Analyzing the Potential Effects of Internet Censorship on the Bitcoin Network.

Jorge Coll, Andrew Fasano, Benjamin Kaiser, Lucy Qin
MIT 6.892: Shared Public Ledgers – Final Project
wehavenoprivacy@mit.edu, *fasano@mit.edu*, *benjamin.h.kaiser@gmail.com*, *lucyzqin@gmail.com*

***Abstract*—Abstract goes here.**

I. Introduction

The Bitcoin cryptocurrency relies on a shared, global transaction ledger that each participant can agree on and independently validate. When new blocks are created that can be added to this ledger, the blocks are broadcast over the Internet between nodes participating in the Bitcoin network. If all published blocks could be guaranteed to reach every other client in the network, all clients would always, eventually, have a consistent view of the network at a given point in time. However, this may not always be the case as a large number of Bitcoin users, such as those in China, are seperated from their peers by internet censorship systems. With specific messages embedded in a block published on the blockchain, it may be possible to force those censorship systems to inadvertently prevent censored minors from viewing a block the rest of the world agrees on. We set out to investigate how Internet censorship systems could prevent Bitcoin users reaching consensus across their borders. In particular, we analyze the feasability and impacts of an attacker embedding a malicious transaction into a block on the Bitcoin blockchain that would cause an internet censorship system to prevent that block from being visible to Bitcoin users behind the censorship system. We examine the current implementation of the Great Firewall of China as well as a hypothetical system that drops any TCP packets that contain a censored word.

We designed a set of simulations to analyze how Bitcoin clients would behave if TCP packets containing information about a block were unable to be sent between groups of users. Our simulations also examine the long-term impact on global blockchain concensus both when a majority of minors are uncensored or when the majoriy is censored.

II. Bitcoin's networking protocol

The message transfer protocol used in Bitcoin is defined abstractly in Nakamoto's original publication [1] and further fleshed out in `bitcoind`, the reference implementation he provided. In this section, we will provide a high-level overview of the relevant portion of the protocol.

The standard relay protocol used to share blocks between clients contains four types of messages:

- *inv*: When a transaction or block has been verified by a node, the node announces that the transaction or block is available by sending an *inv* message to all of their neighbors.
- *getdata*: When a node receives an *inv* message for a transaction or block that it does not already have a local copy of, it responds with a *getdata* message asking for that block.
- *tx*: When a node receives a *getdata* request for a given transaction, it responds with a *tx* message containing the transaction.
- *block* When a node receives a *getdata* request for a given block, it responds with a *block* message containing the block.

A node will only propagate a block or transaction to its neighbors if it can verify it. Unverified transactions or blocks are not propagated.

Partitions occur when two or more competing blockchain heads emerge. By itself, Bitcoin does not attempt to detect or mitigate partitions. If a partition appears due to two groups having different views of the current blockchain, both groups will continue operating independently and their state will diverge. If the divergence results in subnetworks that produce incompatible transaction histories, they will be unable to merge if the partition is later removed. Decker and Wattenhofer [2] observe that detection of network partitions may not be challenging and could be achieved through simple monitoring of the observed aggregate computational power in the network. As Bitcoin has grown in size and influence, such monitoring has begun to take place, and as of this writing it is possible to easily view the total network hashing rate via public websites [3].

III. Traffic Censorship

Censorship of Internet traffic occurs in a variety of forms all over the world. Most common is censorship of Web traffic (e.g., HTTP, DNS, and other TCP protocols related to browsing the Web). The most aggressive censorship takes place in countries where citizens' access to the Internet is very limited. Verkamp and Gupta [4] conducted experiments to infer the mechanics of various countries' approaches to Web censorship and also provided a set of independent sources ranking countries based on the severity of their censorship.

