New Probability Analysis shows that Sub 51% Attacks are Profitable.

Presented at

Western Economic Association International 2019

Jaehyuk Jang and Heung-No Lee
GIST, South Korea
Home page: https://infonet.gist.ac.kr
Facebook/Publication ID: Heung-No Lee

Flow of Talk Today

- Profitable DS Attacks
  - DS Attacks
  - Nakamoto's Results
  - New Mathematical Results

- This presentation is based on the work of *Profitable Double Spending Attacks, co-authored by Jaehyuk Jang.*

https://arxiv.org/abs/1903.01711

Goal

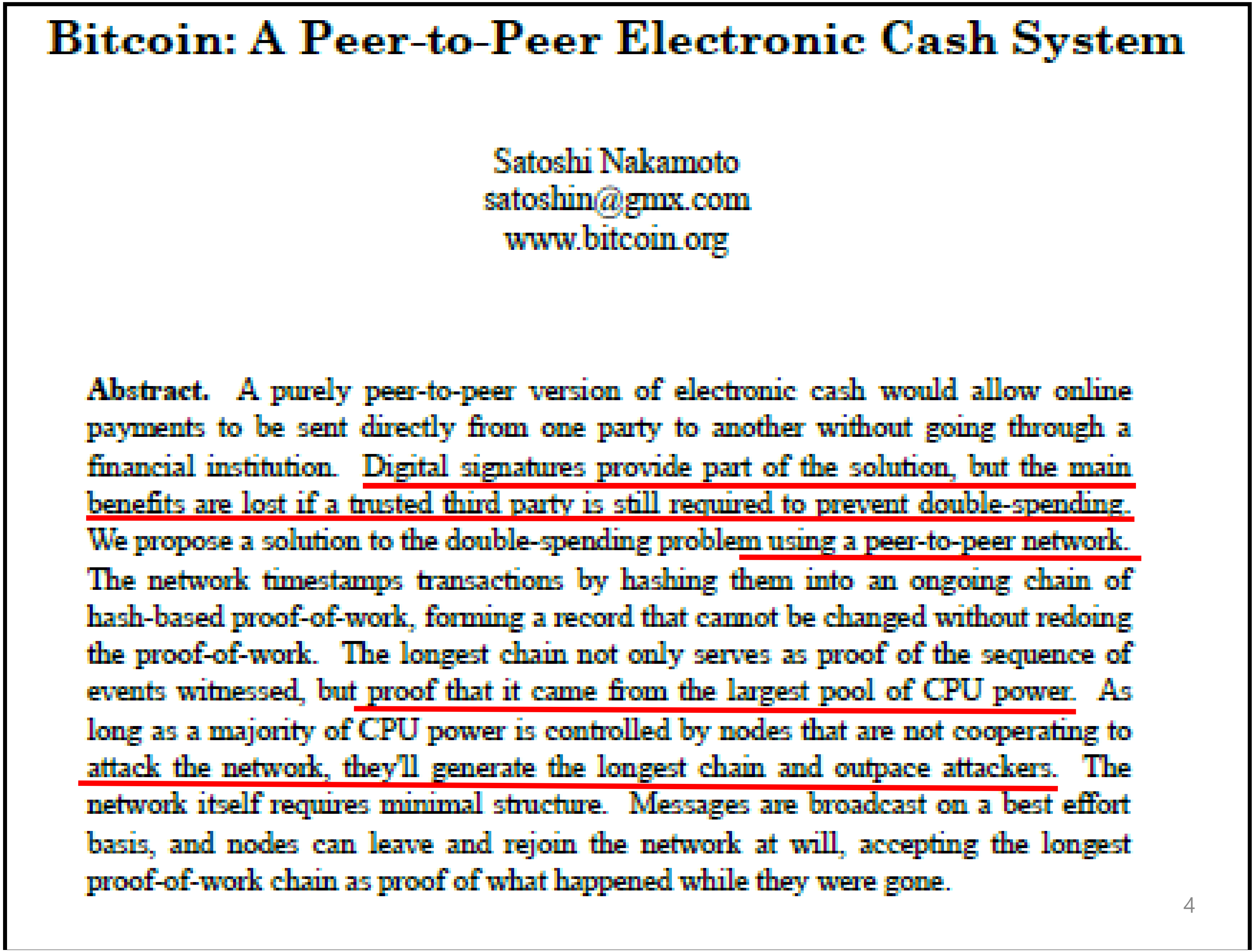
- To analyze double-spending (DS) attacks against public blockchains.

- Satoshi Nakamoto

  "DS attack is **difficult** since its success **requires 51%** of network's total computing power."

- We show

  "DS attacks are **threatening even** with **less than 50%**."



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Profitable Double-Spending Attacks

Jehyuk Jang and Heung-No Lee, *Senior Member, IEEE*

Abstract—**Our aim in this paper is to investigate the profitability of double-spending (DS) attacks that manipulate a priori mined transaction in a blockchain. Up to date, it was understood that the requirement for successful DS attacks is to occupy a higher proportion of computing power than a target network's proportion; i.e., more than 51% proportion of computing power. On the contrary, we show that DS attacks using less than 50% proportion of computing power can also be vulnerable. Namely, DS attacks using any proportion of computing power can occur as long as the chance to making a good profit is there; i.e., revenue of an attack is greater than the cost of launching it. We have novel probability theory based derivations for calculating time finite attack probability. This can be used to size up the resource needed to calculate expected attack cost and expected attack success time. The results enable us to derive sufficient and necessary conditions on the value of a target transaction which make DS attacks for any proportion of computing power profitable. They can also be used to assess the risk of one's transaction by checking whether or not the transaction value satisfies the conditions for profitable DS attacks. Two examples are provided in which we evaluate the attack resources and the conditions for profitable DS attacks given 35% and 40% proportions of computing power against *Syscoin* and *BitcoinCash* networks, and quantitatively shown how vulnerable they are.**

