FuturesMEX
Secure, Distributed Futures Market Exchange

Fabio Massacci
University of Trento, Italy

Joint work with Chan Nam Ngo, Jing Nie, Daniele Venturi, Julian Williams

Some of this work is subject of the following patent applications
US62/625,428 PG448130GB

Outline

• What this is all about?
• Futures trading
  • Exchange functionality
  • Motivation & challenges
• Solution
• Evaluation
The Talk Message in a Slide

• In standard security we have
  • Good Guys, Bad Guys and
  • Failures \( \rightarrow \) caused by Bad guys (or by “Good” guys who ain’t so Good)
• More Good Guys join \( \rightarrow \) old Good Guys still Good
  • So are their security proofs, credentials and all that
• Distributed FinTech is not like that
  • You have “honest failures” \( \rightarrow \) economics forces security to be non-monotonic
  • And this has MAJOR implications for security design
• Key Intuition \( \rightarrow \) Security Protocol Workshop 2018
• Full solution for Chicago Mercantile Exchange \( \rightarrow \) IEEE Symp S&P 2018

Futures market as illustrative of FinTech

• A double auction market
• Bidders on both buy/sell side
• Futures contract
  • standardized promise to buy/sell barrels of oil, bushels of corn, ...
  • made today and to be fulfilled in a future date
  • with cash reserve to meet promises
• Exchange platform for trading activities
  • Chicago Mercantile Exchange \( \rightarrow \) centralized
• Other applications \( \rightarrow \) Invoice Factoring (UNBIAS project)
**How futures trading works?**

<table>
<thead>
<tr>
<th>Trader</th>
<th>Promises</th>
<th>Cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
<td>1200</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>1500</td>
</tr>
</tbody>
</table>

Market price = 10$

Alice sells 100 promises
Bob buys 100 promises

<table>
<thead>
<tr>
<th>Trader</th>
<th>Promises</th>
<th>Cash at the exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Buy 100</td>
<td>2200=1200+100*10</td>
</tr>
<tr>
<td>Bob</td>
<td>Sell 100</td>
<td>500=1500-100*10</td>
</tr>
</tbody>
</table>

At end of (trading) day
Market price = 8$

Promises must be fulfilled at end of day price:
Bob must sell and Alice must buy from the market

Alice made a profit of 200$, Bob lost.

**Centralized futures trading (2)**

<table>
<thead>
<tr>
<th>Trader</th>
<th>Promises</th>
<th>Cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>0</td>
<td>1200</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>1500</td>
</tr>
</tbody>
</table>

Market price = 10$

Alice sells 100 promises
Bob buys 100 promises

<table>
<thead>
<tr>
<th>Trader</th>
<th>Promises</th>
<th>Cash at the exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Buy 100</td>
<td>1400=2200-100*8</td>
</tr>
<tr>
<td>Bob</td>
<td>Sell 100</td>
<td>1300=500+100*8</td>
</tr>
</tbody>
</table>

At end of day
Market price = 12$

Promises must be fulfilled at current price

Bob made a profit but Alice lost 200$
Market price is volatile

<table>
<thead>
<tr>
<th>Trader</th>
<th>Promises</th>
<th>Cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Buy 100</td>
<td>2200</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

100 promises to buy when price was at 10$ looked a good idea but things change

Alice’s cash reserve is now at 0$

→ Exchange must do something

2200 > 12*100
2200 > 17*1000
2200 = 22*100!!!

If price rises further Alice’s going broke

Hi Alice, can you deposit more money?

No? → Alice is liquidated.

The purposes of the exchange?

Publish the order book
Aggregate all orders
Protect traders anonymity
Match orders
Manage risks
Price oscillates
Retail and Institutional Traders
High frequency traders (HFTs)

<table>
<thead>
<tr>
<th>Market price = 11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sell</td>
</tr>
<tr>
<td>500 @ 20$</td>
</tr>
<tr>
<td>400 @ 16$</td>
</tr>
<tr>
<td>1000 @ 12$</td>
</tr>
<tr>
<td>Buy</td>
</tr>
<tr>
<td>900 @ 10$</td>
</tr>
<tr>
<td>600 @ 8$</td>
</tr>
<tr>
<td>700 @ 7$</td>
</tr>
</tbody>
</table>
The purposes of the exchange?