Across our research, four countries were repeatedly identified as the most severe censors: North Korea, Cuba, Iran, and China. Of these, China is the most interesting case for our purposes for two reasons: (1) its Internet-connected population far surpasses

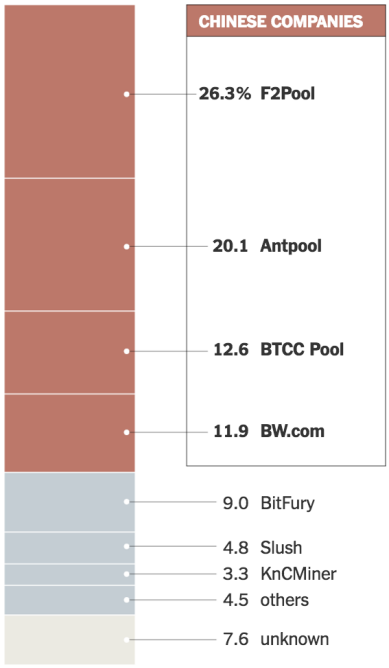

Fig. 1: From [5]: the shares of mined Bitcoin blocks from May 24 to June 24, 2016 by mining pool.

the others' and (2) it is a hotbed of Bitcoin mining activity, with Chinese-run mining pools accounting for an estimated 70% of all Bitcoin mining power as of June 2016 (Figure 1). As a collection, China's censorship measures are colloquially referred to as The Great Firewall of China (GFW).

IV. The Great Firewall of China

There is no consensus about the precise technical underpinnings of the GFW, as conflicting observations have been made. In particular, there is conflict about whether or not it is stateful – i.e., whether it stores and uses information about the packets it intercepts[1]. However, the basic mechanisms employed are known to be[7]:

- *IP address filtering*: The GFW blocks specific IP addresses from receiving traffic by dropping all packets associated with it. This assures that the GFW's reach extends to all content produced by a host's IP, rather than just the few specified domains.
- *DNS misdirection (hijacking)*: When requesting a blocked host name, there are cases in which the DNS servers under the GFW will return a different IP address than the one that corresponds to the domain name requested. The Chinese government can effectively replace the content with material that is more favorable to their interests.
- *Keyword filtering*: If a banned keyword appears in a URL, after a completed TCP handshake, the GFW will send reset packets to both the source and destination, blocking access to the requested content. Even if a keyword is not explicitly in the URL but appears within the HTML response, the content is also denied. In this particular instance, pages often begin to display but are then truncated after the discovery of a keyword.

[1]The most recent results, in Xu et. al[6], indicate that the GFW does record state

A. Impact on Bitcoin

The only direct impact that the GFW has on the Bitcoin network is a minor delay added to every packet. Due to this and normal Bitcoin block propagation delays, Chinese miners will hear about blocks mined outside of China later than ones within the country, causing a communication barrier. This actually creates a disadvantage for those outside of China since the majority of the hash power in Bitcoin is located inside of China. This causes problems, most recently during the debate surrounding a proposed block size increase. If the size were to increase, Chinese miners would be subject to further delays, potentially jeopardizing profits. [**?**]

V. Network Partitioning

Since traffic monitored by the GFW is subject to keyword filtering, we wondered if a situation could arise in which a banned word within a transaction could prevent a valid block from being propagated within China. This could lead to a partition in the network, causing a fork in the blockchain in which those outside of the GFW and those within it build off of different blocks. Given the current implementation of the GFW, only network activity over HTTP is monitored and Bitcoin activity is therefore not subject to filtration. Features could at any point be introduced within the GFW that allow for this vulnerability.

VI. Experiments

This section will describe the experiments we conducted. Our initial aim was to validate our hypothesis through a partition attack on the Bitcoin blockchain by creating a transaction containing a word censored by the GFW. When we determined that this was infeasible in practice, we decided to perform a set of experiments to analyze what effects such an attack might have on Bitcoin. The first simulator we ran abstracted away too many of the important details of the Bitcoin protocol, focusing instead on collecting high-level statistics about the network. Our next step was to try to understand the expected behavior of a Bitcoin client located inside of the GFW by directly inspecting the source code of `bitcoind`. Finally, we worked to deploy and run a more detailed simulator. Unfortunately, configuration issues prevented us from running our intended simulations, so at the moment they are left as future work. However, we do feel that through this process, we developed strong hypotheses about how the network would behave in the presence of a censorship-based partition, and as future work we intend to resolve our configuration issues with the simulator and test those hypotheses.