Index Terms— **Blockchain, Bitcoin, Double-Spending**

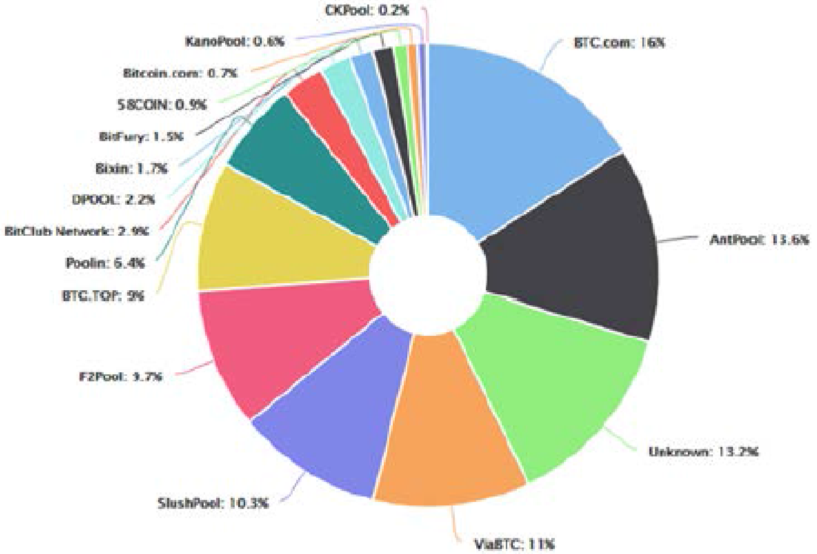


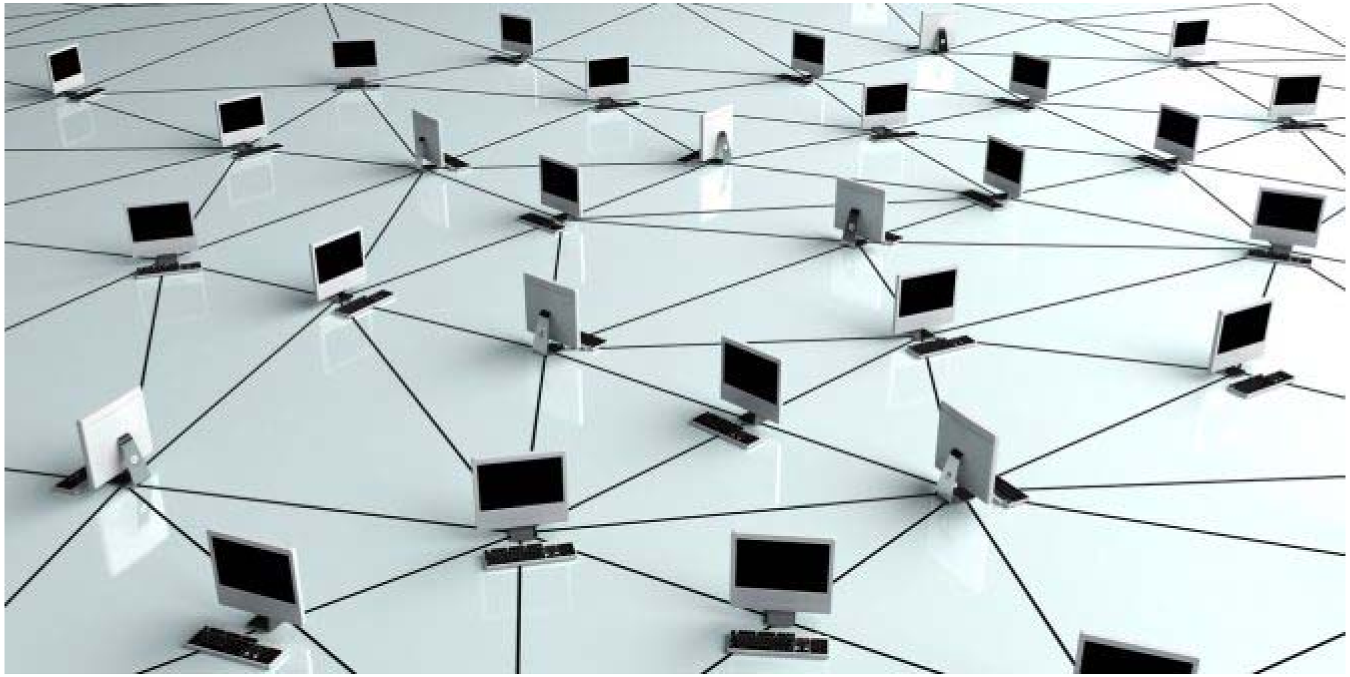
Fig. 1. Computation power distribution among the largest mining pools provided by *blockchain.com* (date accessed: 22 Oct. 2018).

succeeds in generating a new block, he/she has the latest version of the chain. All of the peers continuously communicate with each other to share the latest chain. If a peer suffers from a conflict between two or more different chains, the consensus rule provides a rule that a single chain is selected. Satoshi Nakamoto suggested the *longest chain consensus* for *Bitcoin* protocol which conserves the longest chain among the conflictions [1]. There are also other

How does Bitcoin work?

Bitcoin uses the internet.

Bitcoin attracts P2P nodes.



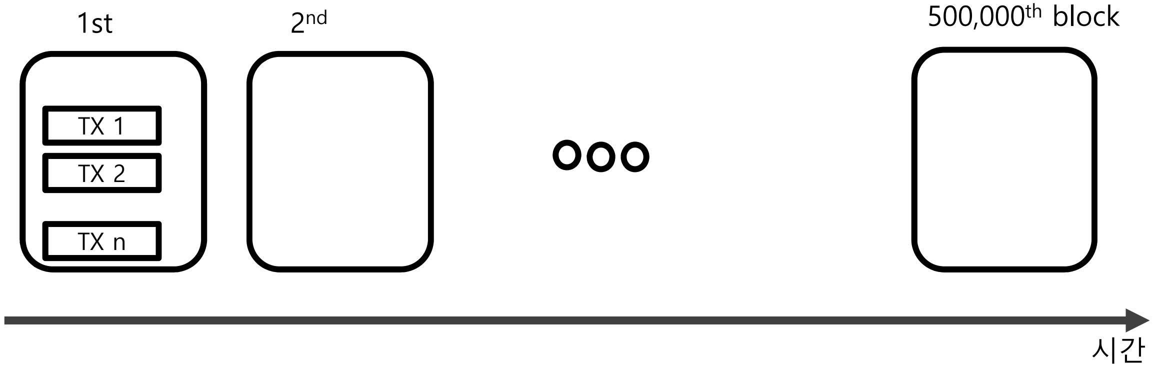
P2P nodes share a blockchain.

- Blockchain is to mean a digital leger:

  Blockchain is a chain of blocks.

  Each block is time stamped.

  Each block stores TXs.
- Blockchain also implies the technology itself.



The blockchain is left open for viewing.

- The digital ledger is left open.
- Anyone can talk to a node and view the ledger. (Public Blockchain)

These ledgers are the same except the most recent blocks.

A ledger in America

A leger in Korea

Any new node can join.

- In the public blockchain network, anyone can join and become a guard (miner).