- Publish the order book
- Aggregate all orders
- Protect traders anonymity
- Match orders
- Manage risks
- Price oscillates

 Леан Хог (низкая частота)
- # traders = 33
- # orders = 6709
- # matches = 536

Евро доллар (высокая частота)
- # traders = 520
- # orders = 300K+
- # matches = 8402

Market price = 11$

Buy
- 900 @ 10$
- 600 @ 8$
- 700 @ 7$

Exchange as reactive functionality

02/09/18

F. Massacci – Univ. of Trento, Italy, Sec. Seminar @ Aalto Univ.
**Exchange as reactive functionality**

- Initialize (deposit)
- Post Order Cancel Order
- Check ALL Positions
- Valid action
- Cancel pending orders
- Liquidate contracts
- Any BROKE
- Matches
- No BROKE
- End of Day OR Cannot fix BROKEs
  - Mark To Market

---

**Exchange as reactive functionality**

- Initialize (deposit)
- Post Order Cancel Order
- Check ALL Positions
- Valid action
- No BROKE
- Cancel pending orders
- End of Day OR Cannot fix BROKEs
  - Mark To Market
**FuturesMEX**

- Replace centralized exchanges
- To allow buy and sell promises based on limited cash reserve
- Enforce trading discipline
- Protect market integrity

---

**Technical challenges**

- Easy to see
  - Market integrity
  - Consensus
- Less obvious
  - Account confidentiality
  - Trader anonymity
  - Non-monotonic behavior
  - Honest actions invalidate past security evidences
  - Proportional burden
    - Retail & institutional traders vs HFTs

ALL come as a package, or NOTHING will work individually.
### Confidentiality & Price Discrimination Attacks

<table>
<thead>
<tr>
<th>T</th>
<th>Promises</th>
<th>Cash</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Buy 90</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>C</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>E</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Market price = 10$**

<table>
<thead>
<tr>
<th>Buy</th>
<th>Sell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E: 20 @ 14</td>
</tr>
<tr>
<td>B: 20 @ 9</td>
<td>-</td>
</tr>
<tr>
<td>A: 90 @ 8</td>
<td>-</td>
</tr>
</tbody>
</table>

**IF E knows**

1. A is tight in cash
2. A must buy 90 contracts

→ Can E bankrupt A?

---

### Confidentiality & Price Discrimination Attacks

<table>
<thead>
<tr>
<th>T</th>
<th>Promises</th>
<th>Cash</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Buy 90</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>C</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>E</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Market price = 10$**

<table>
<thead>
<tr>
<th>Buy</th>
<th>Sell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E’s evil scheme:</td>
</tr>
<tr>
<td></td>
<td>→ Push price to 11.5$</td>
</tr>
<tr>
<td></td>
<td>→ Convince C to cancel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>Promises</th>
<th>Cash</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Buy 90</td>
<td>1000</td>
<td>-35</td>
</tr>
<tr>
<td>B</td>
<td>Sell 30</td>
<td>1200</td>
<td>1545</td>
</tr>
<tr>
<td>C</td>
<td>Sell 20</td>
<td>1310</td>
<td>1550</td>
</tr>
<tr>
<td>E</td>
<td>Sell 40</td>
<td>1090</td>
<td>1540</td>
</tr>
</tbody>
</table>
Confidentiality & Price Discrimination Attacks

<table>
<thead>
<tr>
<th>T</th>
<th>Promises</th>
<th>Cash</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Buy 90</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>C</td>
<td>Sell 30</td>
<td>1200</td>
<td>1500</td>
</tr>
<tr>
<td>E</td>
<td>Sell 40</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

E’s evil scheme:
→ Push price to 11.5$
→ Convince C to cancel

ONLY works if E knows
A’s exact position
Confidentiality is (technically) essential

Non-Monotonicity: What’s different from other crypto protocols?

• In all security protocols we are used to
  • All good guys do the same thing!
  • And they all do it once!
• Authentication
  • Alice wants to be authenticated by a TLS server
  • And so does Bob I, and Bob II, and Bob III, and Bob IV
• E-Voting → Alice casts 1 vote, and so Bob I, Bob II,
• Auctions → Alice makes 1 bid, and so Bob I, ...
• Reputation Systems → Alice posts her rating, and so does Bob I, Bob II, Bob III, ...
Enter Distributed FinTech

- Fat cat Sam is gone → only Alice and the Bobs
- Alice trades in Barrels of Oil with the Bobs
  - Commits she’ll buy B barrels at the end of the day
  - Proves in ZK she has cash $>$ B*P to buy them at current price P
  - Bob III agrees to sell her B barrels at whatever end price
- All is good and the Bobs making offers
- Seems pretty similar to good old protocols…

