The seconomics (security-economics) vulnerabilities of Decentralized Autonomous Organizations*

Fabio Massacci[1], Chan Nam Ngo[1], Jing Nie[1], Daniele Venturi[2] and Julian Williams[3]

[1] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
{massacci,channam.ngo,jing.nie}@unitn.it
[2] Computer Science Department, Sapienza University of Rome, Rome, Italy
venturi@di.uniroma1.it
[3] Durham University Business School, Durham, UK
julian.williams@durham.ac.uk

Abstract. Traditionally, security and economics functionalities in IT financial services and protocols (FinTech) have been perceived as separate objectives. We argue that keeping them separate is a bad idea for FinTech "Decentralized Autonomous Organizations" (DAOs). In fact, security and economics are one for DAOs: we show that the failure of a security property, e.g. anonymity, can destroy a DAOs because economic attacks can be tailgated to security attacks. This is illustrated by the examples of "TheDAO" (built on the Ethereum platform) and the DAOed version of a Futures Exchange. We claim that security and economics vulnerabilities, which we named *seconomics vulnerabilities*, are indeed new "beasts" to be reckoned with.

Keywords: seconomics vulnerabilities, FinTech, security protocols, Decentralized Autonomous Organizations

1 Introduction

Several researchers have traditionally assumed that security and economics functionalities are separate objectives. We argue that for "Decentralized Autonomous Organizations" (DAO) security and economics objectives should be considered as one. A failure of a security property is not simply an annoying part *outside the protocol* (e.g. law enforcement agencies knowing you are using Bitcoin to purchase porn or shady drugs). A *failure of a security property for DAOs may lead to the collapse of the entire economic functionality because such security attack could be combined with an economic attack*. We call such vulnerabilities *seconomics* vulnerabilities.

* This is just a taster (may be a bit detailed) to illustrate what we want to talk about.

In the past security vulnerabilities would translate to safety issues only for safety critical systems (when potentially exploited by terrorists, criminals or malicious governmental actors). Loosely speaking, for DAOs every vulnerability becomes a seconomics one. Different pieces of code are not just distributed but fully under the control of the *autonomous* entities. So at the same time we have the ability to subvert the system *and* the incentives to do so.

The organization of the remainders of the paper is as follows. We first give a general description of DAOs in §2. Next a popular DAO, TheDAO, and its hard fork as a result of an attack shortly after its launch are shown in §3. Then we present the DAOed version of the Futures Exchange (§4) followed by a possible security protocol (§5). A "Price Discrimination" attack mounted from anonymity failure is described in §6. Finally, we "conclude" the paper (§7).

2 Decentralized Autonomous Organization (DAO)

A DAO is a decentralized and allegedly "democratic" organization that is available on a distributed ledger through the combination of smart contracts and a rich scripting language, e.g. Ethereum [6]. Technically, a DAO is an implementation of a financial service by encoding the computations directly into smart contracts using the scripting language. The distributed ledger, e.g. blockchain, provides the secure environment to execute the computations and store the information across the whole network and hence eliminates the need of having a central trusted party.

Historically, Bitcoin [10] has been the first practical DAO that was launched as a payment transaction network in 2008. The applications of "Proof-of-Work" and "Blockchain" are the core components that allow Bitcoin to be decentralized [10]. Extensions of Bitcoin are later provided, e.g. ZeroCoin [9] as a coin washing service (later improved as ZeroCash [13] for private payments). Ethereum with a Turing-complete was the latest platform upon which DAOs could be built.

The first smart-contract-supported DAO, "TheDAO" was launched as a venture capital funding in May 2016. The crowd-funding was $150 million at peak value. TheDAO is supported by and stored entirely in Ethereum currency units (ETH). The objective of TheDAO was to create a venture capitalist fund designed to initiate other projects and demonstrate the creation of DAOs, see `daohub.org`.

Another DAO, Dash [5] also demonstrates great potential. The funding system witnesses quick growth in monthly revenue, from originally $14.000 per month in September 2015, to nearly $30.000 per month in March 2016.

3 The seconomics-TOCTOU attack on TheDAO

Perhaps when mentioning TheDAO, the most known event is the attack that happened shortly after its launch in June 2016. An unknown hacker was able to drain away 3.6 million ETH (which worthed $50 million at that time), approximately a third of the 11.5 million ETH that was committed into TheDAO.

