SECURITY OF BLOCKCHAIN

RYAN MORENO

1 Background and Motivation

"Blockchain" was just a buzzword to me, so I chose the topic to learn how it works. Blockchain is a form of record keeping, marked by its security, privacy, and permanence. Currently, Blockchain is largely used for cryptocurrency. However, it has implications beyond this field, including voting and smart contracts (well-defined contracts that are played out automatically).

Blockchain derives its name because it stores its permanent record as a chain of blocks. Blockchain is decentralized, meaning many copies of the chain are stored across the network of users' devices. Decentralization offers two main advantages. Primarily, the record is public which maintains correctness because any user can view and verify it. Also, having copies of the chain reduces the damage of a potential security attack; even if an attacker destroys the record in one spot, it remains safe in many others.

Bitcoin is the best known application of Blockchain. Bitcoin is a cryptocurrency that has a total value of around 41 billion USD as of June 2019. Along with the benefits of decentralization, Blockchain is useful because the record of transactions cannot be altered. Whereas banks can bounce and return payments, once a Bitcoin transaction has been confirmed, it is permanent. Finally, Bitcoin has superior privacy. Although the transactions are public records, each transaction is encrypted, only linking a transaction with the users' anonymous public keys. Due to this anonymity, Bitcoin is often associated with illegal activity. However, the same privacy, security, and permanence that lends Blockchain to criminal activity is beneficial in many positive applications such as secure banking and voting.

2 Validating Transactions

Whenever a user wants to put a transaction on record, they broadcast this data to the entire network. Any user can verify the data's authenticity and attempt to add it within the next block. Since any user can add a block, an attacker could add an inauthentic block, perhaps one creating currency out of thin air or transferring currency from another user to themselves. However, the other users would be able to tell the block was inauthentic and would create a fork starting from the previously verified block.

In order to authenticate a block, a good actor needs to check that the sender requested the transaction. This is verified using RSA with the transfer request's digital signature. Each user has a private key d that only they know and a public key e that everyone on the network knows. These two keys are chosen such that they are inverses of each other mod n. This means that the sender can create a signature $S = T^d \bmod n$, where T is a a hash of the transfer request. Any user can then verify the signature by checking if $T = S^e \bmod n$. If this equivalence holds, the user requesting the transaction has shown that they know the sender's private key d, proving their identity.

We also need to check that the sender actually owned the coins they sent. This can be verified by simply scanning previous transactions (kept in the chain of blocks) to ensure that the sender had at some point been sent the coins. We also should check that the sender hasn't already spent these coins (an attack called "double spending"). Every transaction record contains a hash of the transaction in which the sender received the coins they're sending, providing a proof of ownership. Because every transaction's hash value is unique, we can scan previous transactions to ensure that no other transaction contains the same proof of ownership.

3 Adding Blocks and Proof of Work

When a block is added, it is assigned an ID by hashing a combination of the block's content and the previous block's ID. In this way, the blocks are chained together. In order to add a block to the chain, a user must fulfill the consensus protocol. If they don't show proof of fulfilling the protocol, the other users won't accept their version of the Blockchain. One important consensus protocol, used by Bitcoin, is Proof of Work, in which a user must solve a time-consuming problem. In the case of Bitcoin, the difficult problem is to find a value that, when combined with the proposed block information, hashes to produce a number beginning with a long sequence of zeroes. Because hash functions produce values in a random distribution, finding a sequence starting with n zeroes should take 2^n tries on average.

Interestingly, Bitcoin self-regulates the difficulty of the problem by adjusting n. The ideal pace of chain growth has been deemed to be 6 blocks per hour, allowing adequate time for the new information to propagate the network and limiting the storage required for the entire Blockchain. Every two weeks, the actual number of average blocks per hour is compared to this goal, and the value of n is adjusted accordingly.

4 Security from Attacks

We've already shown that validating transactions prevents adding inauthentic blocks. The other potential attack is altering an old record in which the attacker paid person A to say that they paid person B or removing the transaction entirely. Because the blocks are chained together, if an attacker alters a block, it will also alter the next block's ID, and the attacker will need to re-solve the time-consuming problem for the next block. In order to change a block that is buried under z other blocks, the attacker would need to recreate each of the z blocks as well. If there is ever a discrepancy in the Blockchain, the users are to assume that the longest chain is the correct one. Thus, assuming that the majority of the users are good actors, the authentic Blockchain will be taken as valid because it will outpace the attacker's inauthentic one.

Let's consider w_i, the probability that an attacker can ever overtake the good actors if the attacker starts z blocks behind. Assume that the attacker begins with i blocks, z blocks behind the good actors. When the attacker creates a block, we consider the attacker as gaining a block to their chain. When the good actors create a block, we consider the attacker as losing a block (because they are getting further behind). The attacker will win if they reach $N = i + z + 1$ blocks, just beating the good actors. To simplify the equations, the attacker will quit if they lose i blocks, reaching 0. We will eventually consider what happens as i approaches infinity, meaning that the attacker never quits.

