

That said, the partial trust of the operator is a strong assumption, and solutions to enhance that trust merit discussion. One answer is to distribute the trust in the market operator among a group of parties, similar to the approach Bogetoft et al. describe [5]. This may be challenging from a business perspective but nonetheless possible. Another solution involves careful network, hardware and software security, employing special purpose hardware (e.g. that used in Trusted Computing architectures) that only runs software approved and signed by a third party, and monitoring all network traffic to detect any communications that might leak information.

4 Example Order Book and Transactions

This section describes incoming orders and how trades are identified and executed. Table 1 shows a sample order book B. The public, encrypted order book $\hat{B}$ is equivalent, except that the quantities and prices are encrypted. **R** indicates $rank$. Orders are always ranked in priority order. Each order's rank is defined according to the priority rules outlined above (best price, oldest) and randomly selected in the case of a tie.

We first consider an incoming market order to purchase 700 shares of the stock. The trader constructs $\hat{b} = (\_, E(700), \_, \mathtt{b})$ and posts it on the bulletin board. The bulletin board assigns ID $i = 25$ and timestamp $t_i =$ 09:44:32 and publishes $\hat{b}_i = (\_, E(700), t_i, \mathtt{M})$. For clarity, we will use i for the ID of each incoming order in the following text to more clearly distinguish it from the limit orders.

The market operator sees the market order on the bulletin board, decrypts $\hat{b}_i$ to $b_i = (\_, 700, t_i, \mathtt{b})$, and matches two trades ($a_1 4$, $a_1 2$) to fill the order. It adds journal entries to the history H and publishes proofs on the bulletin board:

- $H \leftarrow h_i = \hat{b}_i$
- $H \leftarrow h_{14,i} = (\hat{a}_{14}, \hat{b}_i, t_{14,i} =$ 09:44:33)
- $H \leftarrow h_{12,i} = (\hat{a}_{12}, \hat{b}_i, t_{12,i} =$ 09:44:33)
- Proofs of correct quantities: $q_{14} + q_{12} \geq q_i$ and $q_{14} < q_i$
- Sufficient proof of priority: $q_{12} < q_{13}$

R	ID	Time	Qty	Ask
3	11	09:34:42	2500	$20.13
2	13	09:39:23	500	$20.10
1	12	09:39:23	300	$20.10
0	14	09:41:06	600	$20.09
R	**ID**	**Time**	**Qty**	**Bid**
0	22	09:37:14	1000	$20.05
1	24	09:43:42	500	$20.02
2	23	09:41:23	800	$20.00
3	21	09:30:06	1700	$19.96

Table 1. Order Book B_1

The operator then updates B (and $\hat{B}$) by removing order a_{14} and updating $q'_{12} = 300-(700-600) = 200$ (and $\hat{q}'_{12} = E(q_{12})/(E(q_i)/E(q_{14}))$). Anyone can verify that the updated encrypted quantity $\hat{q}'_{12}$ is correct by comparing it with functions of the quantities of the other orders.

In a second example, a trader posts a new limit ask order $\hat{a} = (E(\$20.03, E(1200), \_, \mathtt{a})$ to which the bulletin board assigns $i = 15, t_i =$ 09:46:02. The market operator sees it, decrypts it, and concludes it is more competitive than the most competitive bid. He adds journal entries to H, removes $b_{o(\mathtt{b},0)}$, matches a_i with b_{22} and adds the remainder a'_i to B and $\hat{a}'_i$ to $\hat{B}$, preserving the priority order invariant, and publishes:

- $H \leftarrow h_i = \hat{a}_i$
- $H \leftarrow h_{i,22} = (\hat{a}_i, \hat{b}_{22}, t_{i,22} =$ 09:46:04)
- Proof of correct quantities: $q_i > q_{22}$

– Proof of price position: $p_i \leq p_{22}, p_i > p_{24}$
– Proof of clearing price (as required)

In a final example, a trader posts a limit bid order $\hat{b} = (E(\$19.98, E(400), \_, \mathtt{b})$ to which the bulletin board assigns $i = 26, t_i =$ 09:50:33. The market operator sees it, decrypts it, and places it in the order book in the appropriate position It adds a journal entry to H, adds the order b_i to B and $\hat{b}_i$ to $\hat{B}$, preserving the priority order invariant, and publishes $H \leftarrow h_i = \hat{b}_i$ and the proofs of priority $p_i < p_{23}$ and $p_i > p_{21}$. The order book is now as shown in Table 2.

R	ID	Time	Qty	Ask
3	11	09:34:42	2500	$20.13
2	13	09:39:23	500	$20.10
1	12	09:39:23	200	$20.10
0	15	09:46:02	200	$20.03
R	**ID**	**Time**	**Qty**	**Bid**
0	24	09:43:42	500	$20.02
1	23	09:41:23	800	$20.01
2	26	09:50:33	200	$19.98
3	21	09:30:06	1700	$19.96

Table 2. Order Book B_4

5 Conclusions and Future Work

Clearly, providing controllable transparency of market information in securities exchanges together with proofs of correctness (both of information and of the market operation) is an important application of homomorphic cryptography. The protocol presented here is simple to understand, closely related to existing financial market protocols, and does not rely complex cryptographic primitives that might discourage its use among traders. Finance research has already started to study the implications of different levels of partial transparency, seeking to ensure liquidity and limit exploitation. Cryptography can be used to prove correct operation according to specified rules even under partial transparency.