That was a typical TOCTOU (Time of Check - Time of Use) vulnerability (see [15] for an introduction): an integrity violation by a race attack using a recursive call in TheDAO's implementation. This vulnerability could then be used to mount an economic attack on TheDAO. In economic term, TheDAO suffered from money pumps as TheDAO proceeded with account clearance prior to ledger update[4]:

> *The bug is that when splitDAO() is called, it will then call the recipients code to transfer Ethereum coin, after which the recipients code will call splitDAO() again before finishing. This causes the process to repeat itself, transferring more Ethereum coin, then calling splitDAO() again, which calls the hacker's code, which calls splitDAO(), which calls the hacker's code, and so on. The process will continue endlessly, until it drains all of TheDAO's coin.*

In this case, a security vulnerability (the user was authorized to draw money *only* in the *first* instance) has been combined with an economic attack (the recursive calls keep draining coin from TheDAO).

Several other attacks are possible on the Ethereum "smart contracts". The paper from [1] shows several of them. Yet the very paper fails to see that what is dangerous are not the vulnerabilities by themselves but the combination of the attack to the software with a tailgated economic attack.

Indeed most of the vulnerabilities classified in [1] as new types are classical vulnerabilities discussed, e.g. see [15] for concurrency and [2] for object oriented classes. For example the "call to the unknown" among the "Ethereum-specific" vulnerabilities is a classical problem of inheritance[5] dating back to faults about inheritance [4] where "long standing bugs have persisted because nobody thought to verify that deeply inherited methods [...] were overridden". By itself this would be a classical vulnerability. It becomes a seconomics one when a user can redefine a method that allows money to be sent or received.

Given the current level of enthusiasm over blockchains and the like in the FinTech sector we might as well assume that several other DAOs will emerge. The might be equally vulnerable (even if we assume integrity is not violated but just anonymity is) as we discuss in the next sections.

4 Another potential DAO: Futures Exchange

Futures Market are among the largest markets hence it is likely that the Futures Exchange will be DAOed.

[4] More detail on this hack can be found at http://blog.erratasec.com/2016/06/etheriumdao-hack-similfied.html.

[5] When invoking a contract at another Ethereum address this may have redefined its methods or the fallback method. Therefore the new redefined method will be called instead of the original expected method.

Table 1. Key Compositions and Characteristics of Futures Market

Traders Characteristics:	
Possible Positions	Buy-side traders holding long positions Sell-side traders holding short positions.
Possible Actions	Submit (Market/Limit Orders) and Cancel (Limit) Orders
Exchanges Main Functions:	
Price Discovery:	Disseminating the real-time market data to market participants; Providing a central limited order book: a consolidated tape with an electronic list of all the waiting buy and sell quotes organized by price levels and entry time.
Matching and Clearing	Matching engines use algorithms to match buy and sell quotes with a price and time priority principle. Clearing house is responsible for having a daily/ final settlement by the process of "mark-to-market".
Risk Managements	Traders need to deposit an "initial margin" and maintain a minimum funding in the "margin account" above the "maintenance margin"; otherwise, they will receive a "margin call" to request for additional funding. Any traders fail to meet to the minimum margin, will be forced to liquidate their open positions or even be "netted out" from the market.

A futures contract is a standardised legal agreement between two parties to purchase or sell an underlying asset at specified price agreed upon today with the settlement occurring at a future date.

Fundamental participants in a futures market include traders and exchanges (see Table 1). The central player of a futures market is a futures exchange. Futures contracts are negotiated at futures exchanges, which act as a central marketplace between buyer and sellers. The basic functions of the exchanges are to provide efficient price discovery process in their trading platforms, to match and settle trading activities, and to manage the risks for trading activities [14].

According to different trading positions, traders can be classified as buyers or sellers. Buyers take long positions by purchasing a certain amount of futures contracts, whereas sellers take short positions by selling a certain amount of contracts. The basic rule of trading in the futures market is buyers prefer to purchase contracts at lower prices and sellers prefer to sell contracts at higher prices.

A Futures Exchange DAO must maintain some key security properties:

Confidentiality of Inventory As the counterparty for each trader, exchanges are required to hold all the trading information and each traders identify, including the prices, volume, margin, order type etc. However, in order to maintain the economic viability, an exchange has to protect the trading information and traders credential without leaking to other opposite side traders.

Market Integrity Futures exchanges need to frequently monitor the trading activities (market prices and matching orders), the settlement capability (margin account) of each transaction to ensure the integrity of the marketplace. Many other attempts such as enforcing a maximum limit for a trader's long/short position, etc... are applied to protect market integrity.

