Research on Financial Transactions Using Distributed Ledger Technology
Research on Technological Risks in Financial Transactions Using Blockchain

Research Report

March 2018

Financial Services Agency, JAPAN

Information Services International-Dentsu, Ltd.

The views and opinions expressed in this report do not necessarily reflect the official policy or position of JFSA.

The contents other than past or present facts described in this report are based on the information available at the time of writing. Therefore, it must be noted that actual trends may fluctuate due to various factors.

Table of Contents

【Figures, Charts and Tables】

1. Introduction

1.1. Object of Research

Of the information technologies used in FinTech, the blockchain technology is expected to bring about major changes in the market, substantially reducing costs and improving usability for users.

In this context, the Financial Services Agency announced its financial administration policy for FY2017 in November 2017. At that time the Agency indicated its policy to promote the international cooperative studies on the blockchain technology in order to enhance coordination with leading figures and competent authorities overseas in the field, while taking into account the trend toward international standardization of the technology. This research is carried out as part of the international cooperative studies.

The blockchain is an element technology used, most notably, in the cryptocurrency trading. Although studies and substantiative experiments are under way for application of the technology in various financial transactions, assessment of technological risks and vulnerabilities is not necessarily in good progress. With the transactions in cryptocurrencies increasing more and more recently, including the bitcoin transactions, the incentive is getting high for attackers to carry out malicious attacks, exposing these transactions to technological risks. In Japan, too, there was actually an incident where cryptocurrencies were stolen due to the vulnerability of the systems of the exchange.

In Japan, the Payment Services Act was recently revised, which placed the regulatory framework for cryptocurrency operators in place, including introduction of the registration system. However, in order to promote the state-of-the-art trial experiments and programs in the field of the blockchain technology, we believe that it is important to identify issues and study countermeasures throughout the business ecosystem, particularly that of the cryptocurrency transactions implemented with the use of the public blockchain, including not only the risks associated with the platform of the cryptocurrency operators but also the risks and vulnerabilities of the blockchain technology that forms the technological basis of the business.

Accordingly, in this study, we looked, from the viewpoint of the user protection, at the business ecosystem of the cryptocurrency trading as a whole, with focus on the security of the blockchain technology, to identify what are the technological issues and vulnerabilities and what measures should be taken in the future.

1.2. Current Policy on Cryptocurrencies

Information technologies not seen before are utilized in the cryptocurrencies, including the blockchain technology. Cryptocurrency operators are also required, under the revised Payment Services Act, to equip themselves with highly sophisticated business management structure with focus on computer systems for the protection of the users. At the same time, various movements have been seen in the market, including wild fluctuations in the cryptocurrency values since the beginning of the year 2017 and the split of cryptocurrencies. As such, it is important to identify how the environment surrounding the cryptocurrencies affects the users and financial systems.

Under the revised Payment Services Act, the Financial Services Agency has been watching the trend in the cryptocurrency markets closely and monitoring if the cryptocurrency operators have an adequate business operation structure in place, paying attention to the balance between promotion of innovations and protection of users.

Specifically, the Agency has been checking if the operators have an adequate operational structure for protection of the users, providing adequate explanations and information in response to changes in the environment surrounding the cryptocurrencies. It has also been monitoring if they have an adequate structure in place that appropriately identifies the risks and manages system risks properly, and if they have studied and implemented effective measures for prevention of fraudulent activities such as money laundering, because secure and stable system management and fraud prevention are critical for the operators to gain confidence of the users.

In addition, recently, the number of initial coin offerings (ICOs) has been increasing, which are essentially financial arrangements using cryptocurrencies. Some of the tokens issued at the time of ICOs are considered meeting the definition of cryptocurrency under the Payment Services Act, and it is important to get sufficient information on these activities. In the meantime, with regard to fraudulent ICOs, the Agency will protect the users by cracking down such activities in coordination with other competent authorities while promoting the industry to establish self-regulatory rules and alerting the users and operators about the risks associated with ICOs.

1.3. Objectives and Assumptions

1.3.1. Study Objectives

In the financial transactions using the blockchain, there are various risks involved, from risks associated with technological factors to those related to the community governance including the split of the cryptocurrencies. The type of attacks on the financial transactions using the blockchain differs depending on the type of the blockchain (permissioned/non-permissioned), type of risk and method of transaction finalization.

In this study, in consideration of these differences, we will review existing methods of attacks on the financial transactions that use the blockchain, focusing on factors such as if there are means to avert the attacks and on events that undermine the requirements for sustainability as the basis for financial transactions in the future. We then will carry out the substantiative experiments as necessary. We will also examine technological and institutional countermeasures through these experiments.

1.3.2. Assumptions

1.3.2.1. Forms of Blockchains to Be Studied

There are broadly two types of blockchains:

(1) Public blockchain (non-permissioned): Many and unspecified participants (bitcoin, etc.)

(2) Private blockchain (permissioned): Specified participants only

In consideration of the difference, we will carry out the desktop experiment on (1) above. As a result of the risk assessment, if the risk is high at present and it is identified that there exists vulnerability that may become a major issue, we will carry out the substantiative experiment to see if there are countermeasures and to what extent the vulnerability may affect the blockchain.

1.3.2.2. Type of Risks to Be Studied

In this study, the risk analysis will be performed on the risks attributable to technological factors of the blockchain.

For the attacks that arise from the risks inherent to the blockchain, we will analyze objects of the attack (cryptography, protocol, system, the Internet, etc.) and impacts on the users.

1.3.2.3. Selection of Transactions on Blockchain under Study

There are various mechanisms and use cases of public blockchains that are currently proposed depending on the consensus algorithms (a method to finalize the ledger data on the blockchains stored on distributed nodes) and executable statements (scripts). The blockchains to be examined in this study are those used for bitcoin, for which there are a large number of preceding studies and which has the longest history, and we will look at transactions executed on these blockchains.

[Definition of Terms (1)]

- Blockchain

A blockchain is a distributed database in the distributed ledger technology and is the foundation of cryptocurrencies such as bitcoin and ethereum. The blockchain is a chain-like connection of data units called block that stores the data.

There is so far no established definition of the blockchain, and international standardization organizations are working for standardization of the definition. Although there is no established definition, the Japan Blockchain Association defines the blockchain as follows:

"In a broader sense, a blockchain is a technology with a data structure which can easily detect manipulation using digital signatures and hash pointers, and where the data has high availability and integrity due to distribution across multiple nodes on a network."

The blockchain technology is expected to be used in various areas such as healthcare and supply chain, not limited to cryptocurrencies and financial transactions. In this report, we will explain the mechanism of the blockchain taking bitcoin as an example, which is currently the most popular virtual currency that is dependent on the blockchain technology.

In bitcoin, the currency is represented with "transactions." Each transaction consists of a plurality of inputs (payers) and a plurality of outputs (receivers). All the inputs must be signed with the secret key of the payer. The input has just the transaction identifier (hash) that identifies the transaction causing the transfer of the payer coin, and an index pointing to the output within the transaction. It does not have the address of the payer or the number of coins. The output is generated by inputting the public key, which corresponds to the secret key owned by the payer, to the hash function.

Transactions are combined into a single block about every 10 minutes by the miner node (or simply miner), which will be explained later. When the block is then connected to the previous block and set in the blockchain, the transaction is approved. The act of the miner connecting the block to the previous one is called mining, and the difference between the total number of coins in the inputs and that in the outputs of the transactions included in the block is the fee receivable by the miner when connecting the blocks.

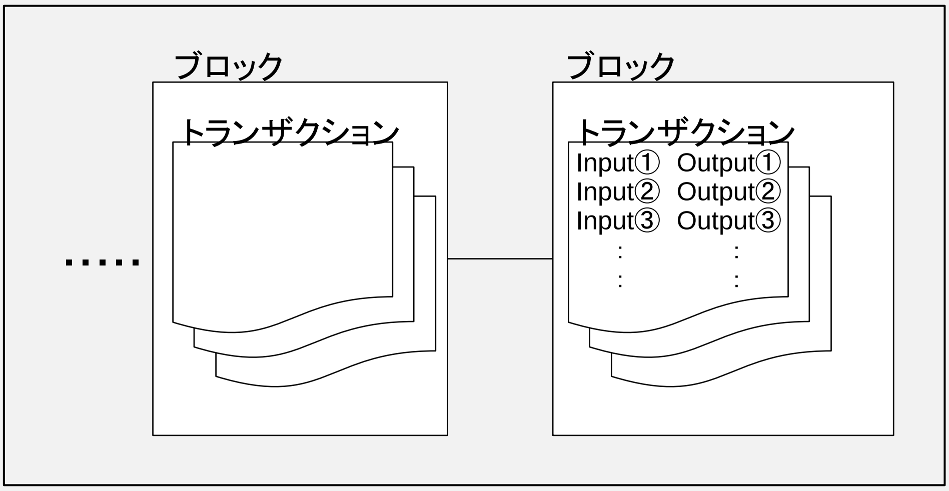


Figure 1-1:Structure of Block and Transaction of Bitcoin

[図要素]
1. Block
Transaction

2. Block
Transaction

[Definition of Terms (2)]

- Bitcoin Network

The bitcoin network is a P2P network, and each computer participating in the bitcoin network is called node. A node is an aggregation of the following four functions: 1) routing; 2) full blockchain database; 3) mining and 4) wallet. A node having all of the four functions is the full node. All the nodes must have the routing function in order to participate in the bitcoin network, but whether it has other functions depends on the role of the node.

1) Routing
Routing is a function for managing connection to other nodes, sending, receiving and verifying the data.

2) Full blockchain database
All the transactions approved as at the present moment are recorded.

3) Mining
It is a function to mine the blocks.

4) Wallet
It is a function to manage the bitcoin address used for receipt/payment of coins.

A node called full node has the full blockchain database, and manages the up-to-date complete blockchain. The full node autonomously verifies the transaction to be described later without reference to the data outside the node. In contrast, there are nodes that do not have the full blockchain database and manage just a part of the blockchain, verifying the transactions with a simplified method called SPV (simplified payment verification). These nodes are called SPV node or lightweight node.

A miner node that has the mining function competes for generation of new blocks, and has a special hardware running for solving the Proof of Work algorithm to be described later. Some miner nodes are also full nodes, and manage the complete copy of the blockchain. Others are lightweight nodes participating in the mining pool to be described later. They depend on the pool server that manages the full nodes.

Some of the user wallet nodes that have the wallet function participate as full node in the form of the desktop bitcoin client. Many user wallets on the devices with limited resources, such as smartphones, are SPV nodes.

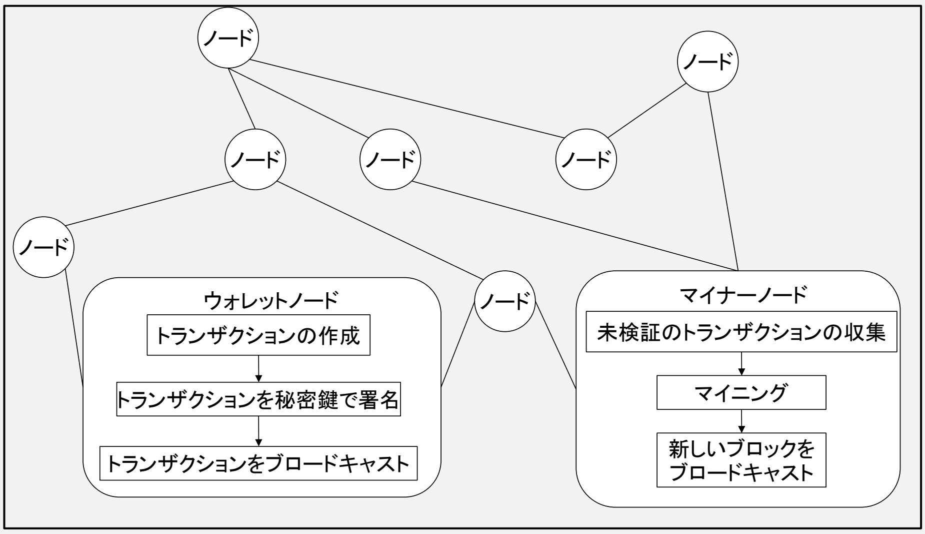


Figure 1-2: Bitcoin Network and Node

[図要素]
1〜7 Node
8 Wallet Node
Creation of transaction
Signing transaction with secret key
Broadcasting transaction
9 Miner Node
Collection of unverified transactions
Mining
Broadcasting new block

[Definition of Terms (3)]

■ Transaction and Consensus Formation in Bitcoin

The process of remitting *x* bitcoins from Alice to Bob on the bitcoin network is as follows.

1) Creation of Remittance Transaction

Select unused outputs in the transactions Alice received in the past so that the total amount will be *x* bitcoins or more, and create a transaction on Alice's wallet that has as input the selected outputs, and as output the address created from the public key corresponding to Bob's secret key and the payment amount (*x* bitcoins). If the total amount of the inputs is *x'*, fees payable to the miner is *x_f* and *x' – x* is larger than *x_f*, Alice adds to the transaction an output consisting of the address created from the public key corresponding to Alice's secret key and the amount *x' – x – x_f* bitcoins, so that Alice can receive *x' – x – x_f* as change.

2) Sending Transaction

When Alice sends the transaction to the adjoining nodes on the network, the transaction is diffused from node to node on the bitcoin network in sequence. In order to reduce the bandwidth load on the bitcoin network, the process is designed so that the node first sends the hash value of the data to the receiver node as inv message prior to sending the body of the data. If the receiving node really requires the body of the data, it sends the getdata message, requesting that the body of the data be transmitted.

3) Receiving Transaction

The miner node, upon receipt of the transaction, stores it temporarily in the memory pool built on the computer. At the same time, it carries out the calculation for the mining as described above. The mining process is as follows. First, the miner selects a transaction from the memory pool that is to be stored in the new block. It usually selects a transaction that has a high fee. It also stores the hash of the immediately preceding block and an arbitrary string of characters called nonce to the block. Next, it repeats calculating the hash value of the entire block, changing one character of the nonce every time. If the hash value is smaller than the value of "difficulty" (difficulty of mining) that was set in advance, the nonce that led to the hash value is the right nonce. The miner that found the right nonce is the winner of the mining competition, and the block of the winning miner is accepted as the latest block to be connected to the blockchain. The other miners are losers, and check if the hash value calculated with the right nonce is indeed smaller than the difficulty, or if the block determined by the winner as having no issues is indeed error-free. When the majority of the miner nodes that participated in the mining approved the nonce, the block is officially connected to the blockchain. The winning node receives the reward for the mining called coinbase, and the fee of the transactions included in the block.