Futures market is non-monotonic

<table>
<thead>
<tr>
<th>App.</th>
<th>Honest move</th>
<th>Affect what?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payment system</td>
<td>A does nothing, B sends X coins to C</td>
<td>B, C’s balance</td>
</tr>
<tr>
<td>E-Voting</td>
<td>... , B casts a vote</td>
<td>B’s vote</td>
</tr>
<tr>
<td>Reputation</td>
<td>... , B does something</td>
<td>B’s reputation</td>
</tr>
<tr>
<td>Futures market</td>
<td>A does nothing, B posts an order, --&gt; Market price changes</td>
<td>ALL positions including A’s → A can become BROKE</td>
</tr>
</tbody>
</table>
Futures market is non-monotonic

<table>
<thead>
<tr>
<th>App.</th>
<th>Honest move</th>
<th>Non-monotonic:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payment system</td>
<td>A does nothing</td>
<td>A does NOTHING but A’s crypto evidence of good standing is invalidated by B’s action (a good guy)</td>
</tr>
<tr>
<td>E-Voting</td>
<td>... , B casts a vote</td>
<td>B’s reputation</td>
</tr>
<tr>
<td>Reputation</td>
<td>... , B does something</td>
<td>B’s reputation</td>
</tr>
<tr>
<td>Futures market</td>
<td>A does nothing, B posts an order,</td>
<td>ALL positions including A’s → A can become BROKE</td>
</tr>
<tr>
<td></td>
<td>→ Market price changes</td>
<td></td>
</tr>
</tbody>
</table>

Why is This important? Technically

• We want to get rid of fat cat Sam doing financial intermediation → MPC of Alice and the Bobs
• Since MPC is costly → Replace MPC by ZK proofs → all asynchronous
  • Alice proves in ZK she is in good standing and sends it off
  • Bob I proves in ZK he is in good standing and sends it off
  • Bob II proves in ZK he is in …
  • They can verify all that asynchronously
  • Minimize MPC step to the crypto magic at the end
• A LOT MORE EFFICIENT → BUT MONOTONICITY NEEDED
  • Bob VIII won’t make Alice claims invalid
**Why is This Important? Economically**

- Fat cat Sam is gone
- Alice committed to buy from Bob III
- Enters Good Bob VIII
  - He wants to buy more oil $\rightarrow$ price surges
  - What happens to Alice? Has she cash enough?
  - Sam would call Alice to make sure she pour cash if price rises but Sam’s gone
  - Who is giving Bob III the money? Sam would but… Sam’s gone and Alice can’t foot the bill…

**Why is This Important? Combined…**

- Bob VIII did a honest protocolsl
  - He is perfectly honest and make a perfectly valid ZK
  - But he bankrupted honest Alice
- All Alice ZK proofs, committment credentials were
  - economically and cryptographically-valid (then)
  - but economically invalid (now)
- Monotonicity is destroyed $\rightarrow$ so is the possibility to replace MPC with straightforward ZK $\rightarrow$ need ad-hoc protocols
Solution Overview

- Confidentiality + integrity
  - Commitments + zk-proofs
- Anonymity
  - Anonymous network + Merkle tree
  - Spent/unspent tokens
- Non-monotonicity
  - Memoization
  - MPC only in checking positions

The Armchair Cryptographer Alternative

- Well, why bother???
  - Just run General MPC and job done
  - We know from the Sugar Beet Danish auction back in 2008 that MPC can stand thousands parties so what’s all that fuss?
- Questions:
  - Who actually read the sugarbeet auction paper?
  - 1229 farmers joined but how many Alices and Bobs (in security terms) joined the MPC protocol?
Crypto overhead for all traders

Lean Hog
Jan-Mar 2017

**Ok, MPC is expensive, but...**

- That’s the price for getting rid of middleman Sam
- Could we implement every step with MPC?
- Yes, but... everybody would do the same computation --- making 1 or 1000 orders
  - Alice and the Bobs are in the same boat aren’t they?
  - Well, Some Bobs are “slightly” more active...
**Frantic Alice and the Sleepy Bobs**

- TSX Market → 300K orders per day
  - 71% are Retail and Institutional Traders
  - 29% are Algorithmic Traders
- Proportions of orders
  - 82% of 300K orders by Algorithmic Traders
  - 99% of those orders are limit orders → never to be matched in an actual trade
    - Basically away from the current price
- But in MPC everybody does the same…