Miners are everywhere.

- Each block is formed by a node.
- A node gathers TXs, validates them, forms a block.
- As a reward, the node which formed a block is given a block mining reward (e.g. 12.5 BTC).
- Thus, they are called miners.

502th block Formed in China

500th block Formed in America

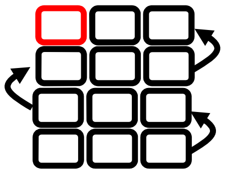
501th block Formed in Korea

Consensus mechanism plays the key role in blockchain.

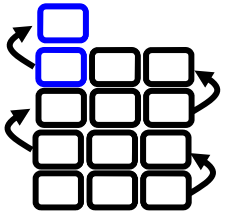
- Multiple different chains are possible, as miners work independently.
- When any two chains are available, miners choose the longer one!

Which one wins when there are two chains announced?

100th block mined
Hooray!



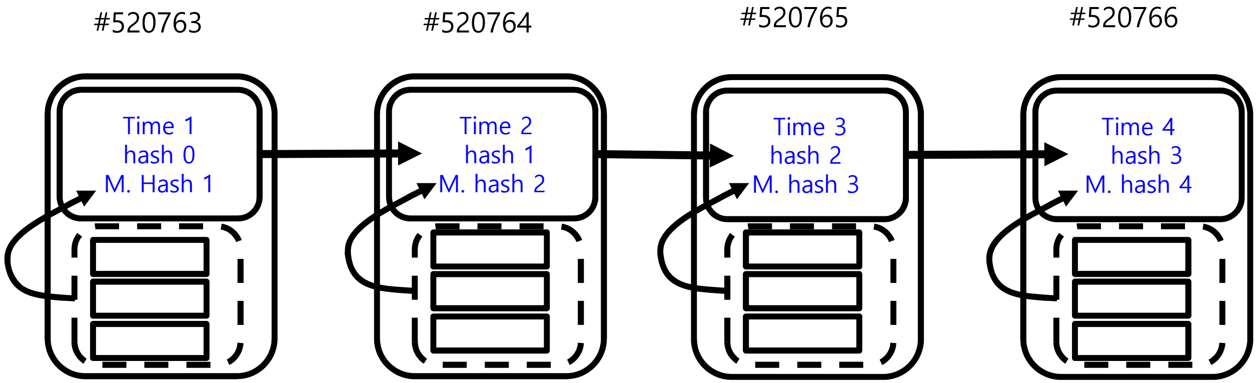
101th block mined
Hooray!



Longer chain wins!

Blocks are cryptographically chained.

- Blocks are cryptographically chained.
- Any alteration made to the content can be easily noticed.

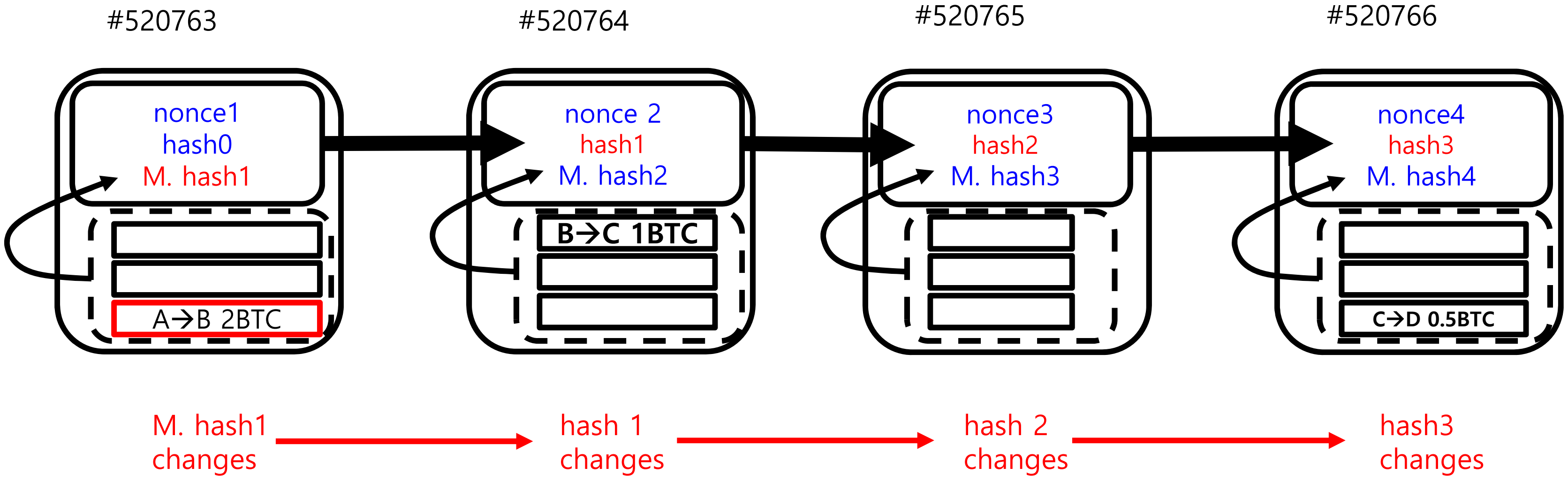


Secure hash function F
F(file) = hash

256 bit string

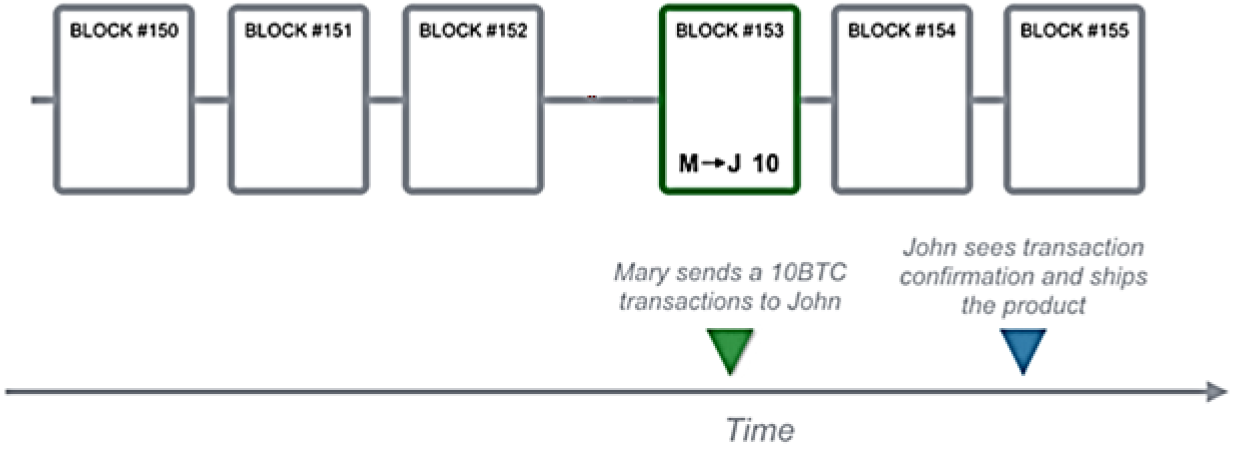
Content in the blockchain cannot be changed (easily).