4) Fork

During the process of connecting the block to the blockchain, the blockchain may occasionally branch. This is called fork. For example, if two (or more) miners simultaneously succeed in mining and simultaneously transmit the new blocks to the bitcoin network, the blockchain branches. At this moment, both are legitimate blocks. When a block created later is connected to one of the legitimate blocks, the one with the new connection remains legitimate but the other is rendered illegitimate. When there is a branch, the longest blockchain is considered "legitimate blockchain." The shorter one is considered "orphan block," and is discarded. This means that the transactions in the orphan block are considered not closed. If the transactions in the orphan block are legitimate, they will be included in the longest, "legitimate blockchain" again.

The consensus formation algorithm above is called Proof of Work.

The volume of calculation work in mining is called hash power, and a high hash power is required for successful mining. For this reason, many miners presently belong to a mining pool, a service to which a plurality of miners come together to perform mining in cooperation.

Works cited: Antonopoulos, Andreas M. Mastering Bitcoin: Unlocking Digital Cryptocurrencies, translated into Japanese by Takaya Imai and Junichiro Hatogai

2. Method of Study

2.1. Analysis of Preceding Research Papers

2.1.1. Selection of Research Papers to Be Reviewed

We will first select the research papers on the general attack methods and countermeasures on the blockchains to be examined in this study. We decided to select these from the survey papers on the attack methods and response technologies reported mainly in the international conferences listed below. (For the list of research papers examined in this survey, please refer to the Appendix "List of Research Papers Examined.")

- Financial Cryptography and Derived Workshops
- IACR Cryptology ePrint Archive

The attack methods described in the selected research papers will be considered as "risk," which will then be assessed.

2.1.2. Establishment of Risk Assessment Approach

There are many approaches to the risk assessment, but there are generally two main approaches: model-based approach and index-based approach. In the model-based approach, a theoretical model is literally used to measure the risks mathematically. In the index-based approach, several indices related to the risks are used for assessment.

It is not the object of this study to develop a quantitative model for the risk of attacks on the financial transactions using the blockchains for sophisticated risk assessment. In the future, it certainly is desired to develop a sophisticated risk assessment method using a model, etc. but in the assessment in this study, we decided to focus on intuitive identification of the risk level and breakdown of the risks into patterns.

For this reason, although we intend to use the basics of the model-based approach such as the risk event probability and influence rate, we will employ a simple scoring method based on the qualitative assessment.

1) Risk Scoring Method
From the two aspects of maintaining the "protection of users" and "soundness of financial transaction systems," we will perform the assessment with focus on the "influence rate (size of damages)" and "event probability," in case the risks materialize.

2) Existence or Non-existence of Countermeasures to Risks
We will perform the assessment on existence/non-existence of countermeasures to the risks, such as aversion, alleviation, transfer and tolerance, and on how these countermeasures, if any, are used in reality.

2.1.3. Risk Assessment Axes

2.1.3.1. Risk Scoring Method

Each risk is assessed with the following assessment axes.
In the risk assessment in this study, the influence rate (size of damages) of the risk of attacks on the existing financial transaction systems will be measured separately for the “users” and “financial transaction systems.”

- Rate of Influence to Users (Size of Damages)
  - Scope of Impact (Large, Medium and Small)
  - Degree of Seriousness (Size of Damages in Case the Risk Materializes) (High, Medium and Low)
- Rate of Influence to Financial Transaction Systems (Size of Damages)
  - Scope of Impact (Large, Medium and Small)
  - Degree of Seriousness (Size of Damages in Case the Risk Materializes) (High, Medium and Low)

In this study, the users and financial transaction systems are defined as follows:

- Users
  - Seller: An entity receiving the cryptocurrencies.
  - Buyer: An entity paying the cryptocurrencies.
- Financial Transaction Systems
  - Miner: An entity contributing to creation of blockchains.
    It digs blocks at nodes and receives cryptocurrencies as reward.
  - Exchange: An entity exchanging cryptocurrencies with other (crypto-) currencies.

In this context, miners may be construed as users of the financial transaction systems because they receive the cryptocurrencies as reward for mining. However, because miners are indispensable entities in running the blockchain, they are defined as part of the financial transaction systems in this study.

- Event Probability
  - Ease of Attacks (High, Medium and Low)
    It is considered that if there are no barriers against the attack (the cost of the attack), the ease of attack is high, and that if there are barriers, the ease of attack is low.
  - Incentive for Attackers (High, Medium and Low)
    It is considered that if the return to the attack is high, the incentive is high for the attackers, and that if the return is low, the incentive is low.

In an ideal risk assessment, it is necessary to define quantitative measures for description of various impacts and degree of seriousness. In the risk assessment of this study, however, as described above, it is not the study object to quantitatively identify various risks in the blockchain: instead, we decided to focus on intuitive identification of the risk level and breakdown of the risks into patterns.

The assessment criteria for the assessment axes are defined as shown on Table 2-1.

Table 2-1: Assessment Criteria for Assessment Axes

Assessment Axes				Assessment Criteria
Influence Rate (Size of Damages)	Rate of Influence to Users	Scope of Impact	Large	Most users are attacked (wide scope).
			Medium	Part of the users are attacked (medium scope).
			Small	Most users are not attacked (limited scope).
		Degree of Seriousness (Size of Damages)	High	Most of the values of the user assets are lost (huge).
			Medium	Part of the values of the user assets are lost (medium).
			Low	Most of the values of the user assets are not lost (minor).
	Rate of Influence to Financial Transaction Systems	Scope of Impact	Large	Most miners and exchanges are attacked, with possible systemic risk (wide scope).
			Medium	Some of the miners and exchanges are attacked (but rather limited in scope) (medium).
			Small	Most miners and exchanges are not attacked (limited).
		Degree of Seriousness (Size of Damages)	High	Most financial transaction systems are invalidated (huge).
			Medium	Some of the financial transaction systems are invalidated (medium).
			Low	Most financial transaction systems are not invalidated (limited).
Event Probability		Ease of Attacks	High	There are very few barriers against the attacks.
			Medium	There are some barriers against the attacks.
			Low	There are strong barriers against the attacks.
		Incentive for Attackers	High	Most attackers have the incentive.
			Medium	Some of the attackers have the incentive.
			Low	Few attackers have the incentive.

Next, we assign scores to the assessment criteria of the assessment axes, with higher risks having higher points: 3 points for the high risks; 2 points to the medium risks; and 1 point to the low risks. Next, the highest of these scores is chosen as the score for an assessment axis: the score of the assessment axis “Rate of Influence to Users” will be 3 if the “Scope of Impact” has the score 3 and “Degree of Seriousness” has the score 1.

From these scores, we will derive the two types of risk: “Rate of Influence to Users” multiplied by “Event Probability;” and “Rate of Influence to Financial Transaction Systems” multiplied by “Event Probability.”

Table 2-2: Risk Level (Rate of Influence to Users multiplied by Event Probability)

Risk Level		Rate of Influence to Users		
		Large	Medium	Small
Event Probability	Large	9 (High)	6 (Medium)	3 (Low)
	Medium	6 (Medium)	4 (Medium)	2 (Low)
	Small	3 (Low)	2 (Low)	1 (Low)

Table 2-3: Risk Level (Rate of Influence to Financial Transaction Systems multiplied by Event Probability)

Risk Level		Rate of Influence to Financial Transaction Systems		
		Large	Medium	Small
Event Probability	Large	9 (High)	6 (Medium)	3 (Low)
	Medium	6 (Medium)	4 (Medium)	2 (Low)
	Small	3 (Low)	2(Low)	1 (Low)

2.1.3.2. Countermeasures

For each risk, the countermeasures against the risk are classified into the following three categories:

1) There are implemented and/or operational countermeasures.
2) There are countermeasures in theory (implementation yet to be developed for the systems).
3) There are no countermeasures.

As the risks to be analyzed here are those attributable to technological factors as stated in 1.3.2.2 “Type of Risks to Be Studied” above, the measures to be described and classified below are also limited to technological measures. Systematic measures including regulations by competent authorities will be reviewed and described later in “5. Discussion.”

3. Results of Risk Assessment

We performed the assessment based on the assessment criteria described above.

As preparation for the assessment, we first summarize the preceding research papers on attacks on the blockchains and prepare the list of conceivable risks.
We then clarify the details of damages and scopes of impact in case one of the risks materializes, as well as measures necessary for aversion or alleviation of the risk.

For each of the risk factors, we prepared the following information items. (For details of specific attack risks, refer to “6. Reference Materials.”)

- Summary of attack scenarios and protocols
- Summary of response measures
- Reference information
  - Details of attack
  - Impact
  - Major attackees
- Research paper describing the attack
- Author(s) and title
- Publication media
- Type of publication media

3.1. Results of Risk Scoring

For each of the risk factors, we performed the risk assessment in accordance with “2.1.3. Risk Assessment Axes,” the results of which are shown below on Table 3-1.
Refer to 0 for details of the risk factors.

Table 3-1: Risk Assessment

#	Risk Factor	Risk Level (Rate of Influence to Users)	Risk Level (Rate of Influence to Financial Transaction Systems)
1	Double spending or Race attack	3 (Low)	1 (Low)
2	Finney attack	3 (Low)	1 (Low)
3	Brute force attack	6 (Medium)	4 (Medium)
4	Vector 76 or one-confirmation attack	2 (Low)	2 (Low)
5	>50% hashpower or Goldfinger (Majority attack)	3 (Low)	3 (Low)
6	Block discarding or Selfish mining	2 (Low)	4 (Medium)
7	Block withholding	2 (Low)	4 (Medium)
8	Bribery attacks	4 (Medium)	4 (Medium)
9	Refund attacks	6 (Medium)	2 (Low)
10	Punitive and Feather forking	3 (Low)	1 (Low)
11	Wallet theft	9 (High)	6 (Medium)
12	Transaction malleability	6 (Medium)	3 (Low)
13	Time jacking	3 (Low)	6 (Medium)
14	Sybil	6 (Medium)	6 (Medium)
15	DDoS	6 (Medium)	9 (High)
16	Eclipse or netsplit	4 (Medium)	4 (Medium)
17	Tampering	4 (Medium)	4 (Medium)

18	Deanonymization	3 (Low)	1 (Low)
19	Compromise of underlying cryptographic algorithms	9 (High)	9 (High)

Chart 3-1 below has the risk level (rate of influence to users) as x axis and risk level (rate of influence to financial transaction systems) as y axis based on the risk levels on Table 3-1.

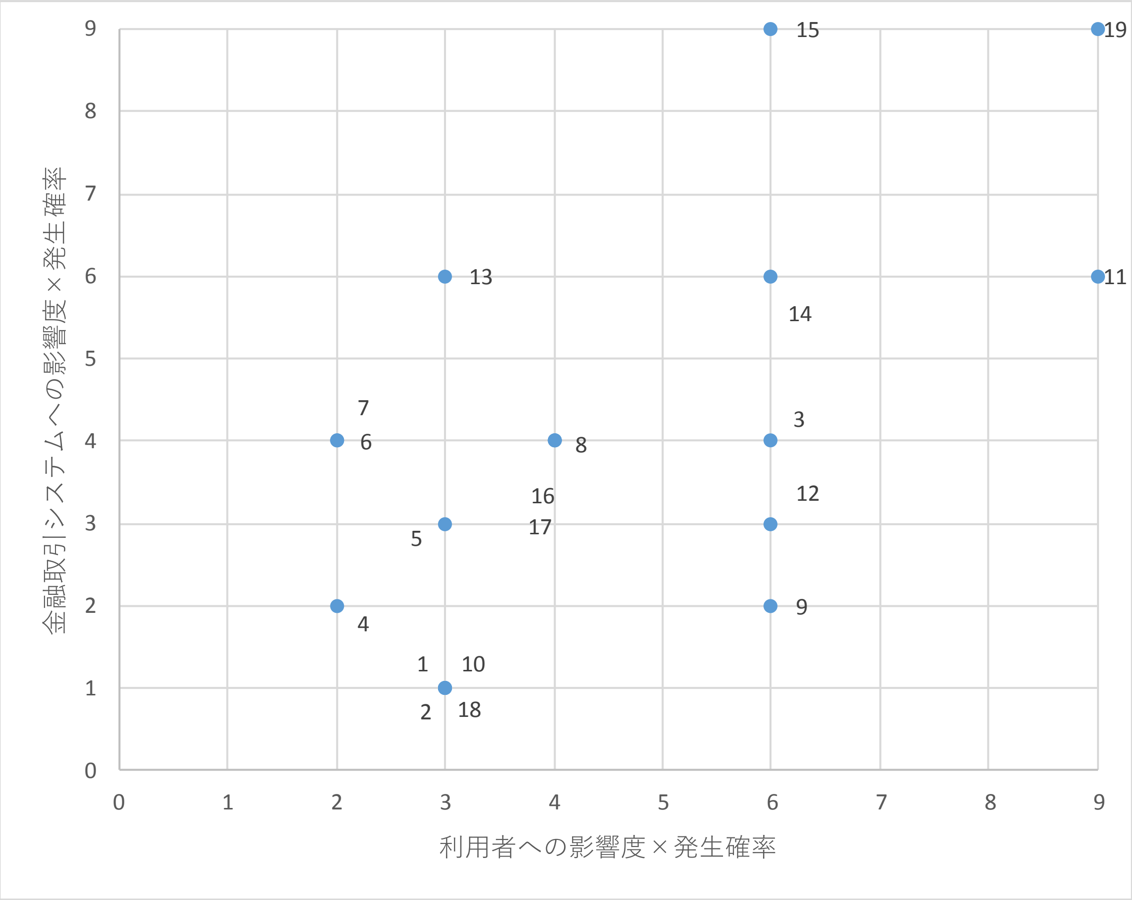


Chart 3-1: Attack Risk Assessment

[図要素]
1. Rate of Influence to Financial Transaction Systems multiplied by Event Probability
2. Rate of Influence to Users multiplied by Event Probability

From Chart 3-1, it is known that the attacks that seriously affect both the users and the financial transaction systems are “DDoS (15)” and “Compromise of Cryptographic Algorithm (19).”

3.2. Countermeasures

Further, for each of the risk factors, we classified the countermeasures, which are shown on Table 3-2.

Table 3-2: Classification of Measures against Risks

Measuers against Risk	Risk Factors
A) There are implemented and/or operational countermeasures.	1. Double spending or Race attack 2. Finney attack 3. Brute force attack 4. Vector 76 or one-confirmation attack 11. Wallet theft 12. Transaction malleability 16. Eclipse or netsplit 18. Deanonymization
B) There are countermeasures in theory.	6. Block discarding or Selfish mining 7. Block withholding 9. Refund attacks 13. Time jacking 15. DDoS 17. Tampering 19. Compromise of underlying cryptographic algorithms
C) There are no countermeasures.	5. >50% hashpower or Goldfinger attack 8. Bribery attacks 10. Punitive and Feather forking 14. Sybil

3.3. Overall Assessment

In the cryptocurrency transactions, it has been known that there are various risks involved. Of the 19 risks extracted in this study, we, using the risk assessment method described above, were more or less able to classify those that are high in terms of attacks (or vulnerabilities) and those that are not very high but countermeasures against which do not exist. In this section, we will outline the attack methods of the risks classified as high risk or as having no countermeasures, assess these risks in consideration of the rate of influence to the users, and present the overall assessment that also has the arguments on appropriate treatment of the cryptographic technologies in the information systems in general taken into consideration.