We say $w_0 = 0$ (because the attacker has already lost all their starting blocks and quits) and $w_N = 1$ (because they have already caught up). The probability that the attacker can win starting with i resources is the probability that they make the first block times the probability that they can win starting with $i + 1$ resources (since they just caught up by one) plus the probability that they lose the first block times the probability that they can win starting with $i - 1$ resources (since they just got further behind by one).

p = fraction of computing power belonging to good actors
$q = 1 - p$ = fraction of computing power belonging to the attacker

$$w_i = q \cdot w_{i+1} + p \cdot w_{i-1}$$

Algebraic manipulation yields the following.

$$(p + q) \cdot w_i = q \cdot w_{i+1} + p \cdot w_{i-1} \qquad (p + q = 1)$$

$$w_{i+1} - w_i = \frac{p}{q} \cdot (w_i - w_{i-1})$$

Specifically,

$$w_2 - w_1 = \frac{p}{q} \cdot (w_1 - w_0) = \frac{p}{q} \cdot w_1 \qquad (w_0 = 0)$$
$$w_3 - w_2 = \frac{p}{q} \cdot (w_2 - w_1) = (\frac{p}{q})^2 \cdot w_1$$

We can generalize this pattern for w_{i+1}.

$$w_{i+1} - w_i = (\frac{p}{q})^i \cdot w_1$$

Consider $w_{i+1} - w_1$.

$$w_{i+1} - w_1 = (w_{i+1} - w_i) + (w_i - w_{i-1}) + \ldots + (w_2 - w_1)$$
$$= \sum_{k=1}^{i} w_{k+1} - w_k = w_1 \sum_{k=1}^{i} (\frac{p}{q})^k$$

$$w_{i+1} = w_1 \sum_{k=0}^{i} (\frac{p}{q})^k = w_1 \frac{1 - (\frac{p}{q})^{i+1}}{1 - \frac{p}{q}} \qquad \text{(geometric series)}$$

Specifically, knowing $w_N = 1$,

$$w_N = 1 = w_1 \frac{1 - (\frac{p}{q})^N}{1 - \frac{p}{q}}$$
$$w_1 = \frac{1 - \frac{p}{q}}{1 - (\frac{p}{q})^N}$$

Plugging w_1 into the equation for w_{i+1}, we get the following.

$$w_{i+1} = \frac{1 - (\frac{p}{q})^{i+1}}{1 - (\frac{p}{q})^N}$$
$$w_i = \frac{1 - (\frac{p}{q})^i}{1 - (\frac{p}{q})^{i+z+1}}$$

Now we consider $i \to \infty$, representing the attacker never giving up. We assume $q < p$, meaning the attacker holds a minority of the computing power in the network.

$$w_i = \frac{1 - (\frac{p}{q})^i}{1 - (\frac{p}{q})^{i+z+1}} = \frac{(\frac{p}{q})^i((\frac{p}{q})^{-i} - 1)}{(\frac{p}{q})^i((\frac{p}{q})^i - (\frac{p}{q})^{z+1})} = \frac{(\frac{p}{q})^{-i} - 1}{(\frac{p}{q})^i - (\frac{p}{q})^{z+1}}$$

We assume $q < p$, so $\lim_{i\to\infty}(\frac{p}{q})^- i = 0$,

$$\lim_{i\to\infty} w_i = \frac{-1}{-(p/q)^{z+1}} = (\frac{q}{p})^{z+1}$$

This demonstrates that if the majority of the network's computing power is controlled by good actors, the probability of an attacker successfully altering a block decreases exponentially as the block gets buried. When can a merchant receiving payment be confident that the transaction is unalterable? In order to be 99.9% sure that the transaction is permanent, the sender would need to wait for 340 blocks if the attacker owned 45% of the network's computing power. If the attacker owned only 10%, then the sender would need to wait for only 5 blocks. Based on Bitcoin's average block rate, if any single attacker owns only 10% of the network's computing power, the sender can be extremely confident in the security of the transaction after only 50 minutes.

What if the attacker owned the majority of the computing power? If $q > p$, $\lim_{i\to\infty}(\frac{p}{q})^i \to 0$, so $\lim_{i\to\infty} w_i = 1$. This means that if the attacker holds the majority of the computing power, they will be able to corrupt the Blockchain by catching up with the good actors. This is called a 51% attack. However, if the attacker owns more than half of the computing power, they would likely make more money by using that computing power to reap the transaction fees from adding authentic blocks than they would make from corrupting the Blockchain, losing user trust, and thereby devaluing the Bitcoin they just stole. In this way, Blockchain's security is upheld by both mathematics and economic incentive.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", October 2008. [Online.] Available: https://bitcoin.org/bitcoin.pdf.

[2] "How Blockchain Works". c2019. Lisk; [accessed Septempber 21, 2019]. https://lisk.io/academy/blockchain-basics/how-does-blockchain-work.

[3] "Money, Banking, and Central Banks: Bitcoin". c2019. Khan Academy; [accessed September 21, 2019]. https://www.khanacademy.org/economics-finance-domain/core-finance/money-and-banking/bitcoin/.

[4] A.P. Ozisik and B. N. Levine, "An Explanation of Nakamoto?s Analysis of Double-spend Attacks", January 2017. [Online.] Available: https://arxiv.org/pdf/1701.03977.pdf.

[5] "Gambler's Ruin Problem". Updated April 2019. Columbia University: Karl Sigman; [accessed September 23, 2019]. http://www.columbia.edu/ ks20/FE-Notes/4700-07-Notes-GR.pdf

[6] Burt Kaliski, "The Mathematics of the RSA Public-Key Cryptosystem", RSA Laboratories, 2006. [Online.] Available: http://www.mathaware.org/mam/06/Kaliski.pdf.

[7] "How Much of the World's Money is In Bitcoin?". June 2019. Investopedia; [accessed September 23, 2019] https://www.investopedia.com/tech/how-much-worlds-money-bitcoin/