Hostile blockchain takeovers (short paper)

Joseph Bonneau

New York University

Abstract. Most research modelling Bitcoin-style decentralised consensus protocols has assumed profit-motivated participants. Complementary to this analysis, we revisit the notion of attackers with an extrinsic motivation to disrupt the consensus process (Goldfinger attacks). We outline several routes for obtaining a majority of decision-making power in the consensus protocol (a hostile takeover). Our analysis suggests several fundamental differences between proof-of-work and proof-of-stake systems in the face of such an adversary.

1 Introduction

Bitcoin [15] has achieved significant popularity since its 2009 launch, with a monetary base nominally worth over US$100 billion at the time of this writing. Perhaps Bitcoin's most important innovation is its decentralised consensus protocol. Bitcoin-style consensus (or "Nakamoto consensus") uses computational (proof-of-work) puzzles to maintain consensus on the blockchain, a public append-only ledger storing all transactions to prevent double-spending. The computational puzzles are intended to make disrupting the consensus protocol expensive, as an attacker must obtain a large fraction of all computational power in the system to deviate from the default protocol. This basic design has been adapted in dozens of follow-up cryptocurrencies with similar consensus protocols, notably Ethereum [21] which is itself worth close to US$30 billion.

It was known from the start that an attacker with a majority of computational power can easily cause arbitrarily deep forks in the blockchain [15]. It has subsequently been shown that an attacker with substantially less power can, at the very least, undermine the fair distribution of rewards in the system [4,16,18]. These attack strategies are profitable in a fixed exchange-rate model (in which an attacker's utility is solely measured in currency units within the system itself). A similar modelling approach has been used in many papers [11,7,10,8,17,6] analyzing Bitcoin-style protocols with the goal of proving positive results about *incentive-compatibility*; that is, that given a specific utility model for miners intended properties of the system will emerge such as an ever-growing longest chain (stability) and proportional distribution of mining rewards (fairness).

An inherent limitation to this approach is that real-world attacks may negatively affect systems' value (and exchange rate with external currencies), making some mining strategies which deviate from the standard protocol to increase nominal miner revenue actually yield less utility. A more realistic utility function is revenue denominated in a stable external currency (such as US dollars).

Because accurately modeling the impact of miner behavior on exchange rates is difficult, analysis along these lines is usually qualitative. Thus, our ability to compare the stability of competing protocol flavours like proof-of-work and proof-of-stake remains limited.

In this work, we analyse the stability of consensus protocols from a different viewpoint. Rather than considering the risk of an attacker undermining desired properties to maximise utility, we consider an attacker whose explicit goal is to undermine and destabilise the consensus protocol. Kroll et al. [11] first considered such an attacker, which they called a Goldfinger attacker after the James Bond villain who attempted to irradiate US Treasury reserves. This attack model has received relatively little research attention since its proposal. Yet if the cost of undermining a currency system is low relative to the total value of currency in the system, the system may not be stable even if there were no motivation to mount such an attack.

Revisiting the dynamics of Goldfinger-style attacks is useful for two reasons. First, the potential motivations for a Goldfinger attack have become more plausible. Kroll et al. [11] hypothesised a government-sponsored attack, a social protest movement, or an attacker with a significant short position on the target currency's exchange rate. In the 4 years since, Bitcoin has received increased attention (often negative) from governments as well as social movements (particularly due to environmental concerns). Shorting a cryptocurrency is also more realistic as cryptocurrency option markets mature. Additionally, the plethora of cryptocurrencies in existence today provide a new motivation: under the simplifying assumption that various cryptocurrencies are competing for adoption from a fixed pool of cryptocurrency users and investors, eliminating a competing system might increase the value of a surviving system. For example, an investor with significant Bitcoin holdings might profit from undermining Ethereum, or undermining a fork of Bitcoin such as Bitcoin Cash.