3.3.1. Attacks and Vulnerabilities Assessed as High Risk

In the risk assessment of this study, we performed assessment of risk levels for two types of influence rates: influence on the users and influence on the financial transaction systems. Looking at the results, we find that “11. Wallet Theft” has the high risk level in terms of the rate of influence to the users on a standalone basis, and that “15. DDoS” has the high risk level in terms of the rate of influence to the financial transaction systems on a standalone basis. In addition, “19. Compromise of Cryptographic Algorithm” has high risk levels in terms of both of the two types of influence rates. In particular, although there are theoretical countermeasures against the Wallet Theft, such measures have not been implemented properly, and this attack has actually been used in theft of cryptocurrencies. With regard to the Compromise of Cryptographic Algorithm, the risk is high that the values of the cryptocurrencies are damaged, and the rate of influence is considered high if the risk materializes. In the following sections, we will describe outlines of the attacks.

3.3.1.1. Overview of Wallet Theft

This is an attack in which the attacker breaks into the wallet of the user for fraudulently obtaining the cryptocurrencies. The attackee is the user of the cryptocurrencies. There are various types of wallets, but particularly in the case of the web wallet, the service provider such as cryptocurrency exchange operator usually manages the secret key, and the user enters the user ID and password for logging onto the service to access the wallet. This means that the attacker may be, by means of hacking or viruses, able to obtain the user ID and password, or even the secret key.

3.3.1.2. Overview of DDoS Attack

In this attack, the attacker transmits a huge quantity of fraudulent data such as bogus blocks and transactions to the network from many client machines, thereby exhausting the resources on the network. This prevents the users from accessing the network, allowing the attacker to let the miners discard the blocks transmitted from the rightful users.

The attack exploits the vulnerability of the bitcoin network that anyone can transmit fraudulent data at a low cost.

3.3.1.3. Overview of Compromise of Cryptographic Algorithm

In the blockchain architecture, it appears that the transition of the cryptographic algorithm is not considered. When the cryptographic algorithm in use in the blockchain is compromised and an appropriate countermeasure is not taken, the entire values of the financial transactions based on the blockchain can be lost.

There are two types of compromise of the cryptographic algorithm. One is use of brutal force in decryption, thanks to the development of the computer power. The other is when an efficient method of decryption is found for the cryptographic algorithm in use[1]. In the modern history of cryptography, the former was observed every couple of decades, whereas the latter may take place at any moment.

3.3.2. Attacks with Theoretical Countermeasures Only (Implementation Yet to Be Developed)

Many of the attacks, against which there are theoretical countermeasures only, are those that discard or delay the blocks or those that exhaust the calculation resources of the mining pool. These attacks exploit the vulnerabilities of the bitcoin architecture, and at the time of this study, no countermeasures have been developed. However, none of these attacks are reported to have damaged the assets of the users so far.

3.3.2.1. Outline of Block Discarding or Selfish Mining

This is an attack designed to have the attacker get more rewards by wasting the calculation resources of general miners and mining pools, thereby making the results of the attacker’s block verification more likely to be chosen. The attacker first secures blocks (*1) that were already mined but that were not released to general miners, and releases them (*1) immediately after a general miner mines the block, thereby increasing the chance that the attacker’s blocks are chosen. A proposed practical countermeasure is to change the algorithm so that, when a miner receives two competing blocks, the block for continued mining will be chosen randomly with equal likelihood.

3.3.2.2. Outline of Block Withholding

The attack is to cause damages to the mining pool or fraudulently gain profits by secretly retaining blocks found in mining. The attackees are the operators and participants of the mining pool. As a prerequisite, the attacker must be a participant of the mining pool. The impact of the attack will be bigger if the hash power of the attacker is stronger. The proposed countermeasure against the attack is to improve the bitcoin protocol so that, prior to submission of the block found by a miner to the mining pool operator, the validity of the block will not be confirmed (so that the miner cannot determine whether it is worthwhile retaining the block in secret).

3.3.2.3. Outline of Refund Attacks

The attack is designed so that the attacker hides the transaction history to fraudulently gain profit, utilizing the refund. Two types of attacks have been devised within this category: Silkroad Attack that exploits the vulnerability in the authentication in BIP70 (Bitcoin Improvement Proposal 70); and Marketplace Trader Attack that abuses the existing refund rules in the payment procedure.

The attack takes advantage of the vulnerability that, in making refunds in the bitcoin transaction, it is not possible to identify whether the refund address of the seller of a product is identical to that of the buyer making the payment. A proposed countermeasure is to provide verifiable evidence that cryptographically proves to the seller that the received refund address was approved by the same person that bought the product and approved the payment.

[1] In 2005, an efficient method of attack on SHA-1, a hash function that was used widely as core of cryptography, was published on a blog of a cryptologist. Refer to https://www.schneier.com/blog/archives/2005/02/sha1_broken.html for the blog.

3.3.2.4. Outline of Time Jacking

The attack is designed to try duplicate use by fraudulently manipulating the network time of the blockchain, or to waste calculation resources of other miners or to reduce the speed of transaction approvals. This attack takes advantage of the vulnerability of the bitcoin network that it is possible to put forward or backward the network time of any nodes by transmitting fraudulent version messages. One of the proposed countermeasures is, when verifying the block, to use, as the timestamp of the block, the time calculated from the median values of the timestamps of the blocks in the past.

3.3.2.5. Outline of Tampering

The attack is designed to delay transmission of transactions and blocks from certain nodes, for use in the DDoS attack, for improvement of relative mining capacity of the attacker, or for the duplicate use attack. One of the proposed countermeasures is to change the system so that each node dynamically changes the time to timeout in accordance with the size of the message, etc.

3.3.3. Assessment

3.3.3.1. Results of Risk Assessment

The results of the assessment so far are shown on Chart 3-2 with the following two axes:

- High risk level
- Countermeasures in theory only

The results are shown on Chart 3-2. From the chart, it is known that the attacks having high risk levels and with only theoretical countermeasures are “DDoS (15)” and “Compromise of Cryptographic Algorithm (19).”

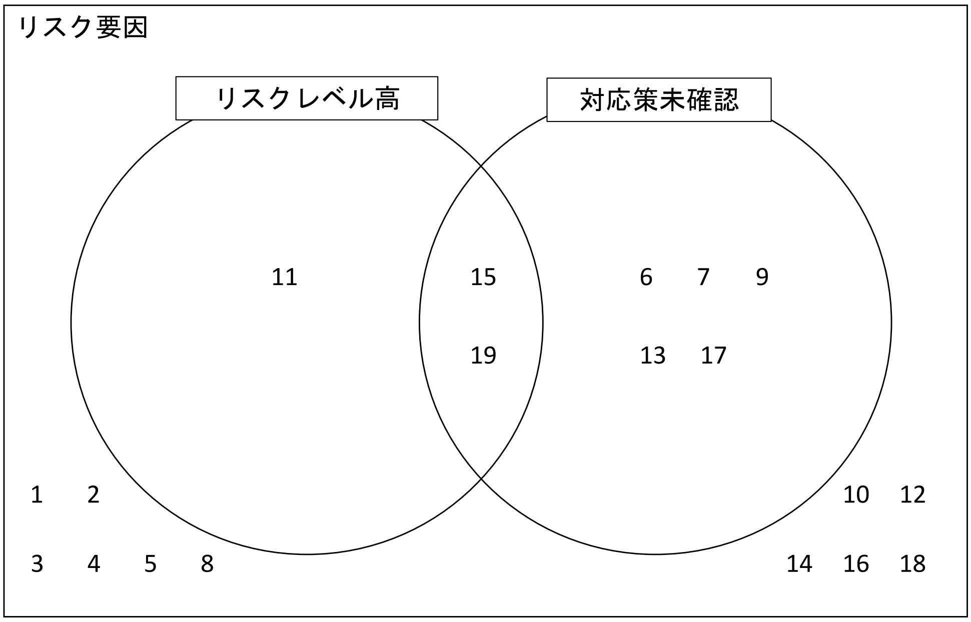


Chart 3-2: Results of Overall Assessment

[図要素]

1. Risk Factors
2. High Risk Level
3. Countermeasures in Theory Only

From this study and the results of the risk assessment in this study regarding the cryptocurrency transactions, it is highlighted that risks exist that damage the asset preservation of the cryptocurrency users or the very values of the cryptocurrencies.

The risk of attacks, including those with the Wallet Theft (11) described above, has already materialized, causing hacking damages to the cryptocurrency exchanges.

The compromise of the cryptographic technology has not, at present, materialized, but this is the risk that the very value of the cryptocurrencies may potentially be lost. At the start of this study, the countermeasures exist only in theory, and the risk level is high that the very value of the cryptocurrencies is damaged. Moreover, it does not appear that the existence of this risk is well known. For these reasons, we determined that it is necessary to perform additional researches on the risk, and we will carry out the substantiative experiment, which will be explained in Chapter 4. Concerning the DDoS attacks that have a high risk level and have countermeasures in theory only, we determined not to perform the substantiative experiment because countermeasures expected to have certain efficacy have been presented.

3.3.3.2. Assessment as Information System Using Cryptographic Technology

The cryptocurrency transactions, for which we carried out the risk assessment as described above, are a mechanism for maintenance and exchange of value on a distributed system that is based on a cryptographic technology called blockchain. The blockchain, which is the underlying technology for the cryptocurrencies, can be considered as an information system that ingeniously utilizes the cryptographic technology. We therefore will focus, in this section, on the blockchain as information system using the cryptographic technology, and, after giving thought to the four principles that need to be considered in using the cryptographic technology appropriately in the information systems in general, we would like to discuss anew if the cryptographic technology is appropriately treated in the current public blockchain.

The four points that need to be considered in using the cryptographic technology appropriately in the information systems in general are as follows:

(1) If the cryptographic technology in use is secure.
(2) If the security is maintained as secure cryptographic communications protocol.
(3) If the security measures using the cryptographic technology are implemented appropriately on the software.
(4) If the secret key is managed appropriately.

First, the points (1) and (2) are the design issues of the information systems that use the cryptographic theory or cryptography. Accordingly, the academic analyses and assessment of security by experts are generally important.

For example, with regard to the security of the cryptographic technology of the point (1), the Cryptography Research and Evaluation Committees (CRYPTREC) assesses the security and performance of implementations, and releases the results for those already in wide use or for those likely to be used widely in the future. These results will be useful.

The point (2) is to see if the system is designed appropriately using the cryptographic communications protocol for realizing the chain with the use of a hash function and electronic signature, and mechanism for the distributed consensus. For example, bitcoin realizes transfer of values without presupposing the existence of the overall operator by utilizing the cryptographic technologies in the generation of the blockchain using the hash function, confirmation of the validity of the transactions using the secret and public keys together with the scripts, and writing into the blocks using the proof of work.

The point (3) is to assess if the appropriate cryptographic method and cryptographic communications protocol are implemented appropriately on the public blockchain. For example, even if a secure cryptographic method is used and an appropriate protocol is designed, a bug in the program that executes the system may render the security of the transactions useless. This means that, when assessing the security of the information systems, it is also necessary to check the risks associated with the implementation.

The point (4) is an issue of the secret key management that must always be examined when the cryptographic technology is used in the information systems. However secure the points (1) through (3) may be, poor management of the secret key in the point (4) endangers the security of the transactions. In particular, in the world of cryptocurrencies, the secret key can in essence be considered identical to the asset. The secret key management is thus critical. In the point (4), it is necessary to adopt and implement the secure cryptographic technology and to manage the secret key appropriately at the same time.

Please note that it is essential for the information systems to meet all of the four points in order to ensure the security. If there is a problem in any one of the four, the transactions based on such information systems cannot be secure.

In the meantime, the blockchain relies more heavily on the cryptographic technology for the security than the information systems using the cryptography in general. Accordingly, it will be all the more important to continue assessing and improving the security with such a viewpoint.

Below, we categorized the 19 attack methods and vulnerability issues that we assessed in this study in consideration of the viewpoint above, with comments. Not all of the 19 attack methods and vulnerabilities are associated with the cryptographic technology, and accordingly, not all of them can be categorized with these points.

Table 3-3: Categorization Based on Points for Appropriate Handling of Cryptographic Technology

Category	Attack Method and Vulnerability	Reason
(1)	19. Compromise of cryptographic algorithm	19. Not designed with the possibility of the compromise of the cryptographic algorithm taken into consideration.
(2)	6.Block discarding or Selfish mining 7.Block withholding 9.Refund attacks 12.Transaction malleability 15.DDOS	6. and 7. The transaction verification process within the block is designed to be realized with the calculation competition of the cryptographic hash functions, which induces creation of blocks suitable for the attacker. 9. Because no cryptographic measures to prove the refund address and payer address are implemented. 12. The vulnerability exists that the transaction ID generated with the hash function can be modified. 15. In the transmission of the transactions, it is designed that the hash value of the transaction, which is called Inventory, is sent, and that the node that receives the Inventory makes the inquiry if the hash value is unknown. Accordingly, by generating a lot of Inventories with illicit transaction hash values, it is relatively easy to cause delays.
(3)	N.A.	
(4)	11. Wallet theft	11. Because the attack is due to the fact that the secret key is not managed appropriately.

3.3.3.2.1 Vulnerability from Viewpoint of (1)

The Compromise of Cryptographic Algorithm (19) falls within this category, which is the vulnerability accompanying the compromise of the cryptographic technology itself. For example, even in the case of the SHA-1 cryptographic hash function that was widely in use, the hash collision was confirmed in 2005 due to the compromise. Many information systems went through the transition from SHA-1 to other hash functions thereafter. The transition was possible because these information systems had the operator, and also because these systems were designed to accommodate changes in the cryptographic technology to be used.