- What happens when any alteration is made?
- Any alteration is easily noticeable!
- If an unnoticeable change were wanted, the whole alteration would be needed.
- The whole alteration is to redo all the hashes of the subsequent blocks.
- Proof-of-Work (PoW) is imposed on the chain and thus the whole job cannot b e redone easily.
- Immutability and openness allow one to transact with the other over the interne.
    - A→ B 2 BTC
    - B → C 1 BTC
    - C → D .5BTC



Transaction: M -> J 10 BTC

- Mary sends Bitcoin(BTC) to John.

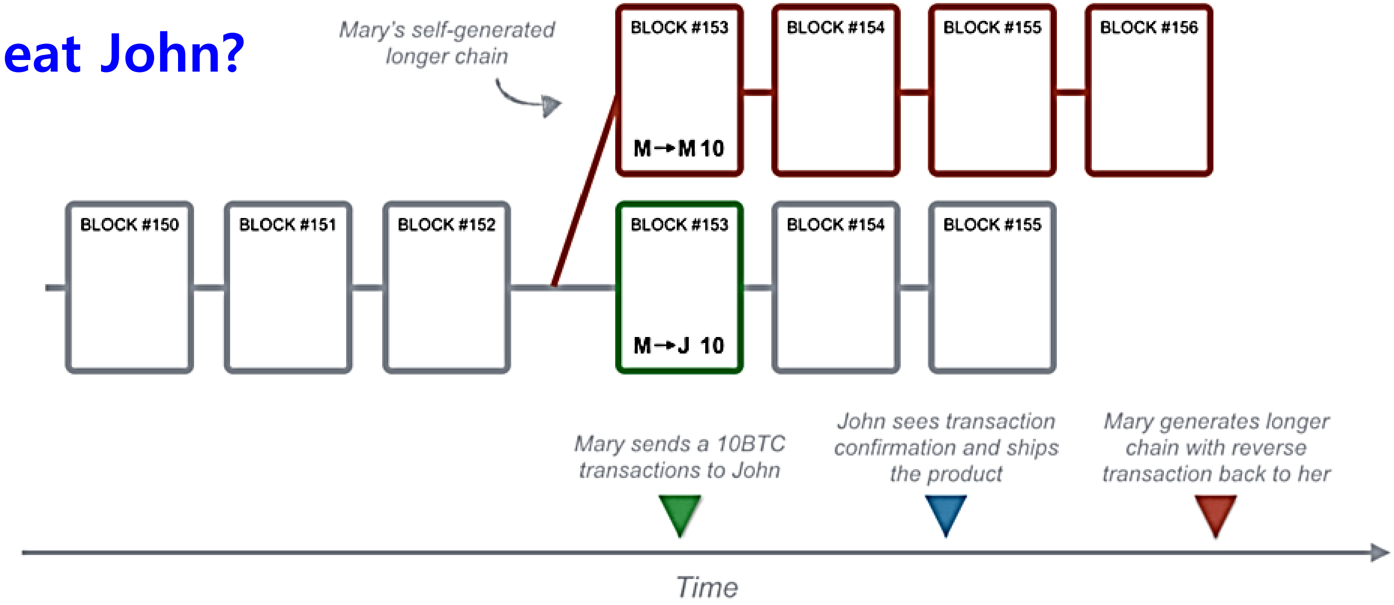

BLOCK #150
BLOCK #151
BLOCK #152
BLOCK #153
M→J 10
BLOCK #154
BLOCK #155
Mary sends a 10BTC transactions to John
John sees transaction confirmation and ships the product
Time

$N_{BC} = 3$

1. Transaction: "Mary → John 10 BTC"
2. Miners put valid transactions into a block which is chained.
3. John checks the number of blocks by which his transaction has been confirmed, "***block confirmation number***."
4. John ships the product to Mary after 2 more blocks, for example.

Double-Spending Attack

Can Mary cheat John?
But how?


Mary's self-generated longer chain
BLOCK #153
M→M 10
BLOCK #154
BLOCK #155
BLOCK #156
BLOCK #150
BLOCK #151
BLOCK #152
BLOCK #153
M→J 10
BLOCK #154
BLOCK #155
Mary sends a 10BTC transactions to John
John sees transaction confirmation and ships the product
Mary generates longer chain with reverse transaction back to her
Time

1. Target transaction "Mary → John 10 BTC" was put inside Block #153."
2. Miners verify the target transaction and put into a block, and continue.
3. Mary builds her own chain in secret.
4. Mary's chain contains a fake transaction that nullifies the target transaction.
5. Mary announces her fake chain to the public if
    1) John has completed shipping the product to Mary, AND
    2) Mary's fake chain has grown longer than the public chain.
6. The public adopts Mary's chain, since hers is longer than their own chain.

What happens when attacker has not enough blocks?

- NBC = 3
- Attacker’s chain is shorter than public chain:
- By 1 block
- By 2 block
- …

Markov Decision Process

- Race between the two chains
- Let S(t) := A(t) – H(t)
- Then,
- S(t) = -1
  - attacker’s chain is shorter by 1
- S(t) = -2
  - attacker’s chain is shorter by 1

Markov Decision Process

- Race between the two chains
- Let S(t) := A(t) – H(t)
- Then,
- S(t) = +1
  - Attack success at that moment!
  - This moment must come after block confirmation is assurred.