Second, many variants of Nakamoto consensus are now deployed. In particular, there are now ASIC-mined blockchains (e.g. Bitcoin), GPU-mined blockchains (e.g. Ethereum) as well as many proposals for proof-of-stake and other variants. Goldfinger attacks provide an interesting comparison of these competing designs.

In the remainder of this work we analyze the difficulty of mounting a Goldfinger-style attack. We focus specifically on attacks in which an attacker obtains significant decision-making power (which we call *capacity*) and uses it to introduce forks in the system to cause significant damage, calling such an attack a *hostile takeover*. For simplicity, we focus on an attacker obtaining a majority, though of course significant damage may be done with less capacity. Other avenues for a Goldfinger attack exist which we do not consider, for example, denial-of-service attacks on the transaction relay network [20,9]. Our primary contribution is categorising the avenues for a hostile takeover and providing some basic analysis. We also posit several hypotheses about the difference in difficulty of mounting a hostile takeover against different variants of Nakamoto consensus.

		duration of control	
		temporary	**permanent**
source	**new**	Rent	Build
	existing	Bribe	Buy out

Table 1. Four basic strategies for gaining capacity in a Nakamoto consensus protocol.

2 Methods of obtaining capacity

An attacker aiming to take over a Nakamoto consensus protocol needs to obtain *capacity*, the details of which vary for different protocol designs. For a proof-of-work blockchain, they must obtain control of a large amount of computational capacity (similarly, a large amount of storage capacity for proof-of-space systems, and so forth). For a proof-of-stake blockchain, they must obtain control of a large amount of stake (currency) in the system.[1]

We consider two primary axes to compare methods of obtaining capacity:

- **New vs. existing capacity:** Is the attacker introducing new capacity into the system which was not previously used for the consensus protocol, or obtaining control of capacity already in use? Note that for proof-of-stake systems, the amount of capacity is fixed so it is not possible to introduce new capacity into the system.
- **Permanent vs. temporary control:** Is the attacker obtaining permanent control of the capacity, or only temporary control?

We can consider whether the attacker is obtaining mining capacity permanently or temporarily, and whether they are introducing new capacity into the system or capturing existing mining capacity. This yields four basic attack strategies, as shown in Table 1. We now consider each of these in turn.

2.1 Rental attacks

In a rental attack, the attacker temporarily obtains control of capacity external to the system. For example, an attacker might rent computational power or storage space from a cloud computing service or rent control of a large botnet. A key advantage of this approach is that the attacker has low up-front costs and no long-term liability. We note that this attack is impossible for proof-of-stake systems (as there is no external capacity to obtain).

Furthermore, rental is generally feasible only for blockchains in which the capacity is a commodity with external applications. Ethereum fits this description today as mining is dominated by graphics cards (GPUs).[2] Proof-of-storage [14,19],

[1] There are other potential attacks on proof-of-stake systems, such as purchasing keys from former stakeholders to induce a long fork (the "nothing-at-stake problem"). In this paper, we assume some solution exists for this problem and that a takeover requires obtaining a majority of the *current* stake in the system.

[2] In addition to rendering graphics, GPUs are now commonly used for a variety of tasks including scientific computing and machine learning.

proof-of-space [5] or proof-of-elapsed-time [3] are also candidates for rental. However, for ASIC-dominated proof-of-work blockchains, such as Bitcoin, the rent strategy is likely not possible because there is a negligible amount of Bitcoin mining hardware that is not already dedicated to Bitcoin mining.

Case study: Rental attacks on Ethereum. As a representative example we consider[3] the cost of renting GPU capacity from Amazon's Elastic Compute Cloud (EC2). Currently, Amazon rents machines with Nvidia K80 GPUs for about $1 per hour at spot prices, with bulk discounts available. These units are estimated to perform 50–100 MH/s. Therefore it might require renting about 1 million GPUs for a price of about $1 million/hr to perform a temporary takeover of the Ethereum blockchain. Even a few hours of disruptive attacks could be sufficient to cause a major loss in value to the system, which has a market cap of almost $30 billion. As a sanity check, Ethereum miners currently earn roughly $250,000/hr in mining revenue (from block rewards and gas fees), so renting capacity would not be profitable on its own, even with a considerable bulk discount.