We envision a broad range of future work based on the protocol we have presented and similar ideas. For instance, market designers might want support for more expressive order types, such as fill-or-kill, immediate-or-cancel, order-cancels-order, or stop orders maintained by the market. Our protocol could also easily be extended to open call auctions or periodic clearing models (such as POSIT). The market operator might wish to prove a less revealing ordering of the limit orders in the order book. Support for other specialists and liquidity providers' functions could be added by selective revelation.

Other more creative exchanges are possible in our setting. For example, integrating other ECN's with a cryptographic securities exchange may be of particular use in bridging the gap between block trades and ordinary securities trading. Cryptographic derivative markets for options and indices whose prices are tied to the activity in underlying securities' order books are another important possible extension of our work.

We have conducted an initial empirical analysis of the computation cost for running such a system, and arrived at a conservatively high estimate of 5 cents (US) to place and verify an order. Our experiments used a low end, dual Pentium IBM x-server with no special cryptographic hardware. This is inexpensive enough to be feasible in practice, although we leave a full efficiency analysis, perhaps in conjunction with a prototype, to future work.

6 Acknowledgments

We thank Michael O. Rabin and Stuart M. Shieber for helpful discussions related to this work and their essential contributions to our related joint work [19]. We are also grateful to Eric Budish for useful references and comments.

References

1. Dreyfus will pay $20.5 million to settle lawsuit. *The New York Times*, 22 June 2001.
2. Settlement reached with five specialist firms for violating Federal securities laws and NYSE regulations. U.S. SEC Press Release, 2004. `http://www.sec.gov/news/press/2004-42.htm`.
3. J. Anderson. S.E.C. is looking at stock trading. *The New York Times*, 6 February 2007.
4. B. Biais, L. Glosten, and C. Spatt. Market microstructure: a survey of microfoundations, empirical results and policy implications. *Journal of Financial Markets*, 8(2):217–264, May 2005.
5. P. Bogetoft, I. Damgård, T. Jakobsen, K. Nielsen, J. Pagter, and T. Toft. A practical implementation of secure auctions based on multiparty integer computation. In *Proc. 10th International Conference on Financial Cryptography and Data Security (FC 2006)*, 2006.
6. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Proc. EUROCRYPT '00*, pages 431–444, 2000.
7. C. Bray. Two ex-Van der Moolen specialists are convicted of securities fraud. *The Wall Street Journal*, 15 July 2006.
8. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Proceedings of Public Key Cryptography '01*, 2001.
9. G. Di Crescenzo. Privacy for the stock market. *Lecture Notes in Computer Science*, 2339:269 ff., 2002.
10. Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect Discourse Proofs": Achieving efficient fair off-line E-cash. In K. Kim, editor, *Advances in Cryptology: Proceedings of ASIACRYPT 1996,* Kyongju, Korea, number 1163 in Lecture Notes in Computer Science, Berlin, 1996. Springer Verlag.
11. G. Gemmill. Transparency and liquidity: A study of block trades on the London Stock Exchange under different publication rules. *Journal of Finance*, 51:1765–1790, 1994.
12. L. Harris. *Trading and Exchanges*. Oxford University Press, 2003.
13. D. B. Keim and A. Madhavan. The upstairs market for large-block transactions: Analysis and measurement of price effects. *Review of Finacial Studies*, 9:1–36, 1996.
14. 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.
15. A. Madhavan. Market microstructure: A survey. 8, March 2000.
16. S. Matsuo and H. Morita. Secure protocol to construct electronic trading. *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences*, E84-A(1):281–288, 2001.
17. P. Paillier. *Cryptographie à Clé Publique Basée sur la Résiduosité de Degré Composite*. PhD thesis, École Nationale Supérieure des Télécommunications, 1999.
18. P. Paillier. Public-key cryptosystems based on composite residuosity classes. In *Proc. EUROCRYPT '99*, pages 223–239, 1999.
19. D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. A. Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. In *ICEC '06: Proceedings of the 8th international conference on Electronic commerce*, pages 70–81, New York, NY, USA, 2006. ACM Press.
20. B. Rindi. Transparency, liquidity and price formation. In *Proceedings of the 57th European Meeting of the Econometric Society*, 2002.
21. H. R. Stoll. Market microstructure. In G. M. Constantinides, M. Harris, and R. Stulz, editors, *Handbook of the Economics of Finance*. Elsevier Science B.V., 2003.
22. M. Szydlo. Risk assurance for hedge funds using zero knowledge proofs. In *Proc. 9th International Conference on Financial Cryptography and Data Security (FC 2005)*, 2005.
23. C. Wang, H. Leung, and Y. Wang. Secure double auction protocols with full privacy protection. In *Information Security and Cryptography - ICISC 2003: 6th International Conference*, 2003.
24. 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.