A. Experiment Design

To test our hypothesis, we needed to conduct two experiments to understand how the the GFW blocks traffic and how

the Bitcoin client behaves when certain blocks cannot be propagated between two groups of clients.

1) Firewall Testing: To check if the GFW would prevent the propagation of a Bitcoin block from crossing through its censorship systems, we obtained two machines: one in China and one in the United States. To confirm that our CN machine was affected by the GFW, we attempted to access several blocked websites and confirmed that they were inaccessible.

We attempted to send a variety of messages from our US machine to our CN machine and vice versa. We used the netcat UNIX utility to send ASCII messages. When we sent a message causing the GFW to block the connection, we expected the CN machine to be unable to reach the US machine on the same port for a few minutes, as described in **??**.

After validating that the GFW was affecting traffic between our two machines, we sent test messages of various formats containing known-blocked words. For each type of message, we recorded if the connection is blocked.

2) Simulation: By simulating a small-scale version of the Bitcoin network with network censorship in place, we can examine how a censorship-induced fork would affect the network over time. The majority of Bitcoin nodes run `bitcoind`, the reference client implementation of the Bitcoin protocol [8].

Data can be collected by simulating an entire bitcoin network with multiple nodes running `bitcoind` or by simulating the network traffic between nodes.

Bitcoin Simulator [9] built on top of NS3 [**?**] simulates network traffic between bitcoin clients.

Shadow [10] running with its Bitcoin plugin [8] emulates real `bitcoind` clients in a simulated network that can be used to conduct experiments.

The possible scenarios we wanted to examine fall into four categories:

1) A majority of mining power is unaffected by censorship, meaning that the minority of mining power behind the censor cannot send/receive messages containing a blacklisted word.
2) A minority of mining power is unaffected by censorship, meaning that the majority of mining power behind the censor cannot send/receive messages containing a blacklisted word.
3) A majority of mining power is unaffected by censorship (partition $P1$), a minority of mining power cannot send/receive messages to/from $P1$ containing a blacklisted word ($P2$), and a small number of miners is unaffected by censorship and can communicate to both Partitions 1 and 2 without any censorship ($P3$).

B. Expected Results

Suppose we have client C, a censored miner sitting behind censorship firewall, client U, an uncensored miner, and block B$_0, whichisthelastvalidblockknownbyallparties.NowsupposeU minesblock B_1 containingatransactionwithablacklistedkeyword.Ut$

To better understand what would happen if a node knew about the existence of a block which it could never retrieve, we examined the source code to bitcoind. In Bitcoin, clients learn of new blocks by listening for block header broadcasts. Once it hears of a new block header, the client attempts to download the block. The reference client keeps track of the active chain; however, it will only ever update the active chain once the block header and the full block are both received and validated. According to a comment in the source[2], this logic was introduced to defend against malicious miners trying to obtain a competitive advantage by broadcasting headers but witholding blocks. Therefore, we expect C to continue mining off B$_0, thepreviouslyknownblocktherebycausingafork.$

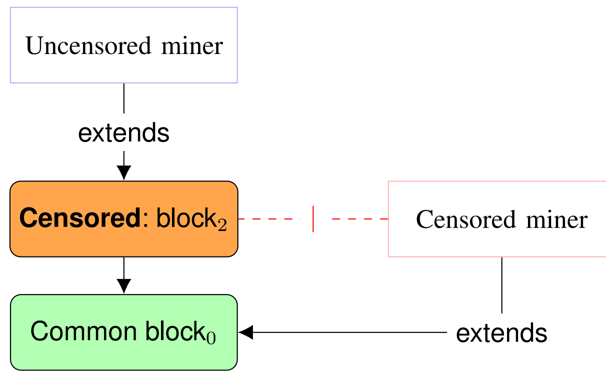

Fig. 2: After a censored block is mined, miners try extending different blocks

C. Results

1) Firewall Testing: To test the GFW we first had to obtain access to a system with a censored Internet connection. When a message caused the GFW to block a connection, the connection was reset and the CN machine was unable to connect to the US machine on the same port for a few minutes.