Possible Decision with MDP

- At each state transition time S(t)
- Attacker can make a decision
  - Continue
  - Stop
    - Success
    - Give up

Definitions

Parameter	Description
p_A	Mary's portion of computing power (0~100%)
p_H	Network's portion of computing power (p_A+p_H=100%)
λ_H	Miners' average block generation speed [blocks/sec]
N_{BC}	Block confirmation number
t_{cut}	Attack cut time for cut loss
T_{AS}	Attack success time (random variable)

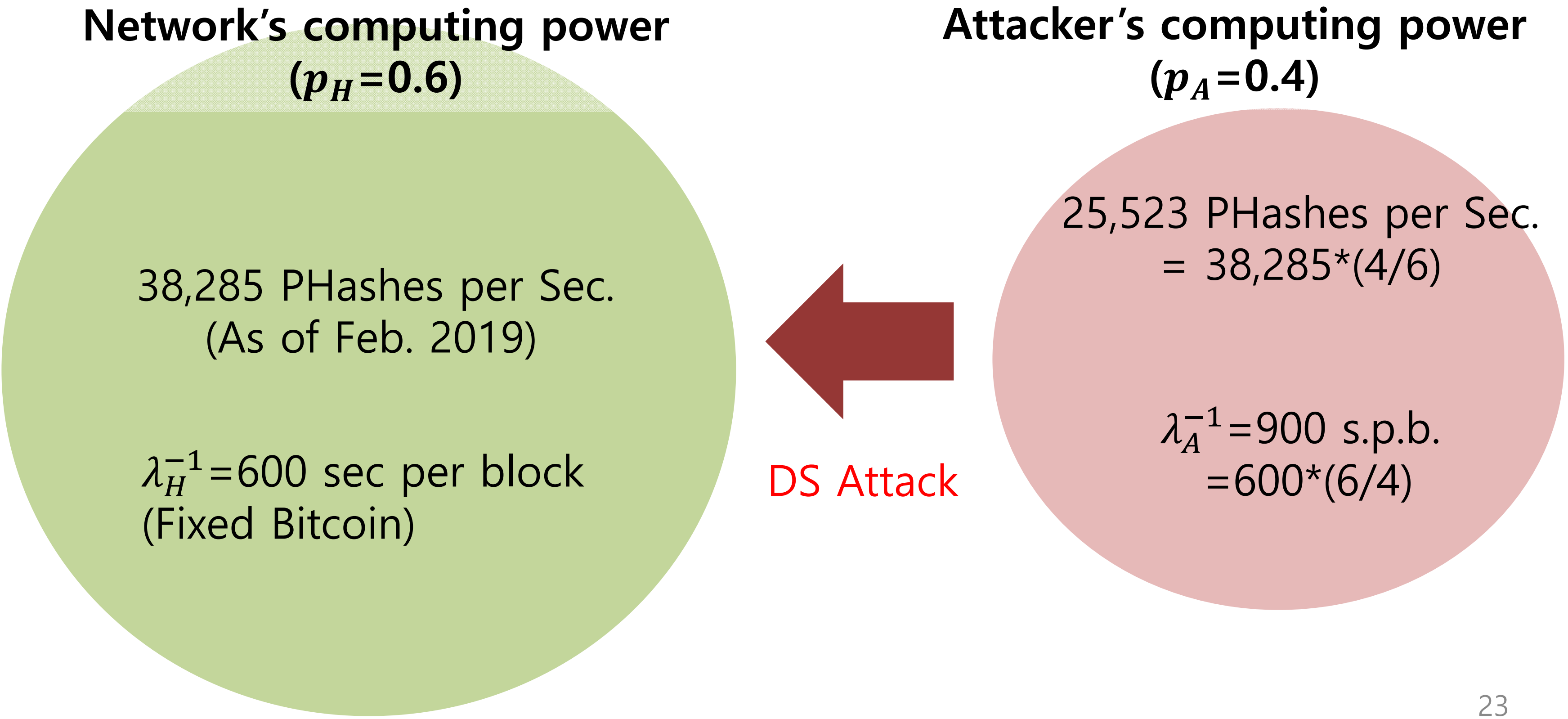
※ Mary's average block generation speed $\lambda_A = \lambda_H\, p_A/(1-p_A)$

Definition. A DS attack succeeds if

- Target transaction has got N_{BC} blocks confirmed ,
- Mary's chain has grown longer than the public chain, and
- the above two conditions have been satisfied within a cut time t_{cut}.

Meaning of Computing Powers

- Network 60% vs. Attacker: 40%
  (Network 100% vs. Attacker 66.6%)



Network's computing power
(p_H=0.6)

38,285 PHashes per Sec.
(As of Feb. 2019)

λ_H^{-1}=600 sec per block
(Fixed Bitcoin)

DS Attack

Attacker's computing power
(p_A=0.4)

25,523 PHashes per Sec.
= 38,285*(4/6)

λ_A^{-1}=900 s.p.b.
=600*(6/4)

Nakamoto’s result

- The probability that double-spending attack will *ever* succeed:

$$\boxed{\mathbb{P}_{AS} = 1 - \sum_{k=0}^{N_{BC}} \frac{\lambda^{k} e^{-\lambda_H}}{k!} \left(1 - \left\{ \begin{array}{cc} 1 & if\ \ p_A \geq 0.5 \\ \left(p_A / (1 - p_A) \right)^{N_{BC} - k} & if\ \ p_A < 0.5 \end{array} \right\} \right).}$$

“**No cut time is set, i.e.** $t_{cut} = \infty$.
If Mary has p_A **greater** than 50%,
then double-spending attack succeeds.”

- According to this result, double-spending attack seems very difficult.
- However, Nakamoto’s result does not say

$$\boxed{p_A < 50\% \quad \overset{?}{\Rightarrow} \quad \begin{array}{l} \text{Profitable DS Attack} \\ \text{is impossible.} \end{array}}$$

Our results (Main)

Definition. A DS attack is *profitable* if and only if the expected revenue is greater than the expected cost.

※ Revenue: cheating value of target transaction
※ Cost: operating expense for computing hash functions

Theorem. For all attacker's fractions of computing power p_A (1%~99%), DS attacks are profitable if the value V of target transaction is greater than