In the meantime, in the cryptocurrency mechanisms, the value itself is stored on the blockchain using the cryptographic technology. This means that any identification of compromise of cryptographic technology can cause serious problems. Bitcoin, which is the research object of this study, uses cryptographic hash functions for various purposes. The major technologies in use and their use purposes are summarized as follows:

Table 3-4: Use Purposes of Cryptographic Technologies Used in Bitcoin

#	Use Purpose of Cryptographic Technologies Used in Bitcoin	Major Purpose	Cryptographic Technology
1	Block hash chain	Maintenance of order continuity of blocks, verification of no missing or added blocks	SHA-256
2	Markle Tree	Verification that the transaction is stored in the block	SHA-256
3	Transaction ID	Pointer for reference to transaction data, verification of integrity of the transaction	SHA-256
4	Bitcoin address	Securement of uniqueness of address by generating the address as hash value of public key	SHA-256, RIPEMD-160
5	Checksum of bitcoin address	Prevention of entry of erroneous address	SHA-256
6	Hash and nonce in Proof of Work	Proof that block verification is completed	SHA-256
7	Digital signature to transaction	Confirmation of transaction creator	ECDSA elliptic curve cryptography

For example, if SHA-256 is compromised, the order continuity of the block will be lost, making it possible to manipulate the blocks in the past or include non-existent transactions in the block without changing the Markle Tree value. The credibility of the blockchain itself will be prejudiced, causing damages to the very value of the cryptocurrencies stored on the blockchain. Bitcoin, unfortunately, does not have the mechanism for transition of the cryptographic technology at present to mitigate such a risk.

3.3.3.2.2 Vulnerability from Viewpoint (2)

Attacks including "6. Block Discarding or Selfish Mining," "7. Block Withholding," "12. Transaction Malleability" and "15. DDoS" fall within this category. For example, the "12. Transaction Malleability" exploits the vulnerability that the transaction ID can be overwritten. This issue surfaces if the transaction is identified only with the transaction ID, and is said to be one of the causes of the Mt. Gox case[2]. Other than this attack, the "6. Block Discarding or Selfish Mining" and "7. Block Withholding" are issues of reversal of blocks or delay in transmission of blocks but have not caused serious problems in terms of the user protection. Generally, however, only time can tell if the information systems using the cryptography are secure. It is also necessary to note that the academic work to demonstrate the security of the protocol as a whole used in the bitcoin system is yet to be completed.

It is necessary to remember that, if there emerge a variety of attack methods that reverse the blocks, or if the attack methods get further sophisticated, there may be the systemic risk for the operators using the virtual currencies as payment means. Furthermore, the cryptographic technologies and protocols used in virtual currencies other than bitcoin may be completely different from those used in bitcoin, which means that these other virtual currencies may not be secure even though no events have been reported for bitcoin to date.

3.3.3.2.3 Vulnerability from Viewpoint (3)

There were no events falling within this category in the 19 research papers examined in this study. Generally, this issue of implementation flaws in the viewpoint (3) (so-called bugs) is most frequently seen in the information systems using the cryptographic technologies. The bitcoin core program is one of the cryptocurrencies implementation programs most intensively reviewed by the developers, and bugs and other flaws have been worked out with thoroughness. However, it is necessary to keep in mind that, in the case of other implementations or other virtual currencies, the security may not be checked so thoroughly.[3]

3.3.3.2.4 Vulnerability from Viewpoint (4)

The "11. Wallet Theft" falls within this category. Currently, the public blockchain is designed so that the users (in this context, cryptocurrency exchanges, retail companies using the virtual currencies as means of payment, and individual users) manage the secret key appropriately. Accordingly, the users need to keep well in mind that, if the presumption is violated, the security of transactions cannot be ensured. Actually, many of the problems of the stolen virtual currencies reported recently were due to attacks exploiting the vulnerabilities related to the secret key management of the public blockchain users. Who manages the secret key in what way is a vital issue in ensuring security of the transactions.

Specifically, in consideration of the fact that, in many cases, the cryptocurrency exchange operators manage the secret key, it is necessary to improve the cybersecurity of these operators[4]. In order to ensure adequate security level, the operators will need to be ready to take measures against vulnerabilities specific to the secret key of the blockchain, in addition to learning fully from the cybersecurity experiences of the information systems utilizing existing cryptographic technologies.

Above, we gave thought to the points (1) through (4) for appropriate handling of the cryptographic technologies in the information systems concerning the transactions on the public blockchain. This methodology has so far been applied to the information systems that are managed by a key operator, but is also considered useful to a certain extent for assessment of the public blockchain that is based on the cryptographic technologies.

[2] The issue of the Transaction Malleability is more or less solved at present, thanks to the introduction of SegWit in 2017.

[3] There was actually a case that some of the hardware wallets had a bug in the implementation and that, when the receiver of the cryptocurrency tried to generate his/her own address, a hacker program that made its way to the system generated the hacker address instead, resulting in remittance of the virtual currencies to the hacker.

[4] As a technique to enhance the security, it is proposed to manage the secret key offline as in the case of the hardware wallets, but it is necessary to also note that there are only several types of hardware wallets even for bitcoin that is supposed to have the largest number of users. If the hardware wallets get widely used by exchanges and end users, a single vulnerability in the implementation of the hardware wallet, if any, can cause damages to many users. It is worthwhile to note that, as only a handful of developers are working on the implementation, there may be a single point of failure even on a distributed system.

3.3.3.3. Conclusion

As shown, we carried out the assessment of the risks associated with virtual currency transactions using the risk assessment method adopted for this study and also from the viewpoints for appropriate handling of the cryptographic technologies in the information systems. As stated, it is essential to take into consideration all of the four points (1) through (4) for ensuring security of the transactions, but as the risk of "11. Wallet Theft" (the point for management of secret key) has already materialized and that of "19. Compromise of Cryptographic Algorithm" will surely materialize over the medium- to long-term period and may do so in the immediate future, we believe we will need to focus on these threats for further review. In "5. Discussion," we will examine the countermeasures that can be taken in the future with focus on these two risks.

3.4. Necessity for Substantiative Experiments

In "3.3. Overall Assessment," we stated that we believe it is necessary to carry out further studies for "15. DDoS" and "19. Compromise of Cryptographic Algorithm" because these are the two largest risk factors that are high in risk level and that do not have countermeasures in place (theoretical measures do exist but no implementation has been developed).

Further researches on countermeasures revealed the following facts.

- 15. DDoS

  Research papers on the countermeasures have theoretical proposals with detailed study of the performance if adopted, accompanied by specific numerical data.

- 19. Compromise of Cryptographic Algorithm

  Research papers on the countermeasures have theoretical proposals, but do not have any discussions on the performance if adopted, and it is necessary to carry out the performance assessment when these additional functions are implemented.

As stated above, we found that, although the countermeasures against "15. DDoS" are not perfect, they are assessed as having certain effects for alleviating the risks. On the other hand, we came to realize that the proposed countermeasures against "19. Compromise of Cryptographic Algorithm" are purely theoretical and that no performance assessment has been carried out to examine the impact on the blockchain network if these are implemented.

The compromise of cryptography is a risk common to the information systems in general, but because the value of the virtual currency depends on the reliability of the cryptographic technologies of the blockchain and the impact of the risk materialization is large, we came to believe that it will be very meaningful to carry out the performance assessment of the measures against the compromise in this study. For objective performance assessment, we believe it best to carry out the substantiative experiments using an international interuniversity public chain such as BSafe network.

3.5. Selection of Object of Substantiative Experiments

The result of the examination of the existing attacks and academic research papers (refer to the examination results of research papers on attacks for details) indicates that the compromise of cryptographic algorithm used in the blockchain is a serious risk to the financial transactions using the blockchain, undermining the very value of such transactions and affecting all the users.

It is proposed to apply the long-term signature technology to the blockchain as countermeasure against the risk of compromise of cryptographic algorithm [1]. In this method, when upgrading the cryptographic algorithm for use with the blockchain[5] to a new one, the data guaranteeing that the blockchain generated prior to the migration was authentic is calculated with the new algorithm, which is then added to the blockchain. This way, even when the cryptographic algorithm used for the blockchain in the past is compromised, it is possible to check the authenticity of the blockchain in the past. The authenticity of the blockchain can thus be guaranteed for a prolonged period of time beyond the life cycle of the cryptographic algorithm.

[5] When adopting a new cryptographic technology, it is necessary to choose an algorithm that is adequately guaranteed to be secure at the time. It is desirable to select, for example, one from among those recommended by NIST (National Institute of Standards and Technology) or CRYPTREC (Cryptography Research and Evaluation Committees).

The blockchain, to which the long-term signature technology is applied, requires additional data on the blockchain that are not necessary in the ordinary blockchain. For this reason, the workload is expected to be heavier. A desktop review, however, will not readily provide the information about the degree of the impact on the nodes (execution performance, data volume, etc.) or on the data communications volume when the blockchain with the long-term signature technology is used for the transactions. Accordingly, we determined that it would be beneficial to collect and analyze such information in the substantiative experiments in this study that would serve as precious guideline for the financial transactions using the blockchain including bitcoin.

4. Substantiative Experiments

In this chapter, we describe the object of the experiments, functions to be examined, results, and review of the experiments.

4.1. Object of the Substantiative Experiments

The object of the substantiative experiments of this study is to examine if, in case the cryptographic technologies used in the blockchain are compromised, the long-term signature technology can be effective as countermeasure, and if so, to assess what impact the long-term signature technology may have on the existing blockchain. In addition to the long-term signature technology, we will also perform experiments of change of the hash function used in Proof of Work and change of the length of the key for use in the transaction signature. In order to examine the impact of these measures on the performance, we will measure the following items:

- Impact on the execution speed at each node
- Impact on the data volume at each node
- Impact on the communications data volume on the network for execution of transactions

We expect that the impacts will be as follows:

- Our expectation of impact on the execution speed at each node

  The long-term signature technology is used for calculation of the historical blocks with a new hash algorithm and will be used just once at the time of the transition. Accordingly, we expect that the impact will be minor, and even if it is not, it will be temporary. On the other hand, the validation of the signature can have a major impact on the processing performance of the blockchain. Accordingly, we believe that the change of the length of the key used for the signature to be described later can have a large impact on the processing performance.

- Our expectation of impact on the data volume at each node

  The block header size will be larger due to introduction of the long-term signature technology but only slightly in comparison to the block size of 1 megabyte. In the meantime, the impact on the data volume will be large because the number of digital signatures increases in line with the number of transactions.

- Our expectation of impact on the communications data volume on the network for execution of transactions

  Most of the data communications of the blockchain are for the transactions, and the impact of the increased size of the block header due to introduction of the long-term signature technology is expected to be minor. On the other hand, the increase in the signature data is likely to impact the data communications volume.

4.2. Functions Examined in Substantiative Experiments and Environment for Experiments

The method of transition, functions to be examined, and environment for the experiments will be described below that are necessary to apply the long-term signature technology to the bitcoin blockchain.

4.2.1. Method of Transition to Use of Long-term Signature Technology

As countermeasure against the compromise of the cryptographic technology used in the blockchain, particularly “19. Compromise of Underlying Cryptographic Algorithms,” a method is proposed where the long-term signature technology is applied to the blockchain. As implementation, two methods are indicated: a method for transition within the original chain (as shown on Figure 6 with reference to risk [19]) (hereinafter referred to as “transition method”); and a method for transition with use of a support chain (as shown on Figure 7 with reference to risk [19]) (hereinafter referred to as “support chain method”). The architecture of each method is shown on Chart 4-1 below. The transition method can be realized with hard fork, while the support chain method with soft fork.
The support chain method, however, is difficult to implement due to the following reasons:

(1) In the support chain method, the transaction showing the reward to the miner on the original blockchain is different from one showing the reward to the miner on the support chain. For this reason, it is not possible to keep the support chain identical to the original blockchain.

(2) The current bitcoin system assumes that there is only one blockchain per network. In order to implement the support chain, it is necessary to modify the storage and P2P layers, increasing the implementation cost.

For these reasons, we determined to implement and assess the transition method only in the substantiative experiments in this study.

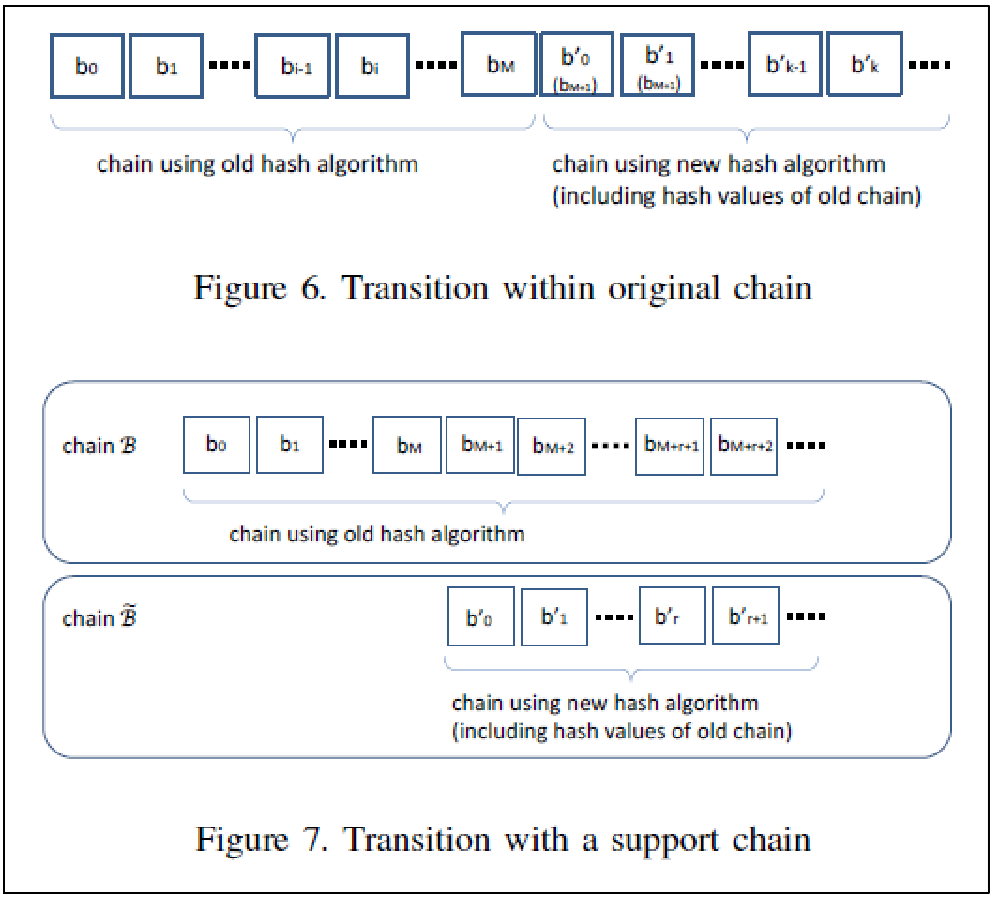

b0
b1
▪▪▪▪
bi-1
bi
▪▪▪▪
bM
b'0 (bM+1)
b'1 (bM+1)
▪▪▪▪
b'k-1
b'k
▪▪▪▪
chain using old hash algorithm
chain using new hash algorithm (including hash values of old chain)

Figure 6. Transition within original chain

chain 𝓑
b0
b1
▪▪▪▪
bM
bM+1
bM+2
▪▪▪▪
bM+r+1
bM+r+2
▪▪▪▪
chain using old hash algorithm
chain 𝓑̃
b'0
b'1
▪▪▪▪
b'r
b'r+1
▪▪▪▪
chain using new hash algorithm
(including hash values of old chain)

Figure 7. Transition with a support chain

Figure 4-1: Methods of Implementation of Blockchain with Long-term Signature Technology

Below, we will describe the outline of the transition method.

b_0 through b_M on Figure 6 above are the blocks (blockchain already generated) to which the data (archiveHash) are to be added to guarantee that the blockchain is legitimate.