Order Anonymity The exchange must prevent the linking of limit orders to uncover the trading strategy of a trader. This is done by the management of an anonymous central limit order book where only the bid and ask price in the order book is available for public. In this way, traders will not be able to identify the other traders and forecast others' trading strategies.

5 Security protocol for Distributed Futures Exchange

A security protocol for a Futures Exchange DAO could be built on a number of existing cryptographic primitives as follows.

Anonymous communication network e.g. Tor, recall that the futures exchange guarantees full anonymity of the traders. Since it is impossible to "create anonymity" from scratch, we assume an underlying anonymous network that hides the traders' identifying information (e.g., their IP address). This assumption was already used in several prior works, most notably [13].

Commitment Scheme and **Secure Addition** over commitments. We also assume **Zero Knowledge** ideal functionality for some standard NP relations for commitments as well as for exchange functionalities such as order fulfillment and mark to market.

Merkle Tree [8] where the leafs are commitments to anonymously commit and retrieve trader inventory as in [12,13].

The overall protocol should implement 4 phases of the "traditional" Exchanges:

Initialization Phase Every trader participating in the futures market has to commit a valid initial inventory (validity can be proven with the standard zero-knowledge proof for commitments).

Order Phase Every trader can post a new order or cancel a previously posted order. S/he will have to prove (possibly in zero-knowledge) that one has enough funds. Whenever a match happens, all traders will compute the new inventories, possibly with a secure multiparty computation and prove (again possibly in zero-knowledge) that they can afford the new liquidity profile of the market.

Margin Settlement Phase This phase is run immediately after the **Order** phase, in case one or more traders were unable to prove to hold a non-negative instant net position. The traders participating to this phase would see all their pending orders being canceled and their account billed for them.

Mark to Market Phase (At the end of the trading day, e.g. between 13:59:00 and 14:00:00) The traders locally update their inventory then commit the new inventory.

Such protocol can be engineered [7]. What we are interested in discussing is that what happens if some security properties fail.

6 The seconomics attack on Distributed Futures Exchange

It is sort of obvious that a failure of integrity may be dramatic to the protocol. We show that anonymity may also be essential.

A fully anonymous network is a quite strong assumption in the context of futures markets. In fact, the anonymous network, e.g. Tor, is not so reliable. It has been shown that traffic correlation attacks could be launched if the adversary control the entry or exit node and the server to deanonymize users [11,3]. Besides, as incentives would be quite strong (downloading porn or posting politically sensitive material is not the same as betting billions) we could assume anonymity would be violated. Considering this matter, we illustrate an attack that anonymity is no longer a matter of convenience. In fact, if anonymity fails, severe damage could be done to the Futures Exchange DAO and drive away the traders.

There is an obvious drawback as strategic actions by traders can result in traders being margin-called maliciously by prices well away from the "efficient" or "true" value of the contract.

For example, we assume Alice, Bob, Carol and Eve are in a futures market. We observe a situation as shown in Table 6

Table 2. Alice holds a short position of 90 contracts at $10 then price shifts to $16

Price = $10			
Trader	**Cash**	**Contracts**	**Position**
Alice	1400	-90	500
Bob	1200	30	1500
Carol	1200	30	1500
Eve	1200	30	1500

Price = $16			
Trader	**Cash**	**Contracts**	**Position**
Alice	1400	-90	*-40*
Bob	1200	30	1680
Carol	1200	30	1680
Eve	1200	30	1680

Alice accumulates a large short position of 90 contracts sell at $10 each, the other traders buy 30 contracts from Alice each at this price. Therefore, Alice's cash account is now $1400 where others are at $1200. For Alice, the inventory liquidation price is $X_{\mathsf{Alice}} = -90 \times (10 + \delta_P)$, and her net position is $N_{\mathsf{Alice}} = 1400 + X_{\mathsf{Alice}}$ where δ_P is the change in the contract price. When $\delta_P = 0$, the evaluation of her account stays the same (at $1000). When $\delta_P = 6$, her net position drops to -$40 and she has to be netted out.

If Alice wants to instantly liquidate her short position of 90 contracts, she has to buy such contracts from the market. Hence she commonly does so from the current standing sell limit orders in the market, this will inevitably be at a higher price (either slightly or markedly) than that shorted initially (as the very process of instigating the short was initially at or equal to the best available sell price), hence her instant net position is slightly worse than the traded price.