We attempted to construct a Bitcoin block that would be filtered by the GFW. Arbitrary data has been embedded into blocks before; the most common places that such data is stored in a block are the address fields (by encoding the data in hex), the coinbase transaction, or the scripts[3].

In the end we determined that we would need to trick the GFW into interpreting a block as Web traffic of some sort (e.g., DNS or HTTP), which was not feasible.

2) Simulation: Initial analysis of the Bitcoin Simulator built on NS3 showed that the simulator didn't high enough level of fidelity to conduct for our desired experiments. Instead, we focused on using Shadow...

We compiled version 0.9.2 of the `bitcoind` client[4] for use in Shadow. The published instructions for running Shadow's Bitcoin Plugin were significantly outdated, so we created a Docker container that builds their simulator and applies patches

[2]Specifically, line 2668 validation.cpp in bitcoind version 0.14: https://github.com/bitcoin/bitcoin/blob/0.14/src/validation.cppL2668

[3]In one notable case, a simple cross-site scripting (XSS) attack was embedded in a block and was apparently demonstrated to work on blockchain.info[11], although the site has since been patched to properly escape HTML.

[4]Shadow required us to use a slightly modified version of 0.9.2 available at https://github.com/amiller/bitcoin/tree/0.9.2-netmine

so that their original experiment can be recreated[5]. In our Docker container, we were able to run a network of 4 bitcoind clients communicating as we broadcast transactions and blocks to one of the clients.

We patched the `bitcoind` client to drop messages it recieves containing a "blocked" keyword.

We reconfigured Shadow to run a network with two groups of two nodes that were unable to communicate with each other. We then sent blocks to only one of these groups, expecting the clients in that group to have a different blockchain than the clients in the other group.We were unable to configure Shadow to run in this fashion due to time limitations.

If we were to get Shadow to seperately partition two network segments, we would have then modified how Shadow sends messages between clients to drop any messages between the two groups where the message contains our blocked keyword.

VII. Limitations

Speaking broadly, experiments performed on a model are only useful to the degree to which the model accurately represents reality. Even though we were not able to instantiate our model or run our simulations, we plan to do so in the future, so we want to make explicit the limitations of our model and the simplifying assumptions that we made when designing our simulations.

- *Mining pools*: In practice, mining pools in which miners cooperate to find new blocks and share the rewards are common [12]. We model mining pools as a single miner who has the cumulative hash power of the pool. This fails to account for scenarios in which miners outside of the GFW contribute to a pool that is managed by a node inside of the GFW.
- *Other limitations?*

VIII. Future Work

IX. Discussion

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system."

[2] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing*, 2013.

[3] [Online]. Available: http://bitcoin.sipa.be/

[4] J.-P. Verkamp and M. Gupta, "Inferring mechanics of web censorship around the world," in *Presented as part of the 2nd USENIX Workshop on Free and Open Communications on the Internet*, 2012.

[5] N. Popper, "How china took center stage in bitcoin's civil war," The New York Times, June 2016.

[6] X. Xu, Z. M. Mao, and J. A. Halderman, "Internet censorship in china: Where does the filtering occur?" in *Proceedings of Passive and Active Measurement: 12th International Conference*, 2011.

[7] M. Hu. (2011, May) The great firewall: a technical perspective. Torfox: A Stanford Project. [Online]. Available: https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreedomOfInformationChina/great-firewall-technical-perspective/index.html

[8] A. Miller and R. Jansen, "Shadow-bitcoin: Scalable simulation via direct execution of multi-threaded applications," in *8th Workshop on Cyber Security Experimentation and Test (CSET)*, 2015.

[9] [Online]. Available: https://arthurgervais.github.io/Bitcoin-Simulator

[10] R. Jansen and N. Hopper, "Shadow: Running tor in a box for accurate and efficient experimentation," in *Proceedings of the 19th Symposium on Network and Distributed System Security (NDSS)*. Internet Society, February 2012.

[11] [Online]. Available: https://redd.it/1n57uj

[12] M. Rosenfeld, "Analysis of bitcoin pooled mining reward system," 2011.

[5]The Dockerfile that builds this container is available at https://github.com/AndrewFasano/shadow-bitcoin-docker