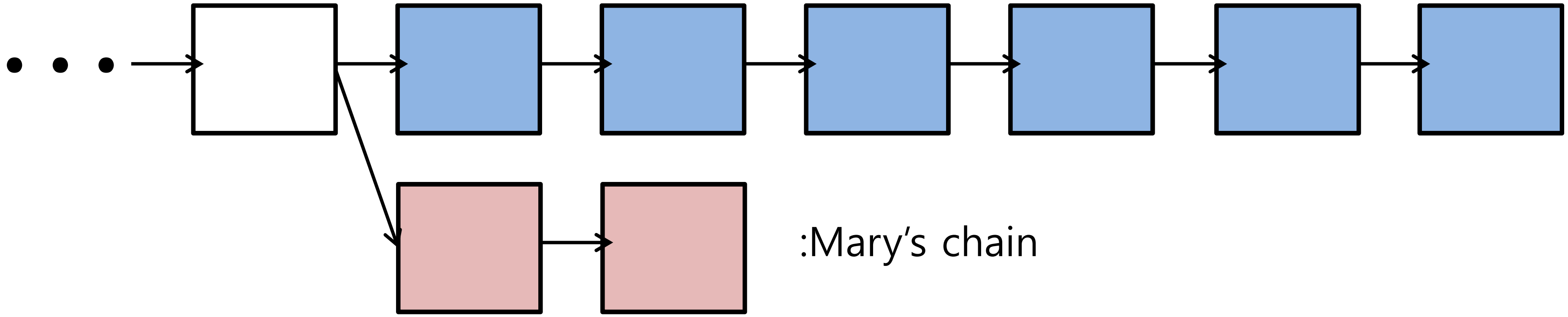
$$V_{Suf.}(p_A; N_{BC}) = \gamma'(p_A) \frac{\lambda_H p_A \mathrm{E}[T_{AS}]}{(1 - p_A)\mathbb{P}_{AS}(p_A, t_{cut}; N_{BC})}.$$

"Even though p_A **is less than** 50% ,
Mary can make a profitable DS attack."

Our results (2)

Theorem. A DS attack using p_A less than 50% are profitable only if a finite cut time $t_{cut} < \infty$ is given.

Example)

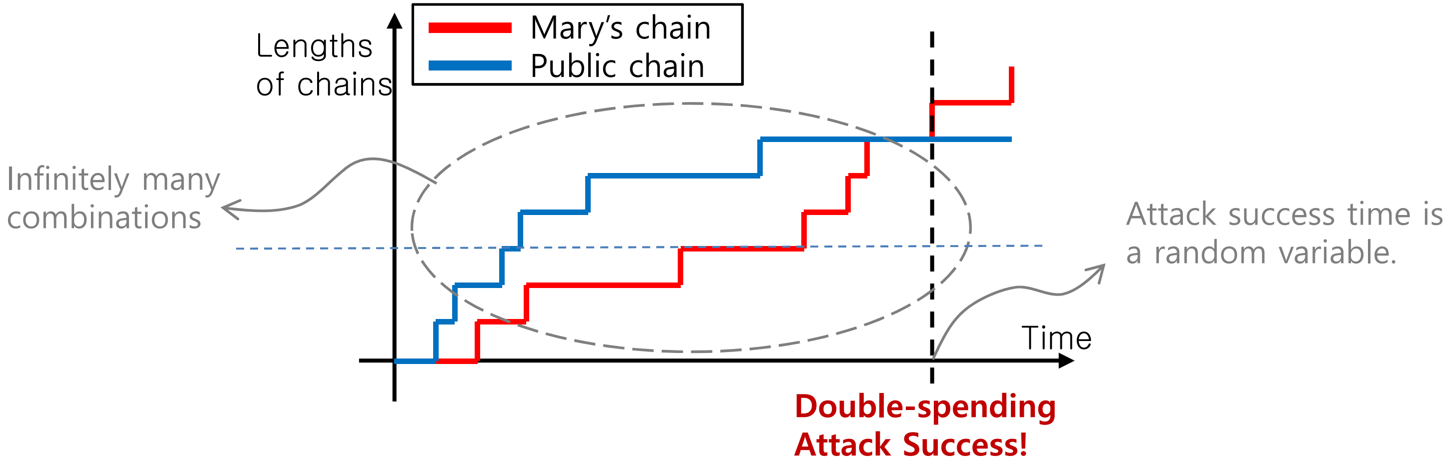


- ➢ The odd that Mary's chain NEVER catch up w/ the network chain is nonzero.
- ➢ If there is no time limit and Mary's chain does not catch up with the public chain, the operating expense is infinity.
- ➢ To avoid this, Mary should give up her attack at a certain **cut time** to cut loss.

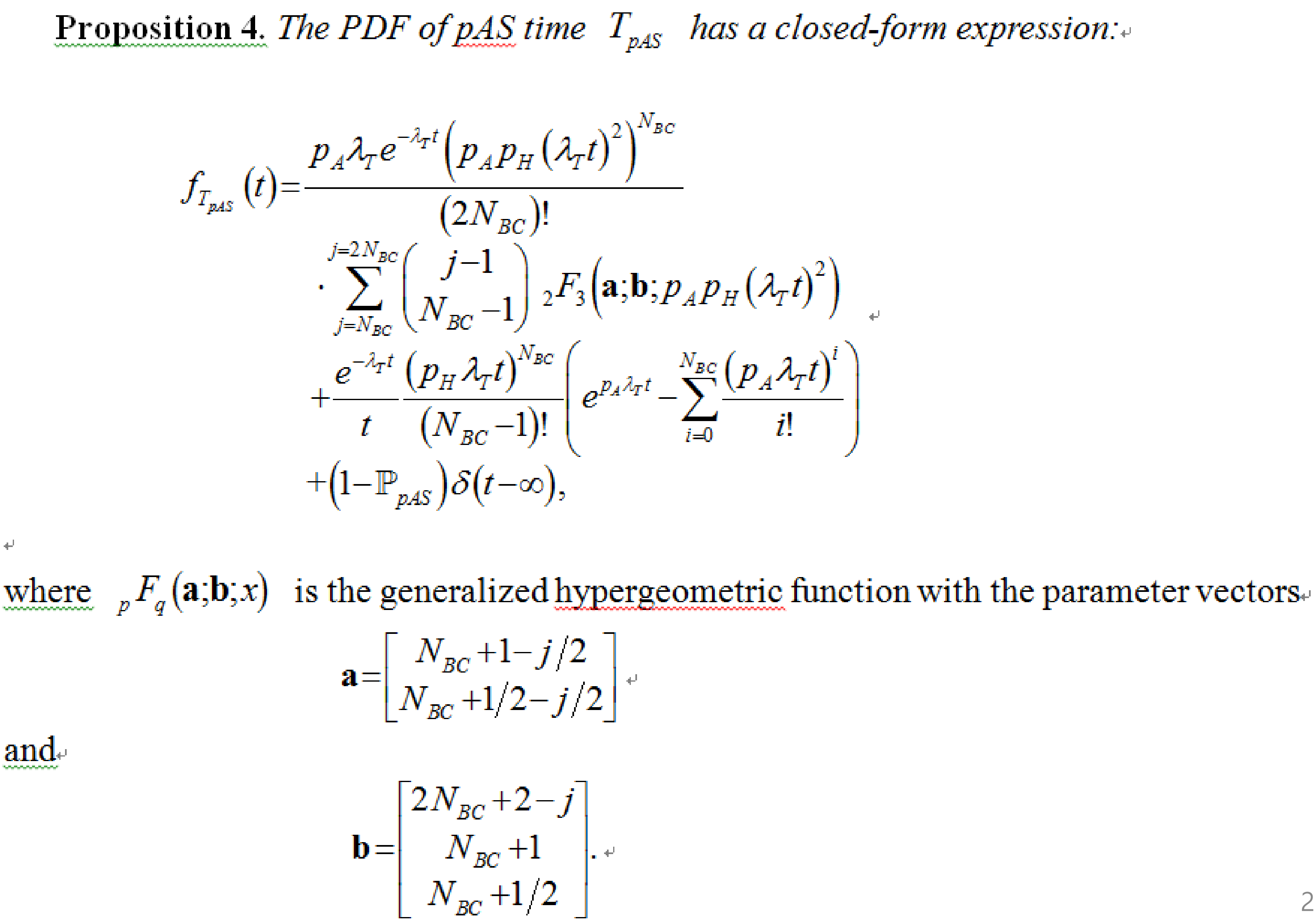
Our results (3)