These blocks are first grouped in to *r* groups, each having the block size of *s*, from the top of the blockchain.

The archiveHash values calculated from each of the *r* groups are embedded into the new blocks b_{M+1} to b_{M+r}. As a result, the archiveHash values are added to the *r* blocks.

In the experiments in this study, we will use the following parameters:

- *s* = 10 blocks; and *r* = 10 groups

4.2.2. Functions to Be Realized in Substantiative Experiments

In preparation for the possible compromise of the cryptographic algorithm, the archiveHash function will be added, which is based on the long-term signature technology and indicates that the blocks up until a certain point in time are legitimate as at the time. In addition, SHA-256 used in Proof of Work will be altered as well as the length of the ECDSA key used in the transaction signature.

1) Function to calculate and verify long-term signature functionality (archiveHash)

In order to implement the long-term signature functionality, we will add the data to the block header that guarantee that the bitcoin blockchain is valid at a certain point in time. The data added are called archiveHash. This function calculates the hash value when the blocks are given as input that were generated in the past with the compromised cryptographic algorithm. The object is to indicate that the blocks generated in the past are legitimate by means of the output of the function included in the header of the new block.

2) Function to replace hash function (SHA-256) used in Proof of Work

At present, SHA-256 is used in Proof of Work in bitcoin. In anticipation of compromise of SHA-256, we will replace SHA-256 with SHA-512. This way, the functionality to maintain the continuity of the blockchain is realized even when SHA-256 is compromised.
In the substantiative experiments in this study, however, the calculation results of SHA-512 are cut down to 256 bits. This is because SHA-256 is used for a variety of purposes in bitcoin, and the calculation results are retained at a fixed length of 256 bits. Changing the fixed length affects too many places in the system to manage in the substantiative experiments in this study.

3) Function to change key length (and curve) of ECDSA

ECDSA with the 256-bit data length is currently in use in bitcoin for providing signature to transactions. In preparation for compromise of the 256-bit ECDSA, we replace it with the 384-bit ECDSA. This way, the function to prevent falsification of the transaction signature is realized.
The signature algorithm of bitcoin is embedded within the system, but in the substantiative experiments in this study, we determined to implement the signature algorithm with ECDSA using an external library.

4.2.3. Environment for Substantiative Experiments

We use BSafe.network, a network for academic research of the blockchain. At each of the following stations at the universities participating in BSafe.network, we deploy the implemented code and launch the full nodes. These stations are at the following three universities:

Toho University (Japan)

University of British Columbia (Canada)

Keio University (Japan)

4.3. Results of Substantiative Experiments

In the substantiative experiments in this study, we measured the impact of introduction of the long-term signature technology with the following three points:

- Impact on the execution speed at each node
- Impact on the data volume at each node
- Impact on the communications data volume on the network for execution of transactions

Below, we present the measurement results of the substantiative experiments.

4.3.1. Impact on the Execution Speed at Each Node

In the substantiative experiments in this study, we replaced SHA-256 for use in Proof of Work with SHA-512, and the impact of the replacement on the processing performance turned out to be minor. The hash function calculation performance for the blocks that have archiveHash in the header was not very different, either. The archiveHash values are, by the way, included in the block header only at the time of transition when these values are calculated for the blocks created in the past: this process is not executed on an ongoing basis.

Table 4-1: Calculation Time of Hash Functions (microseconds/block)

Hash Function[6]	No. of Measurements	Minimum	Maximum	Average
BlockHash	1,179,648	0. 846	0. 983	0. 888
BlockHashNew	1,310,720	0. 733	0. 983	0. 809
BlockHashArchive	917,504	0. 539	1.222	1.120

On the other hand, it turned out that it took about 10 times the time to generate the signature due to the change in the ECDSA key length. We believe that, because the time required to generate a signature is rather short, it will not be a major issue for light-weight processes such as payment process, but may have certain impacts on full-node operations such as verification of all the transactions within a block.

Table 4-2: Calculation Time of ECDSA (microseconds)

ECDSA Key Length	No. of Measurements	Minimum	Maximum	Average
ECDSA (256 bits)	24,576	21.223	46.720	41.296
ECDSA (384 bits)	1,792	282.156	598.073	575.776

4.3.2. Impact on the Data Volume at Each Node

The results of the experiments in this study indicated that the increase in the block size due to inclusion of the archiveHash values is about 100 bytes. The impact of the change of the hash function used for Proof of Work from SHA-256 to SHA-512 was about 32 bytes (256 bits). Accordingly, the size of the block header will increase by about 132 bytes, or about 0.013% in comparison to the block size of 1 megabyte, at the time of transition to the new system with the long-term signature.

In addition, the change of the ECDSA key length from 256 bits to 384 bits increased the size of the signature by about 50 bytes per signature.

[6] BlockHash: Time to calculate the hash value of a block using the old-version Proof-of-Work hash function
BlockHashNew: Time to calculate the hash value of a block using the new-version Proof-of-Work hash function
BlockHashArchive: Time to calculate the hash value of a block with archiveHash using the new-version Proof-of-Work hash function

4.3.3. Impact on the Communications Data Volume on the Network for Execution of Transactions

In the substantiative experiments in this study, we made changes that affected the block size, such as introduction of the long-term signature and change of the key length, but the experiment results indicated that the increase in the data volume due to these changes had little impact on the volume of data sent and received on the network. Despite the fact that most of the data sent and received are those for transactions (about 96% of all the data were transaction data in the experiments), the experiment results indicated that the increase in the data volume due to the change of the length of the key for the signature algorithm had little impact on the volume of data sent and received on the network.

4.4. Review of Results of Substantiative Experiments

In the substantiative experiments in this study, we checked if the long-term signature technology is effective as countermeasure against compromise of cryptographic algorithms. We implemented archiveHash, which is based on the long-term signature technology and hashes the blocks generated in the past with the new cryptographic algorithm. We also implemented SHA-512 for use in Proof of Work, and changed the ECDSA key length to 384 bits that serves as transaction signature. With these alterations, we were able to demonstrate that the system works on BSafe.network, the academic network.

It is, however, necessary to point out that there are certain constraints on implementation in the substantiative experiments, and the experiment results described in section 4.3 above may not be directly applicable to the real bitcoin environment. Accordingly, we will review the experiment results below, paying attention to the constraints specific to the experiment environment as well as to the points to note with regard to the implementation in the substantiative experiments in this study.

At the same time, we would like to give thoughts to the issues identified in the substantiative experiments when applying the long-term signature to the bitcoin blockchain.

4.4.1. Impact of Application of Countermeasures against Compromise of Cryptographic Algorithms, including Long-term Signature Technology, on Resources Load

4.4.1.1. Impact on Execution Speed

The experiment results indicated that the addition of archiveHash and change of the hash function for Proof of Work do not substantially affect the performance. On the other hand, the change of the ECDSA key length increases the execution time by about 10 times. We believe the latter was strongly affected by the use of the external library for the ECDSA signature algorithm while the current bitcoin system has the algorithm embedded. As the validation of the ECDSA signature is the most time-consuming process for the full nodes, we believe it is important to carry out further study on the impact on the performance in this respect.

4.4.1.2. Impact on Data Volume

The experiment results indicated that, while the increase in the block header size is not likely to have a major impact, the increase in the signature size poses concern over an increase in the storage volume. The volume of the signature data increases in line with the number of transactions, and the number of transactions per block in bitcoin is about 1,000. If we assume that there are two signatures per transaction on average, the block size will increase by about 0.1 megabytes due to the larger signature size. The number of transactions using Segwit[7] is less than half at present, and even if Segwit is used more and more widely, we believe that the impact on the storage is large.

4.4.1.3. Impact on Data Communications Volume

The experiment results indicated that the impact on the data communications is not very large. However, it is necessary to note that the data communications in the substantiative experiments in this study are substantially different from those in the real-world bitcoin system in that the experiments used only three nodes, that only a specific node sent out transactions, that the number of transactions was rather small, and that the number of signatures per transaction was rather small, too.

Despite the constraints in the implementation or environment specific to substantiative experiments, it was demonstrated that application to bitcoin of the long-term signature technology with the use of archiveHash has minor impacts on the resources and can be acceptable for the bitcoin operations.

[7] One of the scalability measures for bitcoin released in August 2017, whereby the singature from each transaction is segregated and stored in the signature area (Witness).

In the meantime, the impact due to the change of the hash function for use in Proof of Work cannot be evaluated fully from the results of the substantiative experiments in this study. At present, the mining process in bitcoin runs on dedicated hardware equipment with an ASIC on board, and cannot be compared directly with the process run on a CPU. The change of the algorithm will render all the hardware equipment useless, and studies as to whether it is acceptable to make such a change need to be carried out as well.

Also, in the substantiative experiments in this study, the impact of the change of the signature size on the storage was not measured. In addition to the impact on resources, it could lead to an increase in the bitcoin transaction fee, which depends on, among other factors, the transaction size. At present, use of the Schnorr signature is being discussed in the bitcoin community that is a technology to reduce the size of the multisignature. Various researches will need to be carried out on use of various signature technologies including the Schnorr signature before the switchover can be discussed seriously.

4.4.2. Practical Issues in Application of Long-term Signature Technology

In consideration of the experience of implementing the code for the substantiative experiments and experiment results, we can raise the following points for consideration when operating the blockchain with the long-term signature incorporated.

4.4.2.1. Points to Consider in Switchover of Cryptographic Algorithm

Our experiments had only a limited number of nodes, and a plurality of changes can be implemented simultaneously. In practice, however, it is desirable that the change of the hash function for use in the Proof of Work, implementation of the archiveHash function and other changes should be released separately. In particular, SHA-256 for use in Proof of Work is actually used widely in bitcoin for various purposes, and we believe that it is essential to carry out in-depth validation work prior to the switchover.

4.4.2.2. Points to Consider Regarding Coexistence of Old and New Cryptographic Algorithms

During the transition period, it is possible that there are both clients compatible with the old cryptographic algorithm only and those compatible with both the old and new cryptographic algorithms in use. For example, problems may arise that the clients not compatible with the new algorithm are unable to confirm the receipt of money because they cannot validate the signature.

4.4.2.3. Points to Consider When Launching New Nodes

When launching new nodes after the long-term signature is incorporated in bitcoin, the blocks generated in the past before incorporation of the long-term signature cannot be trusted until those that have archiveHash incorporated are received. Since these incidents are more likely to occur as the time passes, it may be necessary to re-design the order of downloads of the blocks.

4.4.2.4. Points to Consider in Validation of Transactions by SPV Nodes

The SPV nodes store the block header only, and therefore must trust other old nodes for validation of the transactions. For the validation of old transactions for which the archiveHash values are calculated, it is necessary to obtain the Merkle tree within archiveHash rather than the traditional Merkle tree, and it is necessary to re-design the P2P protocol.

Up until now, use of the long-term signature technology was in theory only and was not demonstrated. The substantiative experiments in this study proved that it is possible to implement the long-term signature technology in the form of the archiveHash function and run the system. We were able to prove that it is possible to incorporate in the bitcoin blockchain a function to guarantee the legitimacy of the information in the blocks created in the past with the use of the compromised cryptographic algorithm, which we believe is one of the fruits of this study.

The experiments, however, are just at an early stage of the researches. As described above, with the incorporation of archiveHash, there still are many points to solve before this single technology can be employed, such as the order of receipt of the blocks and issues of the SPV nodes. We believe it is essential that the bitcoin community should carry out adequate discussions on the technological and economic aspects for solving these issues, and urge the community to start the discussions on these issues at the earliest stage possible.

5. Discussion

This research study has assessed the risks associated with a total of 19 attack methods and vulnerabilities. Cryptocurrency trading entails the risks that have already materialized, such as wallet thefts in which cryptocurrencies themselves are stolen by third parties, as well as latent risks that may damage the value of cryptocurrencies but are hard to predict when to occur, an example being a cryptographic compromise. In addition, attacks like block discarding and selfish mining involve the risk that, by having blocks discarded or delayed, users' transactions in cryptocurrency trading may not be incorporated into blocks or may be delayed.

As stated in 3.3.3.3, this chapter first clarifies issues requiring consideration concerning the management and operation of cryptographic keys as well as cryptographic compromises, and then discusses what goals should be aimed for in the future.

5.1. Issues Requiring Consideration

5.1.1. Cryptographic Technique Compromise

Cryptographic algorithms used in cryptocurrencies are constantly exposed to the risk of being compromised. In particular, if that risk suddenly materializes, the only measure available at present is likely to be a hard fork. In fact, in a 2017 incident in which defects were found in a hash algorithm uniquely implemented in a certain cryptocurrency, a hard fork was performed to address the defects.

This incident provides two points to note.

One is to ensure that determinations have been made as to the suitability of cryptographic algorithms used in cryptocurrencies. In this regard, both the US National Institute of Standards and Technology (NIST) and Japan's CRYPTREC[8] evaluate the security of cryptographic techniques. What is needed is a mechanism that verifies whether a cryptographic technique used in a particular cryptocurrency has undergone security evaluation by these institutions and has been correctly implemented.

Secondly, given that cryptographic algorithms are constantly exposed to the risk of being compromised, measures have to be in place to prepare for cases in which the algorithm is compromised. To satisfy this, the cryptographic algorithm must be implemented in an updatable manner and have a migration plan ready. Special attention needs to be paid to migration as it may require corresponding measures to be taken for users' wallets.

In the proof of concept conducted as part of this research study, we implemented cryptographic compromise countermeasures using a hard fork. Given that the actual Bitcoin system takes no account of the possibility of migrating from the underlying cryptographic algorithms, the full-scale implementation of those countermeasures may require considerable time and effort. Especially, migrating from SHA-256, which is widely used throughout the Bitcoin network, might necessitate 1,000 or more modifications. As such, SHA-256 being compromised would pose a great risk to Bitcoin software. Another concern is RIPEMD-160, which has already been placed on CRYPTREC's Monitored Ciphers List as a cipher not recommended for use. The fact that RIPEMD-160 is still in use to generate Bitcoin addresses suggests that the points mentioned above are not an issue limited to certain cryptocurrencies alone.