2.2 Building attacks

In a building attack, the attacker permanently obtains new capacity. For example, the attacker might building a new mining farm. Again, this approach is not applicable to proof-of-stake, but is possible for all types of proof-of-work system.

Case study: Building attacks on Bitcoin. We consider the AntMiner S9, a state-of-the-art ASIC miner built with 16 nanometre features. It retails for about $2,000 and can perform about 14 TH/s (consuming over 1 kW of electricity). Given the Bitcoin network's current hash rate of roughly 10^{18} H/s, an upfront capital cost of roughly $1.5 billion would build enough capacity to take over the Bitcoin blockchain. Of course, this figure is approximate. It would be far cheaper to buy this hardware in bulk, however there would also be additional infrastructure and cooling costs when building a large mining farm.

Case study: Building attacks on Ethereum. For Ethereum, we consider the Radeon Rx Vega 56 GPU as an example mining card offering among the best performance for cost. Each card can perform about 36 MH/s and costs around $550. Although less powerful than the Nvidia units available for rent, lower unit costs mean the Radeon cards are more cost-effective. Given Ethereum's current hash rate of approximately 10^{11} H/s, this means an attacker must spend roughly $1.5 billion to build enough capacity to take over the Ethereum blockchain.

Comparison Interestingly, we obtain similar figures for a building attack against both Bitcoin and Ethereum—about $1.5 billion. This indicates there has been higher investment in Ethereum hardware relative to the system's total market

[3] Our case studies are based on market data as of November 2017. We leave all values approximate to two significant figures. All values are in US dollars.

value. There are two simple explanations: first, while Ethereum overall has a lower total value by a factor of more than three, the rate of revenue earned by Ethereum miners is relatively higher, about half that of Bitcoin. Second, nearly all current Bitcoin mining hardware was built specifically for mining Bitcoin, whereas Ethereum hardware may be acquired used or rented. Similarly, Ethereum miners may be more willing to invest in hardware knowing that they can sell if the system declines in value.

From either figure, we see a roughly thousand-fold increase between the cost of a building attack and the cost (per hour) of a rental attack against Ethereum. A building attack is also much slower and more logistically complex to execute. This is an argument in favor of ASIC-friendly mining puzzles as a defense against rental attacks.

2.3 Bribery attacks

In a bribery attack, the attacker offers payments to existing miners to deviate from the default protocol and mine on the attacker's branch. Note that we do not use the term "bribery" to indicate illegal or unethical behavior, simply that a side payment is being made. Several mechanisms for bribery have been proposed with various trust and risk properties [1,12]. For an example, an attacker might pay miners outside the protocol directly or through a negative-fee mining pool, or within the system by broadcasting anybody-can-spend transactions or transactions with abnormally high fees which are redeemable only on the attacker's branch. We suggest that it is also feasible for an attacker to create a smart contract to autonomously bribe miners working on another blockchain by checking that they have found blocks building on a designated starting point (similar techniques have been developed for implementing a mining pool in Ethereum [13]).

Previous analysis considered bribes motivated by executing a fork-and-double-spend attack (a "Finney attack"). In the simplest model, the attacker only needs to ensure that mining on the attack chain is more profitable than mining on the longest chain. Unlike renting or building attacks, the miner only needs to bribe half of the current capacity (rather than duplicating all of it), meaning about \$125,000/hr for Ethereum or \$250,000/hr for Bitcoin.[4] Of course, successfully executing a bribery attack may require paying a premium to override miner loyalty and convince miners to work on a fork that would be highly detrimental to the system. Though as argued previously [1] refusing to accept bribes representing a significant increase in revenue would be a tragedy of the commons. Presumably, similar dynamics would apply to proof-of-stake systems.