“We provide the probability density function of attack success time.”

- DS attack is modeled as a competition of two Poisson processes.
- There are infinitely many combinations for the two Poisson processes to compete which give a DS attack success at the end.
- There are infinite possibilities Mary’s chain can catch up with the public chain.
- We came up with a novel way of calculating probabilities, using combinatorics and generating functions.


Lengths of chains
🟥 Mary’s chain
🟦 Public chain
Infinitely many combinations
Attack success time is a random variable.
Time
Double-spending Attack Success!

Our results (3)



Proposition 4. *The PDF of pAS time* T_{pAS} *has a closed-form expression:*

$$f_{T_{pAS}}(t) = \frac{p_A \lambda_T e^{-\lambda_T t}\left(p_A p_H (\lambda_T t)^2\right)^{N_{BC}}}{(2N_{BC})!} \cdot \sum_{j=N_{BC}}^{j=2N_{BC}} \binom{j-1}{N_{BC}-1} {}_2F_3\left(\mathbf{a};\mathbf{b};p_A p_H (\lambda_T t)^2\right) + \frac{e^{-\lambda_T t}}{t} \frac{(p_H \lambda_T t)^{N_{BC}}}{(N_{BC}-1)!}\left(e^{p_A \lambda_T t} - \sum_{i=0}^{N_{BC}} \frac{(p_A \lambda_T t)^i}{i!}\right) + \left(1-\mathbb{P}_{pAS}\right)\delta(t-\infty),$$

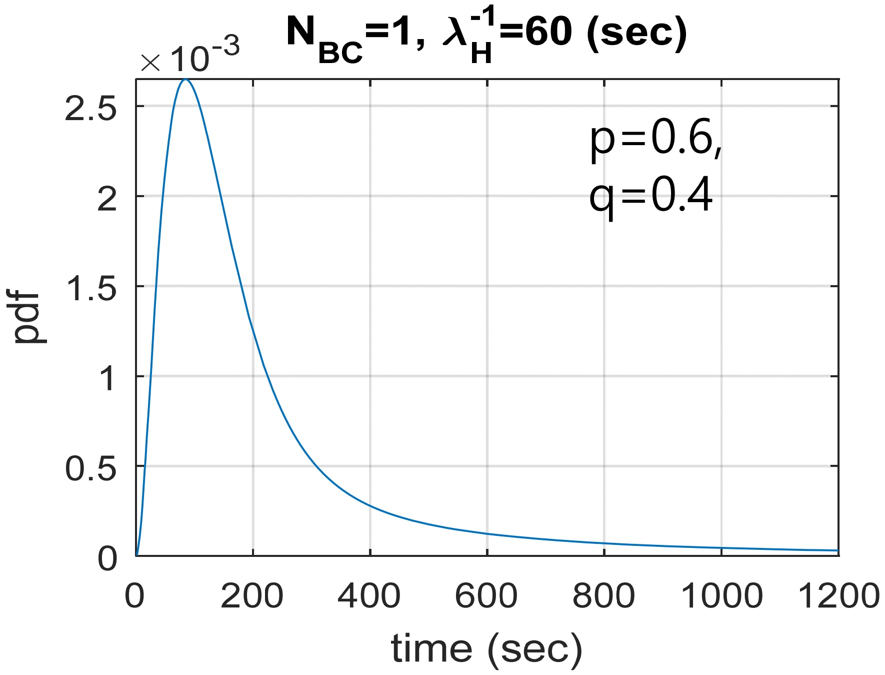
where ${}_pF_q(\mathbf{a};\mathbf{b};x)$ is the generalized hypergeometric function with the parameter vectors

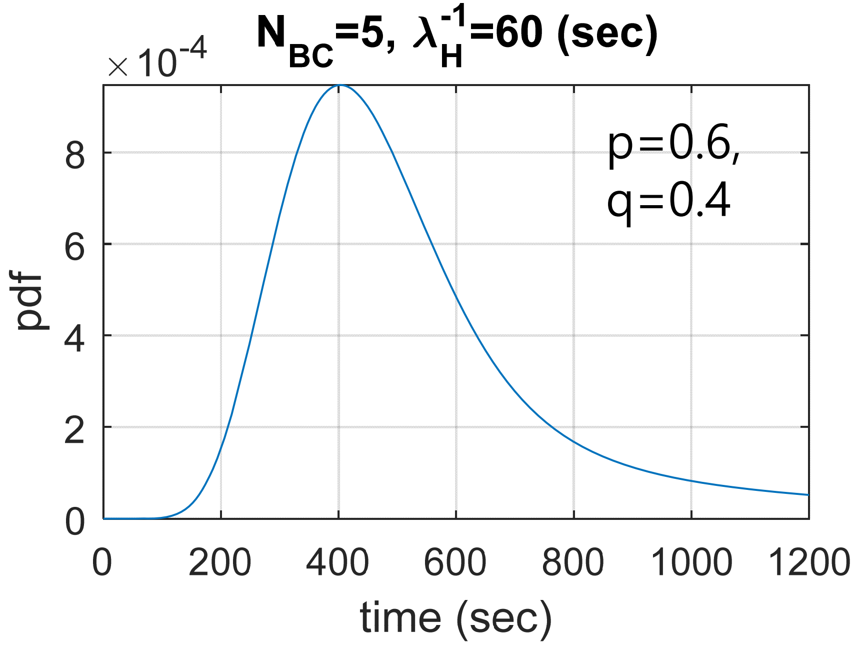
$$\mathbf{a} = \begin{bmatrix} N_{BC}+1-j/2 \\ N_{BC}+1/2-j/2 \end{bmatrix}$$

and

$$\mathbf{b} = \begin{bmatrix} 2N_{BC}+2-j \\ N_{BC}+1 \\ N_{BC}+1/2 \end{bmatrix}.$$