In any case, as there exist institutions that standardize, or evaluate the security of, cryptographic techniques, it is useful to refer to the best practices of leading players.

If a cryptographic technique is compromised, that may affect not only cryptocurrency trading but also many other financial systems. The risk of cryptographic technique compromise has been under study in the academic and financial fields, but should be addressed through broad collaboration among academia, industry, regulators, engineers developing cryptocurrencies, and other stakeholders. As long as the value of cryptocurrencies is based on the robustness of the underlying cryptographic algorithms, consideration needs to be given to the development of experts capable of objectively evaluating cryptographic algorithms and their implementation as well as to collaboration with specialized institutions.

[8] CRYPTREC consists of three organs: the Advisory Board for Cryptographic Technology, which is coadministered by the Ministry of Internal Affairs and Communications and the Ministry of Economy, Trade and Industry; and the Cryptographic Technology Evaluation Committee and the Cryptographic Technology Promotion Committee, both of which are coadministered by the National Institute of Information and Communications Technology and the Information-technology Promotion Agency.

5.1.2. Management and Operation of Cryptographic Keys

As described earlier, when trading Bitcoin and other cryptocurrencies, it is paramount to safeguard the users' private keys. Failure to properly safeguard a private key might result in losing the asset tied to that key. In the blockchain transaction system, although the results of a transaction in which value is transferred from one address to another are broadcast as a rule, to whom these addresses belong remains unclarified. Under this system, only the user possessing the private key tied to the recipient address can claim that the value transferred to the address belongs to him or herself. Therefore, in cryptocurrency trading, the theft of a private key is deemed equivalent to the theft of the very cryptocurrency owned by that user.

In traditional financial trading, even if online banking IDs or passwords are stolen, there are cases where the financial institutions compensate the users in one way or another. This is because financial institutions can sometimes disable the stolen accounts before money is ultimately withdrawn in the form of cash. Some financial institutions choose to provide compensation from the perspective of user protection.

In cryptocurrency trading, however, once a private key is stolen and used to record a transaction in a block on the blockchain, it is impossible to undo the block. In other words, there is no way to protect that user unlike in traditional financial trading.[9] Users are asked to fully understand this risk before participating in cryptocurrency trading.

This risk suggests two issues concerning the safeguarding of private keys.

The first is to identify a technological means that ensures the safeguarding of private keys and define standards to objectively evaluate the security of the means.

The other issue is who should safeguard private keys. Although users ought to safeguard their own private keys in principle, their computer literacy also needs to be taken into account. In some cases, cryptocurrency exchange and other service providers keep their users' private keys in custody, which is equivalent to being entrusted with their cryptocurrencies. Consideration has to be given to whether the very action of keeping users' private keys in custody should be deemed equivalent to funds being deposited, as well as to what technological measures are being employed to safeguard private keys.

Another point of note in relation to the second issue is that, when an information system employs a cryptographic technique, the administrator of the information system usually assumes responsibility for managing the key. However, in the case of public blockchains, which have no administrator, the keys need to be managed by the parties using those blockchains, whether cryptocurrency exchanges or ordinary users.

[9] The so-called DAO incident of June 17, 2016, was an example in which the cryptocurrency stolen was restored. In this incident, implementation bugs in the Ethereum blockchain's smart contract functionality were exploited to steal DAO tokens. Subsequently, the Ethereum community implemented a hard fork of the Ethereum blockchain, thereby effectively restoring the stolen DAO tokens. The incident, on the other hand, resulted in the splitting of the Ethereum community, breaking Ethereum into two separate blockchains. The incident taught the blockchain industry a number of lessons. One of them is that, if vulnerabilities exist in a contract that is an application on the public blockchain, it is the platform that addresses them. Another lesson is that a hard fork can be carried out at the convenience of certain stakeholders.

5.2. Measures Requiring Consideration

Based on the measures suggested in 5.1 to safeguard private keys using wallets and address cryptographic algorithm compromise, this section discusses the specific goals of measures conceivable at present. It should be noted, however, that the measures and other matters discussed below need to be continuously reviewed in light primarily of the increasing sophistication of cryptocurrency trading and technological advancements. Careful consideration has also to be given to which institution should develop relevant standards and guidelines. We propose that the goals of those measures be considered going forward by reference to the existing guidelines for financial and information systems.

5.2.1. Measures against Compromise

Possible measures are to set standards for evaluating the security of cryptographic techniques used in cryptocurrencies and to conduct compliance examinations. The results of these examinations need to be considered when cryptocurrency exchange service providers register the cryptocurrencies they handle. Their registration should continue to be reviewed thereafter, taking these results into account.

5.2.1.1. Measures to Urge Evaluation and Monitoring of Security of Cryptographic Techniques Used in Cryptocurrencies

5.2.1.1.1 Development of Security Practice Standards

Clear standards must be set to enable evaluating the security of cryptographic techniques used in cryptocurrencies. To that end, it is conceivable for such standards to be developed and published by FISC,[10] the IPA, and other existing institutions, for example by reference to cryptographic techniques evaluated by CRYPTREC.

5.2.1.1.2 Evaluation and Examination of Security

We propose that security be evaluated pursuant to the aforementioned security practice standards through the following two means: self-evaluation to be conducted by cryptocurrency exchange service providers; and third-party evaluation (audit) to be conducted by self-regulatory organizations, audit corporations, or similar entities. This scheme can be designed to obligate a cryptocurrency exchange service provider to perform self-examination when applying for registration of the cryptocurrencies handled at its exchange, or a self-regulatory organization to evaluate in advance the security of the cryptocurrencies that the provider may handle. In view of the results of these security evaluations, confirmation should be made whether the cryptocurrencies handled by the provider need to be reconsidered by means of the provider's self-review as well as regulatory inspection and oversight.

5.2.1.2. Measures to Urge Migration from Cryptocurrency Algorithm at Risk of Being Compromised to Another

The standards for the examinations mentioned above may be defined to obligate the risk of being compromised to be disclosed if any. In addition, consideration should be given to a framework that, if a cryptocurrency is at greater risk of being compromised, allows regulatory authorities to demand exchange service providers suspend the exchange for fiat currencies, or handling, of the cryptocurrency.

The risk of cryptographic technique compromise has been under study in the academic and financial fields, but should be addressed through broad collaboration among academia, industry, regulators, engineers developing cryptocurrencies, and other stakeholders. As long as the value of cryptocurrency is based on the robustness of the underlying cryptographic algorithms, consideration needs to be given to the development of experts capable of objectively evaluating cryptographic algorithms and their implementation as well as to collaboration with specialized institutions.

It is also imperative that cryptocurrency exchange service providers and developer communities be informed about the findings of these studies.

5.2.1.3. Fostering of Cryptographic Engineers

As discussed above, engineers well versed in cryptographic techniques are essential to nurturing and growing new businesses surrounding cryptocurrency trading. Fostering such businesses is also expected to require existing institutions examining cryptographic techniques to reinforce personnel.

[10] Center for Financial Industry Information Systems (website: https://www.fisc.or.jp/english).

The development of cryptocurrencies increasingly hinges on the existence of engineers well versed in cryptographic techniques. Likewise, securing and fostering engineers skilled at using and applying cryptographic techniques is considered indispensable to businesses surrounding cryptocurrency.

5.2.2. Measures to Enhance Security of Private Keys

While some users manage their private keys on their own, the prevailing practice is believed to be management by cryptocurrency exchange service providers. Accordingly, these providers can be regarded as keeping their customers' cryptocurrencies in custody. Given the importance of being entrusted with customers' assets (private keys), these providers may be required to ensure certain security levels in view of the measures taken by traditional financial institutions and other players serving a similar function. In that event, in addition to requiring the implementation of cybersecurity practices currently available, issues specific to cryptocurrency need to be considered.

For example, it may be necessary to urge those providers to safeguard private keys in hardware wallets or similar environments that are isolated from the network. Besides simply requiring isolation from the network, consideration also needs to be given to the development of security standards for hardware wallets and to the way these wallets should be administered.

Below we look more closely at management using hardware wallets and other measures conceivable at present. Attention, however, should be paid to the fact that these measures are not expected to achieve complete security and the sufficiency of measures will have to be widely discussed, participated by security experts.

5.2.2.1. Measures to Enhance Security of Cryptocurrency Using Hardware Wallets

5.2.2.1.1 Management Using Hardware Wallets

It is necessary to consider a mechanism that urges cryptocurrency exchange service providers to manage all or part of customers' cryptocurrencies in their custody using anti-tampering hardware wallets.[11]

By transferring part of customers' assets in their custody to hardware wallets or similar environments and thereby physically separating these from the network, the providers can prevent hacking from the outside. It should be noted, however, that there are some cryptocurrencies that have no hardware wallets effective thereto or are difficult to be contained in hardware wallets.

5.2.2.1.2 How to Manage Hardware Wallets

Even if cryptocurrency exchange service providers are obligated to store their customers' private keys in hardware wallets, introducing those wallets alone will not necessarily solve the problems. What is additionally needed is to identify all use cases in which hardware wallets are accessed and to define the way those wallets should be managed, including the management of access authority to hardware wallets, the physical isolation of those wallets themselves, and the preservation of evidence of access.

5.2.2.1.3 Security Standards for Hardware Wallets

A mechanism is needed whereby the security of hardware wallets is verified. Although several hardware wallets for Bitcoin have been developed and are currently sold, some of them are not certified in accordance with applicable international standards such as those for information security. The situation demands the definition of security practice standards for hardware wallets as well as the mechanism and institution to assess whether they are implemented appropriately.

As a means of storing cryptographic keys in hardware devices, hardware security modules (HSMs) have been available for some time. To ensure the security of HSMs, the NIST mentioned earlier has developed the Federal Information Processing Standard (FIPS) 140-2 *Security Requirements for Cryptographic Modules*. Similarly, in Japan the IPA[12] operates a testing and validation program for cryptographic modules.[13]

With regards to these existing security practice standards for cryptographic modules, it should be considered to use them as references in developing security practice standards for the management of blockchain cryptographic keys, as well as to utilize their evaluation frameworks.

[11] An anti-tampering hardware wallet refers to one from which the information contained therein cannot be physically removed.

[12] Information-technology Promotion Agency (website: https://www.ipa.go.jp/index-e.html).

[13] A program under which cryptographic modules for use in hardware and software are tested and validated. For details, see https://www.ipa.go.jp/security/english/jcmvp.html.

The fact that hardware wallet products currently lack variety can also be a problem. The lack of product variety means that consumers are forced to choose from a few available models; then, if one of those hardware wallet models is found to have a vulnerability, damage may be suffered by many consumers. To avoid that situation, security practice standards to be satisfied by hardware wallets should be defined and certification methods established. Implementing these is believed to be essential to encouraging new entrants into the wallet development market while offering a variety of choices to consumers.

5.2.2.2. Measures to Enhance Security of Cryptocurrency Using Multi-signature Scheme

The damage caused by attacks against cryptocurrency exchanges has highlighted the importance of adopting a multi-signature scheme. Requiring multiple private keys to validate the signature for one transaction, this scheme represents a measure that urges the introduction of a system whereby the necessary private keys are separately managed by multiple parties. Under this system, the theft of one of the necessary keys alone would not enable the cryptocurrency to be transferred. Attention, however, needs to be paid to some cryptocurrencies whose methods of implementation make the use of a multi-signature scheme impossible.

5.2.2.2.1 Entrusting Part of Private Keys

It is conceivable that, on top of introducing a multi-signature scheme, a system be established whereby a part of the necessary private keys is entrusted to a third party that is not a cryptocurrency exchange service provider. Doing so separates the management of those keys from cryptocurrency exchange service providers and leads the keys to be managed off-line. In that case, even if a cryptocurrency exchange is hacked, the theft of the assets in its custody should be able to be prevented unless third parties to which the keys are entrusted are simultaneously hacked.

Implementing this measure, though, necessitates a number of actions. The first is to define relevant processes, especially the process through which a user trading a cryptocurrency instructs the third party to sign a transaction using the private key. The second is to fully make sure that those processes will not create new vulnerabilities. Furthermore, attention needs to be paid to the possibility that implementing this measure may require not only examining technological aspects but also changing the existing frameworks.

5.2.2.2.2 Defining Matters to Be Complied With by Parties Entrusted with Private Keys

Matters to be complied with by the parties entrusted with private keys need to be determined in view of the management structure required of cryptocurrency exchange service providers. It is desirable that these matters be determined taking into account the administration of traditional financial institutions and similar organizations whose management process is strictly defined.

5.2.2.3. Other Measures to Enhance Security of Cryptocurrency

5.2.2.3.1 Obligation to Disclose Methods of Managing Customers' Cryptocurrencies in Custody

Another conceivable measure is to obligate cryptocurrency exchange service providers to disclose the methods of managing customers' cryptocurrencies in their custody. It is also possible to obligate them to clarify the customer private key management method and system per cryptocurrency, according to which regulatory authorities conduct inspection and supervisory monitoring. This measure can be said to enable customers to select those providers in terms of security.

At the same time, in order to enable or empower ordinary customers to make informed decisions, it is also advisable to explore measures to increase their understanding of how important private keys are to cryptocurrency.

5.2.2.3.2 Storage in Multiple Addresses

It may also be necessary to introduce a measure that obligates cryptocurrency exchange service providers to store customers' cryptocurrencies under their management in multiple addresses. If such customer assets under their management are divided among different addresses, the theft of private keys stored in one address would not lead to the loss of the cryptocurrencies of all customers. As this measure poses no technological challenges, it is expected to be introduced with relative ease.

5.2.2.3.3 Users’ Practices

Users must recognize anew that, in cryptocurrency trading, losing the private key is equivalent to losing the cryptocurrency itself. Presumably, they do not carry a large amount of cash in their wallets on a daily basis; but they may be unaware of how risky it is to store a large amount of cryptocurrency in one wallet or entrust their private keys to third parties like cryptocurrency exchange service providers. A mechanism is needed to make them fully aware of the risk involved in those practices.

5.3. Recent Uncertainties Surrounding Cryptocurrency Trading

We have so far discussed the security of cryptocurrency trading mainly from a technological perspective and considered necessary measures. It should be noted, however, that trading cryptocurrencies involves not only technological risks but also risks derived from the lack of governance over developers, miners, and other community participants.

Recently, against the backdrop of rising exchange rates of cryptocurrencies against fiat currencies (i.e., cryptocurrency prices) and other trends, users trading cryptocurrencies have been increasing. Especially in the case of Bitcoin, with the growing number of users, the delayed processing of transactions has been a problem in the past few years (the so-called Bitcoin scalability problem). Over how to address this problem, interests collided between developers and miners. Consequently, one group implemented the Segregated Witness (SegWit) soft fork while the other initiating a hard fork, with the result that the split of the cryptocurrency led to a new currency, Bitcoin Cash. Thereafter, the situation escalated into the hard fork that created another new cryptocurrency split from Bitcoin.