We note that bribery appears cheaper than even rental attacks and thus could be a significant threat to distributed consensus protocols. The cost is directly proportional to the rate of miner revenue, implying that even in a proof-of-stake system stability may require paying a non-trivial portion of the system's total value in fees. It has previously been argued [2] that Bitcoin may be unstable

[4] Note that we only consider bitcoin-denominated revenue. Many Bitcoin miners earn a small amount of additional revenue through merge-mining other currencies.

without the fixed block reward as rewards become time-varying. It also may be unstable simply because fees are too small relative to the value of the system.

2.4 Buy-out attacks

A buy-out attack would involve purchasing the majority of existing capacity from current owners. For proof-of-stake systems, the cost is half of the current monetary base,for example about $15 billion for Ethereum or $50 billion for Bitcoin. For proof-of-work systems, the cost should be about half of the net present value of all future mining rewards. It appears that proof-of-stake systems are much more secure here, as the attacker must buy half of all value of the system, whereas with proof-of-work the attacker must only buy half of the future mining rewards (which should be less than the entire market cap).

Traditionally, external buyers hoping to obtain a majority stake in a firm (in a hostile corporate takeover) must pay a premium over the current market price. This may not be true in a cryptocurrency buyout; in fact the opposite may hold due to the interesting possibility of a *race to the door* among current capacity owners. If an attacker can credibly commit to buying out half of all capacity and using it to destroy the system, current owners will have a strong incentive to sell to avoid being left in the 49% which does not sell and ends up holding worthless capacity. As the attacker gains more capacity (which is easy to authentically signal by including messages in block headers), the perceived likelihood of a successful attack increases. In response more owners may sell, potentially leading to a vicious cycle as owners race to avoid missing their chance to sell. This scenario does not occur in hostile corporate takeover because current shareholders retain (sometimes increased) value if they refuse to sell. The purchased firm will usually rise in market cap; if the firm's management do not believe the takeover will increase market value they can employ a wide variety of anti-takeover manoeuvres, none of which apply in a cryptocurrency takeover.

We observe that an attacker might credibly commit to a buy-out attack using a smart contract programmed to buy a large amount of stake through a reverse-price auction. This is similar to the suggested use of a smart contract above to implement bribery. Note, of course, that this is only feasible against a substantially smaller system, as the smart contract must be able to hold significantly more funds than the value of the target system.

Proof-of-stake systems are the most vulnerable to a race-to-the-door, since the stake has no value if the system crashes. ASIC-resistant proof-of-work systems appear less likely to suffer from a race-to-the-door, since capacity owners who do not sell to the attacker can still sell their hardware even if the attack succeeds. With ASIC-friendly proof-of-work, miners may retain some salvage value in unsold hardware, but this amount is likely small enough to ignore.

2.5 Countermeasures.

For all of the above attack models, there is the possibility of countermeasures by current capacity owners in the face of an attack. In theory, current owners can

deploy any of the applicable attack strategies themselves as a counter-measure, though it likely makes the most sense to respond in kind. In all cases, there is a collective action problem as all current owners would like to see the system continue, though there is no mechanism to compel them to contribute equally to defensive action.

The collective action problem is particularly acute for temporary (bribing or renting) attacks, as the temporary counter-measure yields no long-term benefit to those participating. In contrast, those responding to an attack by buying out or building will can benefit from the increased capacity for the future. Against a buy out, this may be a particularly lucrative (if the attack fails) a defensive buyer may profit as the currency gains value in light of a thwarted attack.

Proof-of-stake systems have one distinct disadvantage, which is that a successful buy-out attack will be permanent. In contrast, it is possible for proof-of-work protocols to recover from a successful attack by increasing total capacity, though significant damage may have already been done.

3 Discussion and open questions

The difficulty of hostile takeovers provides an interesting new lens for comparing decentralised consensus protocols. Our hope is that this manuscript is a starting point for further modeling and discussion.

Among proof-of-work systems, our analysis indicates a clear security advantage for ASIC-dominated mining, as rental attacks are not possible and existing miners should have more incentive to resist bribery attacks. However, the ability to rent capacity may be an advantage for ASIC-friendly mining in some cases.