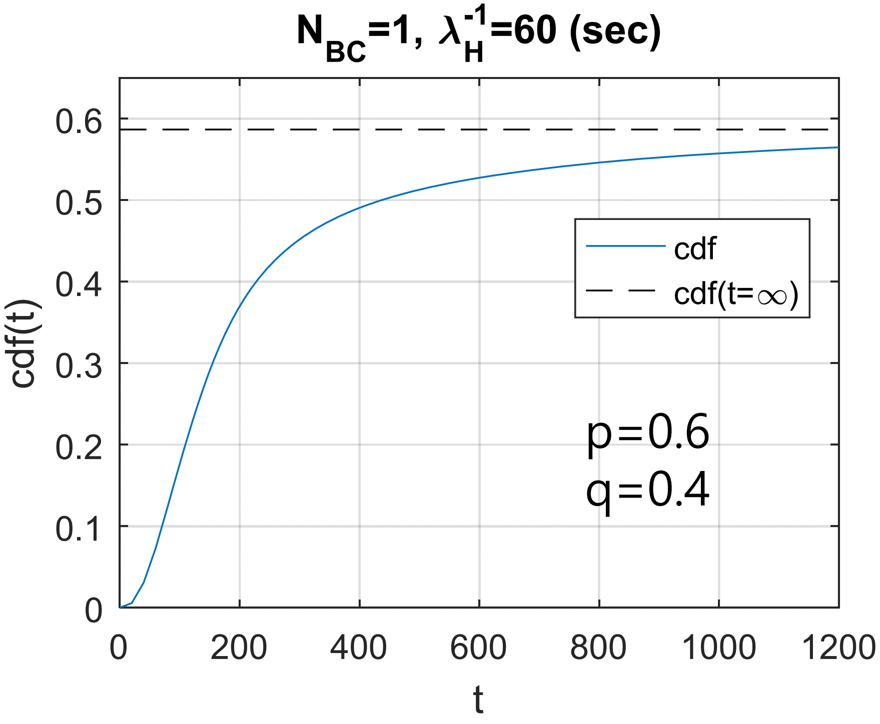
New Result on PDF over time

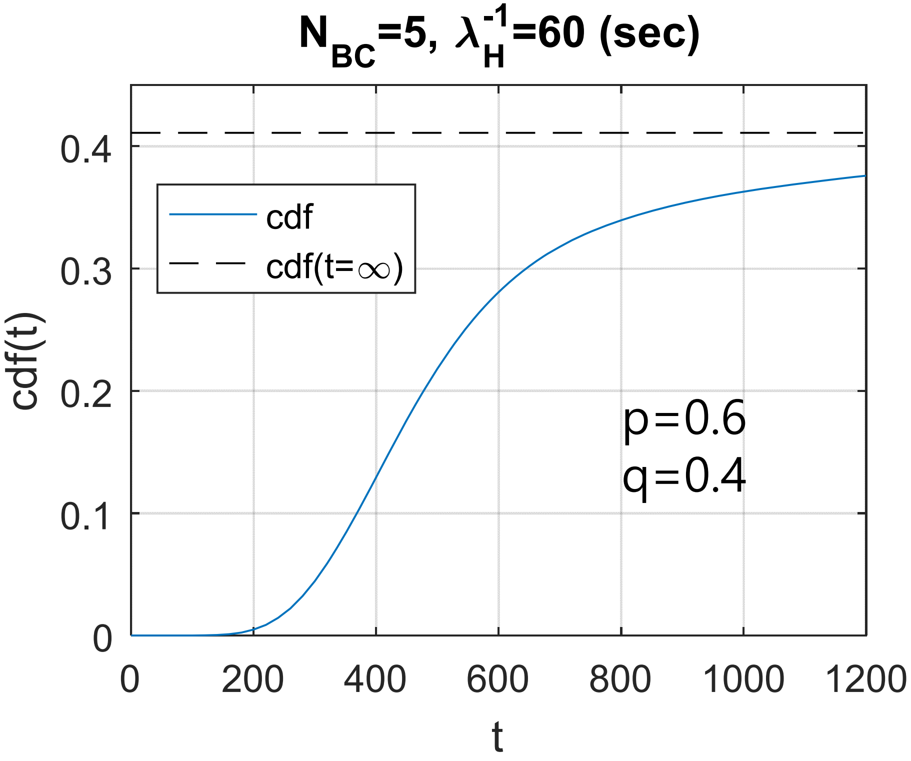




New Result on PDF over time

(Dashed black is Rosenfield Result.)





Ex: *BitcoinCash* network

BitcoinCash Public Info.

- The amount of TXs over 24 hours is about 10 billion dollars.
- Miners' average block generation time λ_H^{-1} is fixed to 600 secs.
- Block confirmation number (N) of target transaction: 5

Mary's Hidden Info.

- Average block generation time: 1143secs (p_A =35%)
- Attack cut time (t_{cut}): 3hours 36mins
- Operating cost per time: γ

Analysis Result

- The value of target transaction: V
- Attack success probability within the cut time: 22%
- Expected attack success time(T_{AS})(if attack succeeded): 1hour 42mins

Profit Info.

- Expected revenue: $0.22 * V$
- Expected cost: 0.22*(1hour 42mins)*γ+0.78*(3hours 36mins)*γ
- Profit=expected revenue-expected cost

Ex with *BitcoinCash* network

Profit Info.

- Expected revenue: $0.22 * V$
- Expected cost: 0.22*(1hour 42mins)*γ+0.78*(3hours 36mins)*γ
- Profit=expected revenue-expected cost

- How to make attack profitable?

"Make a target TX value V i.e. Profit>0."

- The operating expense per time (γ) is given in internet.
- For example, *nicehash.com* provides a rental service of hash power.
- According to nicehash.com, the **expected cost is 2.909 BTC.**

"If $V > 13.225$ BTC, this attack is profitable."

Summary of Contribution

New theorems and propositions are developed such as

1. probabilistic behaviors of attack success time and
2. conditions for profitable DS attacks.

These tools enabled our analyses such as

1. riskiness of DS attacks even with an attack less than 50% of hash power and
2. guidance to prevent profitable DS attacks.

Q&A