These developments have created a number of uncertainties typified by those listed below. While they are discussed primarily keeping Bitcoin in mind, trading other cryptocurrencies may also be susceptible to uncertainties caused by community governance issues. Accordingly, when trading a cryptocurrency, it is crucial to pay attention to the initiatives being taken by its developers, miners, and other stakeholders, as well as to how the cryptocurrency price has been trending.

5.3.1. Trading Restriction at Time of Fork

When SegWit was implemented, exchanges temporarily suspended Bitcoin trading, with the result that home appliance mass retailers and other stores having accepted payments in Bitcoin stopped doing so. Although this did not cause much confusion because users had been properly preinformed through email and websites, it is necessary to examine whether these measures are sufficient to address trade restrictions in the future that are likely to involve a greater number of users.

5.3.2. Cancellation of SegWit2x

SegWit2x, which had been planned to be carried out via a hard fork in November 2017 for the purpose of increasing the block size from one megabyte to two megabytes, was suddenly called off. Although having no apparent impact on users due to being cancelled, the planned SegWit2x seemingly costed cryptocurrency exchange service providers considerable time and effort in devising their measures and responding to user inquiries, among other actions. As this development suggests, the decision-making process concerning cryptocurrencies entails the uncertainties inherent therein. The current environment in which cryptocurrency exchange service providers have to individually deal with those uncertainties must be reconsidered going forward.

Incidentally, the planned SegWit2x is said to have been cancelled owing to it requiring a hard fork and having no protection against replay attacks (to be discussed below).

5.3.3. Replay Attacks following Hard Fork

A replay attack is a form of attack whereby, following a hard fork, a transaction on the original blockchain is also executed on the forked blockchain without the user's intent in the event of the latter not employing a signature specific thereto. If the planned SegWit2x hard fork had been carried out, that would have led to the coexistence of the SegWit blockchain and the forked SegWit2x blockchain; in that event, implementing a transaction on the SegWit blockchain would have caused the identical transaction to be executed on the SegWit2x one, risking a great deal of confusion among users.

5.3.4. Treatment of Forked Coins

When a coin is split via a hard fork, each pre-fork user is, in principle, to receive the forked coin in the same amount as the user had of the pre-forked coin. Under the present practice, however, the possibility for each user to receive it depends on whether the cryptocurrency exchange service provider handles the forked coin because treatment differs from one provider to another. In other words, users' asset value may change according to their providers' response to the forked coin.

5.3.5. Cryptocurrency Exchange System Failure

Occasionally there have been cases where a cryptocurrency exchange system failure made it impossible to conduct cryptocurrency trading. In addition to the problem of cryptocurrency users unable to trade, such failures sometimes affect other cryptocurrency exchanges as well. In one actual case, a cryptocurrency exchange, which engaged in cryptocurrency proprietary trading based on prices quoted on another exchange, suffered damage due to the latter exchange's system failure that caused the prices displayed to deviate from the market prices.[14] Given the extremely high volatility of cryptocurrency prices, the present trading systems that permit trades at prices deviating from market prices to be established can be characterized as vulnerable.

[14] This indicates that the system was not equipped with the capability to stop orders placed at prices deviating from the market prices. A problem like this occurred on Japan's stock exchange. In the so-called J-Com incident, a security brokerage firm mistakenly placed an order to sell 610,000 shares in the personnel service company J-Com (currently, Like) at one yen per share when it actually intended to sell one share at 610,000 yen. The order caused the share price to fluctuate wildly, and the brokerage firm to suffer a massive financial loss.

5.4. Issues to Be Further Researched and Studied in the Future

Cryptocurrencies rely on cryptographic techniques more heavily than traditional financial systems have. Nevertheless, no programs are in place to evaluate the security of, and certify, those techniques used in cryptocurrencies. Moreover, the prevailing practice is that the safe storage of private keys for cryptocurrency trading is up to those who use them (i.e., cryptocurrency exchange service providers or users).

This situation calls for research and studies on security practice standards for cryptocurrency trading.

5.4.1. Research and Studies on Security Practice Standards for Cryptocurrency Trading

As mentioned earlier, NIST and CRYPTREC evaluate the security of cryptographic techniques in the United States and Japan, respectively. In addition, the standardization of cryptographic techniques applied to financial trading is discussed at the Subcommittee 2 of the Technical Committee 68, a forum that deals with the security of financial services at the International Organization for Standardization. The issue being discussed at these institutions is the application of cryptographic techniques to traditional information systems, which means that the standards being worked on are premised on each system having the central administrator.

Cryptocurrency trading, however, is processed by decentralized systems that have no administrator. Hence, at present the users (i.e., cryptocurrency exchange service providers or ordinary users) are compelled to single-handedly decide the ways and means of safeguarding private keys and ensure whether cryptographic techniques implemented on blockchains are properly handled.

These circumstances suggest the need to research and study the following themes in the future:

Research and studies toward developing security practice standards for cryptographic techniques used in cryptocurrencies

- Research on evaluating the security of existing cryptographic techniques
- Study on issues toward evaluating the security of, and developing security practice standards for, wallets used for cryptocurrency trading
- Study on issues toward evaluating the security of, and developing security practice standards for, cryptographic techniques used in cryptocurrency

Engaging in research and studies on these themes is likely to demand collaboration not only with existing institutions performing security evaluation but also with self-regulatory organizations that are expected to be formed in the future to govern cryptocurrency exchange service providers.

5.4.2. Research and Studies on Private Key Custody Systems

As described earlier, thefts of cryptocurrencies have occurred in Japan and abroad, exploiting the vulnerabilities of the systems of cryptocurrency exchange service providers. Nonetheless, with the growing popularity of cryptocurrency trading, there exist users who entrust considerable assets to such providers.

While cryptocurrency exchange service providers are naturally required to strengthen their information security practices to protect users, some also call for considering the creation of systems whereby private keys for cryptocurrency trading are entrusted to third parties. Whether this idea merits serious consideration needs to be examined. Then, if that results in recommending the creation of third-party custody systems, research and studies on the following themes may be needed:

Research and studies toward devising third-party custody systems for cryptocurrency private keys

- Research and study on the methods of generation and storage of private keys to be entrusted (identification of matters to be complied with by trustees)
- Research and study on vulnerabilities existing at the time of users instructing transactions to be signed
- Study on issues toward designing third-party custody systems for private keys

6. Appendixes

6.1. Details of Risks Discussed in the Studied Papers

6.1.1. Double-Spending (or Race) Attack

6.1.1.1. Outline

This attack is aimed at obtaining a product from the vendor without the attacker paying for it with its own Bitcoins. The attacker achieves this aim by tricking the target vendor into accepting the transaction that will subsequently become invalid. This attack targets a vendor (*V*) allowing for fast payments that accept transactions without fully verifying the legitimacy of blocks.

In order for this attack to succeed, the attacker (*A*) needs to know the Bitcoin address and IP address of *V*, the target, but does not need to access the target's private key or computer (whether desktop or mobile terminal).

This attack is conducted following the steps below.[15] (See Figure 6-1: Schematic overview of double-spending [or race] attack.)

1. *A* creates a transaction (*TR_A*) that uses the same Bitcoins as those used in the payment transaction to *V* (*TR_V*). During that process, *A* changes the payee of *TR_A* to an address under *A*'s control, from *V* that is the payee of *TR_V*.
2. *A* sends out *TR_V* and *TR_A* in the manner that meets the following requirements:

   (1) *V* first receives *TR_V*, ahead of *TR_A*; and

   (2) *TR_A* is confirmed in the blockchain network, as a result of which *TR_V* is rejected.

These two requirements can be met by taking the steps below.

To satisfy the first requirement, *A* connects as *V*'s immediate neighbor node in the P2P network.[16] Additionally, *A* creates one or more helper nodes (*H*) and ensures that *H*s do not connect directly to *V*. Having set this connection environment, *A* sends out *TR_V* to *V*, and thereafter *TR_A* to *H*s. In consequence, *V* receives *TR_V* first, then receives *TR_A* after a certain amount of time that is the total of the interval between sending out *TR_V* and *TR_A* and of the difference in time it takes for the two transactions to propagate in the P2P network to *V*.

Moreover, to increase the probability of satisfying the second requirement, *A* takes such steps as (i) sending out *TR_A* ahead of *TR_V* and (ii) using multiple helper nodes to have *TR_A* spread faster in the network. As regards (i), to meet the first requirement at the same time, *A* can send out *TR_A* a certain amount of time in advance of sending out *TR_V*. That amount is the difference in time it takes for the two transactions to propagate in the P2P network to *V*.

This attack exploits the Bitcoin network's vulnerability that derives from the several seconds it takes for a transaction to propagate from one node to another. In the event that two transactions having the same input but different outputs spread through different routes, said several seconds delay *V*'s detection of the fraudulent transaction.

[15] G. O. Karame, E. Androulaki, and S. Capkun, *Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin*.

[16] The Bitcoin protocol dictates that *V* continue to accept connection requests from other nodes until the maximum number of its inbound connections (i.e., connections to *V* from others) has been reached. Therefore, the knowledge of *V*'s IP address enables *A* to try to connect directly to *V*.

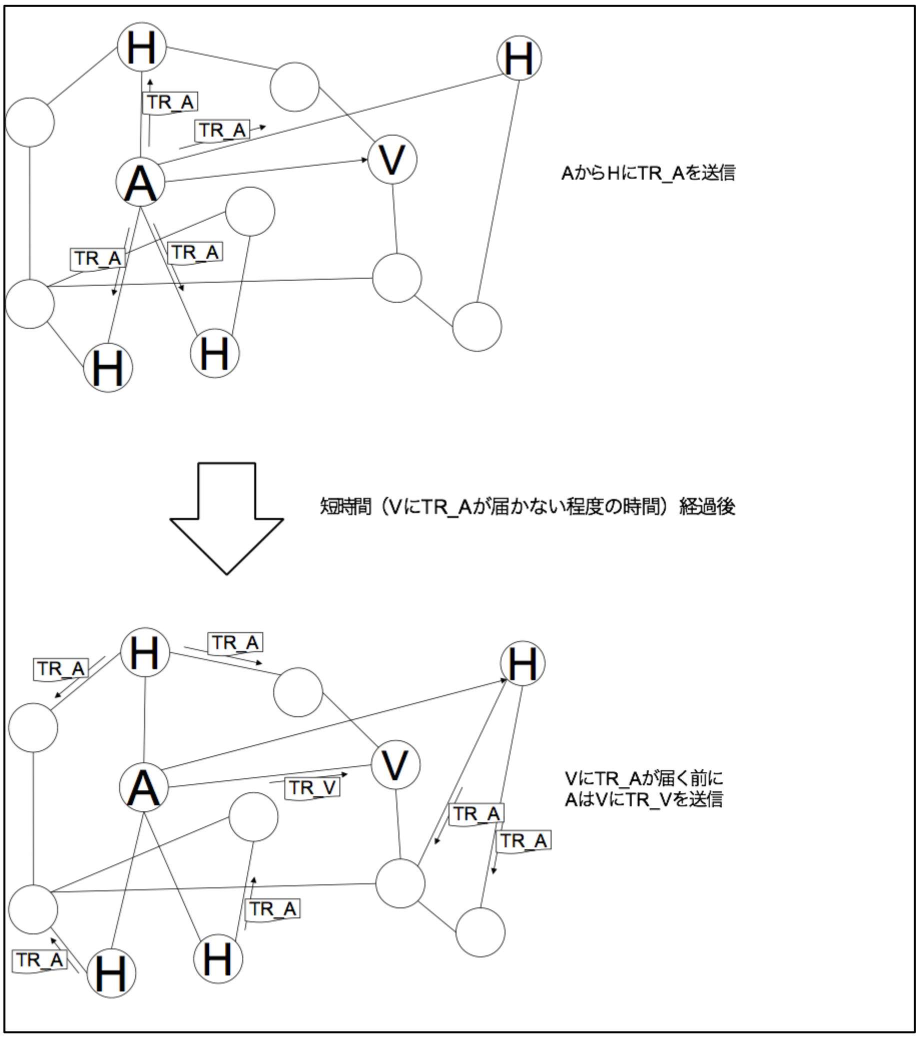


Figure 6-1: Schematic overview of double-spending (or race) attack

=============

図 6-1 の訳

1. *A* sends out *TR_A* to *H*s.
2. After a brief period of time (an amount of time during which *TR_*A does not reach *V*)
3. Before *TR_A* reaches *V*, *A* sends out *TR_V* to *V*.

=============

6.1.1.2. Countermeasures

The following three countermeasures are proposed against this attack:

- Using a listening period

V adopts a listening period of several seconds before providing its product to *A*. During this period, *V* monitors all the transactions it receives, checking whether any of them attempt to double-spend the coins.

This countermeasure can be circumvented, however, if *A* sends out *TR_V* first and then, after a period of time longer than the listening period, sends out *TR_A*. In that case, as the probability of *TR_V* being confirmed and *TR_A* rejected in the blockchain network increases in proportion to the length of the listening period, *A* has to increase the number of *H*s for its attack to succeed. Therefore, although the probability of the attack succeeding decreases in proportion to the length of the listening period, that also means customers are forced to wait for a longer period time, resulting in decreased convenience. As this countermeasure does not require technological upgrades, *V* can easily implement it.

- Introducing observer nodes in the network

V introduces observer nodes that relay all the transactions they receive to *V*. Receiving *TR_A* from one of those nodes enables *V* to detect the double spending within several seconds. The problem is that, to ensure any double spending is detected, *V* has to introduce a considerable number of observer nodes capable of connecting to many nodes, which makes implementing this countermeasure costly to *V*.

- Communicating alert messages

  A mechanism can be introduced whereby Bitcoin nodes, upon receiving transactions having the same input but different outputs like *TR_V* and *TR_A*, broadcast alert messages. The Bitcoin network has originally been equipped, albeit for different purposes, with the mechanism to send an alert message to all the clients. Hence, this countermeasure does not incur any additional cost to *V* and is impossible for *A* to circumvent. Even if *A* manages to preclude *V* and its observer nodes from receiving *TR_A*, a considerable number of other Bitcoin nodes receive both *TR_A* and *TR_V*; then each of these nodes immediately broadcasts an alert message, which reaches *V* within several seconds.

6.1.1.3. Assesment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 1 (Low)

The assessment is based on the grounds below.

- Impact on users
  - ➢ Scope: Broad ・・・ Anyone could be targeted by this attack.
  - ➢ Severity: Low ・・・ The financial damage would be small due to this attack primarily occurring only where payments are made using fast payments.