Our model of ASIC-friendly proof-of-work is also simplistic, in that for Bitcoin there are now multiple competing systems (e.g. Bitcoin Cash) which use the same proof-of-work. These systems may effectively provide a pool of rentable mining capacity. It is also possible that rentable capacity exists from older mining hardware which is no longer profitable to operate, but may be operated at a loss by an attacker. This may be particularly dangerous as this capacity is essentially free to rent (or buy) as it has little other value.

At first glance, proof-of-stake systems appear less vulnerable to hostile takeovers than proof-of-work. They are not vulnerable to rental or building attacks. Bribery attacks appear similar, while buy-out attacks appear strictly more costly. However, proof-of-stake may be more fragile due to its vulnerability to an attacker inspiring a race-to-the-door. Additionally, renting or building new capacity is not available as a countermeasure.

Consistent with previous work, our analysis suggests bribery is a particularly troubling avenue of attack. Previous work suggested the problem that miner revenue is low relative to the potential profits to be had from double spending. We further suggest here that miner revenue is inherently low compared to the total value of the system and hence feasible for a Goldfinger attacker to match with relatively small bribes. It remains unclear what rate of miner revenue is required to ensure stability in practice.

References

1. Bonneau, J.: Why buy when you can rent? In: International Conference on Financial Cryptography and Data Security. pp. 19–26. Springer (2016)
2. Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A.: On the instability of bitcoin without the block reward. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 154–167. ACM (2016)
3. Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., Shi, W.: On security analysis of proof-of-elapsed-time (poet). In: International Symposium on Stabilization, Safety, and Security of Distributed Systems. pp. 282–297. Springer (2017)
4. Eyal, I., Sirer, E.G.: Majority is not Enough: Bitcoin Mining is Vulnerable. In: Financial Cryptography (March 2014)
5. Fuchsbauer, G., Park, S., Kwon, A., Pietrzak, K., Alwen, J., Gazi, P.: Spacemint
6. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol with chains of variable difficulty. In: Annual International Cryptology Conference. pp. 291–323. Springer (2017)
7. Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: EUROCRYPT (2). pp. 281–310 (2015)
8. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 3–16. ACM (2016)
9. Johnson, B., Laszka, A., Grossklags, J., Vasek, M., Moore, T.: Game-theoretic analysis of ddos attacks against bitcoin mining pools. In: International Conference on Financial Cryptography and Data Security. pp. 72–86. Springer (2014)
10. Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Blockchain mining games. Proceedings of the 2016 ACM Conference on Economics and Computation
11. Kroll, J.A., Davey, I.C., Felten, E.W.: The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In: WEIS (Jun 2013)
12. Liao, K., Katz, J.: Incentivizing double-spend collusion in bitcoin. In: Financial Cryptography Bitcoin Workshop (2017)
13. Luu, L., Velner, Y., Teutsch, J., Saxena, P.: Smart pool: Practical decentralized pooled mining. IACR Cryptology ePrint Archive 2017, 19 (2017)
14. Miller, A., Juels, A., Shi, E., Parno, B., Katz, J.: Permacoin: Repurposing bitcoin work for data preservation. In: IEEE Security & Privacy (2014)
15. Nakamoto, S.: Bitcoin: A peer-to-peer electionic cash system (2008)
16. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. IEEE EuroS&P (2016)
17. Pass, R., Shi, E.: Fruitchains: A fair blockchain. In: Proceedings of the ACM Symposium on Principles of Distributed Computing. pp. 315–324. ACM (2017)
18. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 515–532. Springer (2016)
19. Sengupta, B., Bag, S., Ruj, S., Sakurai, K.: Retricoin: Bitcoin based on compact proofs of retrievability. In: Proceedings of the 17th International Conference on Distributed Computing and Networking. p. 14. ACM (2016)
20. Vasek, M., Thornton, M., Moore, T.: Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. International Conference on Financial Cryptography (2014)
21. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151 (2014)