**Our hybrid protocol**

- Use mostly zk-proofs
  - Traders posting order
  - must prove
  - Passive traders
  - only verify
- Only use MPC for checking positions
**Our hybrid protocol**

- Crypto overhead for **retail** traders (few orders, mostly **passive**)
  - Passive traders
  - → only verify
  - Only use MPC for checking positions

1 day of trading would require almost 3 years to run on MPC

---

**Beyond security-with-abort**

- Malicious party can abort
  - Not joining MPC, not proceeding to match
- Can we stillmark to market?
- We need to penalize the mailcious party
- How?

- Claim-or-refund
  - [Kumaresan 2016]
  - **NO** → Uneven amount of deposit

- Lock-then-release
  - [Kosba 2016]
  - **YES** → Just lock the initial cash
Beyond security-with-abort (cont)

- Malicious party can abort
  - Not participate in MPC, not proceed to match
- Can we still mark to market?
- We can also penalize the malicious party
- Lock-then-release by [Kosba 2016]
  - Lock the initial cash
  - To join the mark to market → prove you did what you should have done it
  - Else lose the deposit & divide money among others

Evaluation + Optimizations

<table>
<thead>
<tr>
<th>Action</th>
<th>Trader</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init</td>
<td>11s</td>
<td>-</td>
</tr>
<tr>
<td>Post</td>
<td>39s</td>
<td>148s</td>
</tr>
<tr>
<td>Match</td>
<td>29s</td>
<td>148s</td>
</tr>
<tr>
<td>Mark</td>
<td>28s</td>
<td>-</td>
</tr>
</tbody>
</table>

Timing on AWS Large (128GB RAM Instance)

Many intermediate commitments
Streamline the # of commitments
→ Reduce zk-proof gen time

Combine MPC + penalty
1. MPC without consistency check
2. If there are broke traders, pick 1 volunteer to prove
1. If there are no broke traders, everyone prove
**Evaluation + Optimizations**

<table>
<thead>
<tr>
<th>Action</th>
<th>Trader</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init</td>
<td>11s</td>
<td>-</td>
</tr>
<tr>
<td>Post</td>
<td>39s</td>
<td>148s</td>
</tr>
<tr>
<td>Match</td>
<td>29s</td>
<td>148s</td>
</tr>
<tr>
<td>Mark</td>
<td>28s</td>
<td>-</td>
</tr>
</tbody>
</table>

**Take away**

- Distributed financial protocols are desirable, but ...
  - Financial protocol is not always monotonic
  - Viable protocol requires crypto effort proportional to activities
- FuturesMEX is feasible for low-frequency market
- There is more to do !!!
  - Hi-frequency market, e.g Eurodollar?
  - Dropout tolerant?
  - Dark pool: orders only visible conditionally
Acknowledgement

• Part of our research on FinTech and Blockchains has received funding from the European Institute of Innovation and Technology (EIT) through the UNBIAS Project of EIT Digital.
• This body of the European Union receives support from the European Union's Horizon 2020 research and innovation programme.
**Limits of “effective” MPC (SPDZ)**

- SHA-2p
- SHA-3p
- SHA-5p
- SHA-7p
- AES-2
- O-RAM-2p
- Dijkstra-2p
- Vickrey-2p

<table>
<thead>
<tr>
<th>MEMORY SIZE MB</th>
<th>TIME IN SECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>64</td>
<td>1024</td>
</tr>
<tr>
<td>256</td>
<td>4096</td>
</tr>
<tr>
<td>1024</td>
<td>16384</td>
</tr>
<tr>
<td>4096</td>
<td>65536</td>
</tr>
</tbody>
</table>

- 65GB Memory for SHA with 7 parties
- The Elephant in the off-line room nobody talks about
- Final Size of Circuits to be used in on-line room Reported in Papers

**Well, it’s off-line, so what?**

- **IF** you only vote **ONCE** or you only bid **ONCE** then it is off-line
- If every time “yet another Good Bob” does something Alice and the Bobs need to run an MPC round, well, it ain’t “off-line” anymore
  - For example if they need check Bob VIII bid didn’t bankrupt Alice and so Bob IX and so Bob X...
  - And you can’t recycle the off-line part (unless you’re so green to recycle also one-time pads...)

---

Presentatio at Aalto University - Security Seminar