Carol and Eve, if they know that Alice is a small investor and needs cash, can generate an instant profit by changing the liquidity profile of the market. As

Alice's action space for new orders is limited, for example Alice posts a buy order of $9.50, the other traders can instantly set their buy orders at unilaterally high prices, pushing the liquidation price of the position higher. Alice can try to sell to those buy orders, but this pushes the contracts more deeply negative in a rising market exacerbating her problem of being close to the margin call. Eventually, the liquidation price, e.g. $16, is high enough that Alice's net position is below the margin call threshold and Alice is now cashed out, resulting in a realized payout to the other traders, i.e. her $500 is given to the other traders.

The other traders can then cancel their orders and the price could then decrease back to $10 or even lower (when Alice's trades would have been profitable), but Alice cannot benefit from this price as she has already been cashed out. The other traders do not have to actually trade anything, they have forced Alice to a margin call just by adjusting their buy quotes upwards strategically. The opposite problem can be generated from a long position and the market then being artificially deflated.

7 Conclusion (?)

The same problem of TheDAO might happen to the Futures Exchange DAO subjects to seconomics attack combining anonymity failure and price discrimination. Some parties may ask for the reversal of some transactions perceived as "unfair". But they will have no way to reverse them without changing the very system and network of participants. If enough people refused to join this would "balkanize" the market.

This leads to a central question: *"When the entire system collapses how could parties fix it?"* As TheDAO is distributed there is no way to actually "fix" the protocol backward as this would violate the other still standing security properties.

In the attempt to reverse TheDAO financial crisis, Ethereum designers proposed a solution *outside the protocol* itself, i.e. the hard fork: encourage parties to upgrade to a protocol client version that makes it impossible for the "hacker" to monetize the solution.

The attempts to fix the TheDAO proved difficult as to rewrite the central nexus of contracts forming the organization requires the majority of members to agree and this level of cooperation proved elusive. Indeed, a large fraction of the members of the Ethereum Community refused to join the new redressed ledgers, issued a Declaration of Independence[6] and continued to maintain the "classic" ledger:

> *Let it be known to the entire world that on July 20th, 2016, at block 1,920,000, we as a community of sovereign individuals stood united by a common vision to continue the original Ethereum blockchain that is truly free from censorship, fraud or third party interference.*

[6] Available at https://ethereumclassic.github.io/assets/ETC_Declaration_of_Independence.pdf.

We can therefore speculate that *seconomics vulnerabilities cannot be patched* as the economic damages they may cause are unlikely to be reversible by purely technical means.

Thus seconomics vulnerabilities are different "beasts" to be reckoned with.

References

1. Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts. Technical report, Cryptology ePrint Archive: Report 2016/1007, https://eprint. iacr. org/2016/1007, 2016.
2. Robert V Binder. Testing object-oriented software: a survey. *Journal of Software Testing Verification and Reliability*, 6(3):125–252, 1996.
3. Sambuddho Chakravarty, Angelos Stavrou, and Angelos D Keromytis. Traffic analysis against low-latency anonymity networks using available bandwidth estimation. In *European Symposium on Research in Computer Security*, pages 249–267. Springer, 2010.
4. Brad J Cox. The need for specification and testing languages. *Journal of Object-Oriented Programming*, 1(2):44–47, 1988.
5. Evan Duffield and Daniel Diaz. Dash: A privacy centric cryptocurrency. 2014.
6. Ethereum. A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2015. Accessed: 2015-12-30.
7. Fabio Massacci, Chan Nam Ngo, Jing Nie, Daniele Venturi, and Julian Williams. FuturesMEX: Secure Distributed Futures Market Exchange. In preparation, 2017.
8. Ralph C. Merkle. A digital signature based on a conventional encryption function. In *CRYPTO*, pages 369–378, 1987.
9. Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 397–411. IEEE, 2013.
10. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Unknown, 2008.
11. Gavin O'Gorman and Stephen Blott. Improving stream correlation attacks on anonymous networks. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 2024–2028. ACM, 2009.
12. Tomas Sander and Amnon Ta-Shma. Auditable, anonymous electronic cash. In *Annual International Cryptology Conference*, pages 555–572. Springer, 1999.
13. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.
14. Daniel F. Spulber. Market microstructure and intermediation. *The Journal of Economic Perspectives*, 10(3):135–152, 1996.
15. Junfeng Yang, Ang Cui, Sal Stolfo, and Simha Sethumadhavan. Concurrency attacks. In *Presented as part of the 4th USENIX Workshop on Hot Topics in Parallelism*, 2012.