- Impact on financial trading systems
  - ➢ Scope: Narrow ・・・ The impact of a blockchain fork would be small.
  - ➢ Severity: Low ・・・ The impact of a blockchain fork would be small.

- Occurrence probability
  - ➢ Ease of attack: Low ・・・ This attack could occur only where payments are made using fast payments.
  - ➢ Incentive: Low ・・・ Proceeds would be insignificant as this attack could occur only where payments are made using fast payments.

6.1.2. Finney Attack

6.1.2.1. Outline

A form of double-spending attacks, the Finney attack is aimed at obtaining a product from the vendor without the attacker paying for it with its own Bitcoins. The attacker achieves this aim by tricking the target vendor into accepting the transaction that will subsequently become invalid. A particularly likely target for this attack is a vendor (*V*) that waits for one confirmation block or less before accepting the transaction, thereby deeming the payment complete without fully verifying the legitimacy of blocks.

In order for this attack to succeed, the attacker (*A*) needs to have the capability of pre-mining blocks that contain transactions between the nodes it controls.

This attack is conducted following the steps below.[17] (See Figure 6-2: Schematic overview of Finney attack.)

1. *A* pre-mines a block that contains a transaction (*TR_AA*) between the nodes it controls (can be a payment from *A* to itself), and keeps the block without broadcasting it.
2. *A* creates a transaction to *V* (*TR_AV*) using the same coins.
3. *A* waits for *TR_AV* to be accepted by *V*. *TR_AV* is contained in a blockchain (*B*).
4. Once receiving the product from *V*, *A* sends the pre-mined block to the network, creating *B*'s fork (*B'*).
5. If the next mined block is added to *B'*, *B'* becomes the longest, resulting in *B* being ignored and *TR_AV* becoming invalid.[18]
6. Owing to *TR_AA* contained in *B'*, *A* retrieves the coins to be paid to *V*.

This attack exploits the vulnerability existing in small-value transactions in which *V*, for customer service purposes, has to accept the payment without waiting for a sufficient number of confirmation blocks on the Bitcoin network.

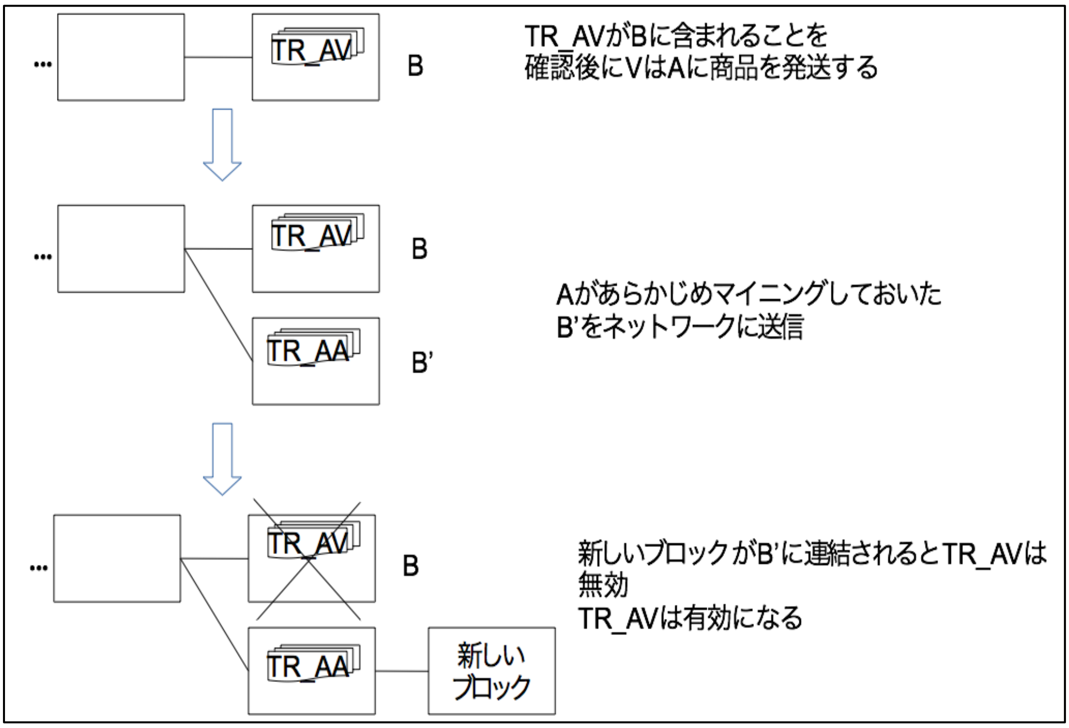


Figure 6-2: Schematic overview of Finney attack

[17] M. Conti, S. Kumar E, C. Lal, and S. Ruj, *A Survey on Security and Privacy Issues of Bitcoin*.

[18] As both *TR_AA* and *TR_AV* are effective transactions from the perspective of miners other than *A*, whether the next block is added to *B* or *B'* depends on luck.

=============

図 6-2 の訳

1. Having made sure that *TR_AV* is contained in *B*, *V* delivers the product to *A*.
2. *A* sends the pre-mined *B'* to the network.
3. If a new block is added to *B'*, that makes *TR_AV* invalid and *TR_AA* valid.
4. New block

=============

6.1.2.2. Countermeasure

To counter this attack, *V* must affirm multiple steps of confirmation before providing the product—that is, wait until a chain of multiple blocks has been added to the end of *B* that contains *TR_AV*. As this countermeasure does not require technological upgrades, *V* can easily implement it. However, this countermeasure is not perfect and may be circumvented, in particular by means of the advanced form of the Finney attack to be explained in the next section.

6.1.2.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 1 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted by this attack.
  - Severity: Low ・・・ The financial damage would be small due to this attack primarily occurring only where payments are made using fast payments.

- Impact on financial trading systems
  - Scope: Narrow ・・・ The impact of a blockchain fork would be small.
  - Severity: Low ・・・ The impact of a blockchain fork would be small.

- Occurrence probability
  - Ease of attack: Low ・・・ This attack could occur only where payments are made using fast payments, and also requires the attacker to possess mining capabilities.
  - Incentive: Low ・・・ Proceeds would be insignificant as this attack could occur only where payments are made using fast payments.

6.1.3. Brute-Force Attack

6.1.3.1. Outline

An advanced form of the Finney attack, this attack is aimed at obtaining a product from the vendor without the attacker paying for it with its own Bitcoins. An attacker perpetrating the Finney attack mines only one block that contains a transaction convenient to itself. On the other hand, an attacker perpetrating the brute-force attack mines the ensuing blocks as well to increase the probability of success of the attack. By means of this tactic, the attacker can trick the target vendor into accepting the transaction that will subsequently become invalid, achieving its aim.

In order for this attack to succeed, the attacker (*A*) must have sufficient hashpower to quickly mine the number of blocks that is equal to the number of confirmations the vendor (*V*) requires.

This attack is conducted following the steps below. (See Figure 6-3: Schematic overview of brute-force attack.)

1. *A* pre-mines a block, which contains a transaction (*TR_AA*) between the nodes it controls (can be a payment from *A* to itself), and the ensuing blocks; and keeps these blocks without broadcasting them.
2. *A* creates a transaction to *V* (*TR_AV*) using the same coins.
3. *A* waits for *TR_AV* to be accepted by *V*. If *V* needs to wait for *x* number of confirmations before accepting a transaction, *x*–1 number of blocks are meanwhile added after the block that contains *TR_AV*.
4. Having received the product from *V* and secretly mined *x*+1 or greater number of blocks, *A* sends these blocks to the network, creating the blockchain (*B*)'s fork (*B'*).
5. As this makes *B'* the longest, resulting in *B* being ignored and *TR_AV* becoming invalid.
6. Owing to *TR_AA* contained in *B'*, *A* retrieves the coins to be paid to *V*.

Even if *V* waits for multiple confirmation blocks in the Bitcoin network before accepting the payment, *A* with sufficient hashpower is able to turn the blocks containing the confirmed payment from *A* to *V* into orphan blocks. That is the vulnerability the brute-force attack exploits.

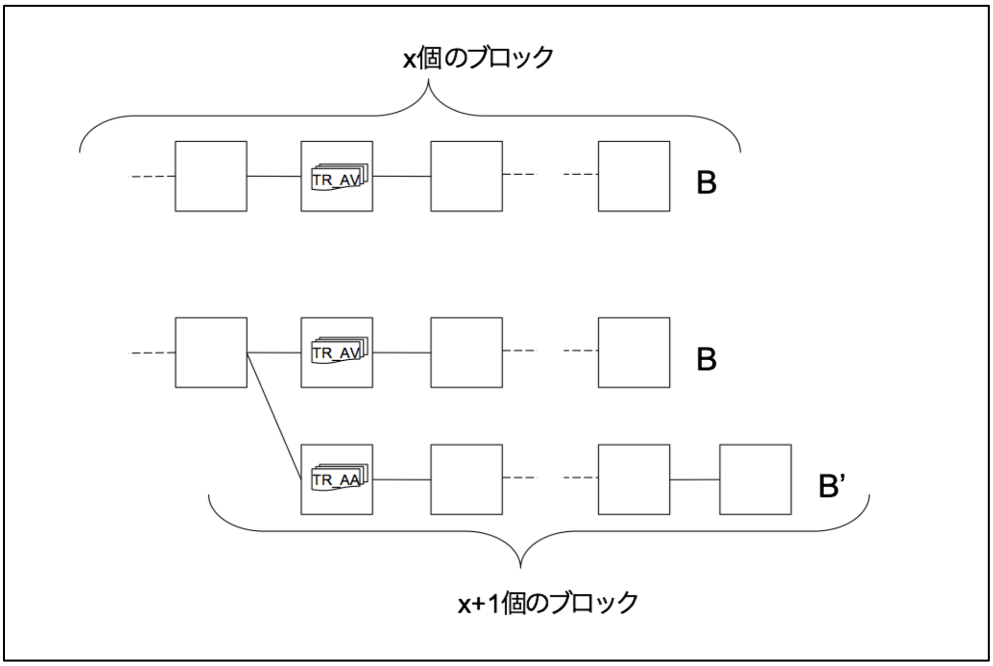


Figure 6-3: Schematic overview of brute-force attack

=============
図 6-3 の訳
1. *x* number of blocks
2. *x*+1 number of blocks
=============

6.1.3.2. Countermeasure

The probability of the success of this attack is determined by *A*'s hashpower as a proportion of the network's total hashpower and by the number of confirmations *V* waits for. The lower the proportion or the greater the number, the lower the probability. We use a model that simplifies the Bitcoin network.

According to theoretical calculations, if *A* controls 10% of the network's total hashpower and *V* waits for six confirmations, the probability of the success of this attack is 0.1% or less.[19] However, as waiting for six confirmations takes about an hour, the circumstances in which *V* can use this countermeasure are limited.

6.1.3.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 6 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted by this attack.
  - Severity: Medium ・・・ The financial damage would be limited and depends on the situation.

- Impact on financial trading systems
  - Scope: Medium ・・・ A major fork might occur.
  - Severity: Medium ・・・ A major fork might occur.

- Occurrence probability
  - Ease of attack: Low ・・・ The attacker must possess substantial capabilities.
  - Incentive: Medium ・・・ Proceeds would be limited to moderate amounts.

[19] Meni Rosenfeld, *Analysis of Hashrate-Based Double-Spending*, Figure 4.

6.1.4. Vector 76 (or One-Confirmation) Attack

6.1.4.1. Outline

A combination of race and Finney attacks, this attack is aimed at fraudulently withdrawing coins from the target, using an invalid transaction. The attack targets exchanges, Bitcoin mixers, and other entities providing accounting services off-chain.

In order for this attack to succeed, the attacker (*A*) must be capable of mining and know the target's address.

This attack is conducted following the steps below.[20] (See Figure 6-4: Schematic overview of vector 76 [or one-confirmation] attack.)

(1) *A* has a full node (*N_A*) that connects only to the node of its target (*V*), in addition to another full node (*N_B*) that is linked to one or more nodes.

(2) *A* creates two transactions that use the same coins: one is a deposit transaction into *A*'s account at *V* (*TR_dep*), and the other is a payment transaction into *A*'s own wallet (*TR_AA*). Both transactions remain unsent to the network at this stage.

(3) *A* mines a block in such a way that it contains the first transaction. Upon completion of mining, instead of publishing the block, *A* simultaneously executes the following two actions:

- Sends the first transaction *TR_dep* to *N_A*; and
- Sends the second transaction *TR_AA* to *N_B*.

(4) *A* sends the pre-mined block to *N_A*.

(5) Having seen the deposit transaction into *A*'s account, *V* pays the corresponding amount into *A*'s wallet, whereupon *A* immediately withdraws that amount.

(6) As the number of nodes *N_A* connects to is limited, most of the nodes throughout the network accept *TR_AA* that was sent to *N_B*. Consequently, *TR_dep* that was sent to *N_A* is rejected in the network.

The vulnerability this attack exploits is that, if the target accepting a payment with one confirmation only is directly accessed and receives a block containing a double-spending transaction, the target is unable to immediately recognize that the block is fraudulent.

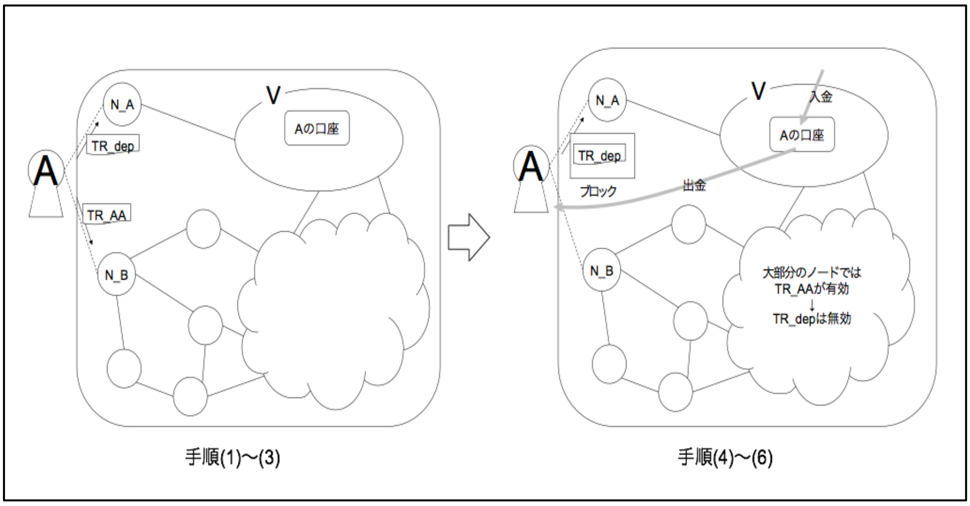


Figure 6-4: Schematic overview of vector 76 (or one-confirmation) attack

=============

図 6-4 の訳

1. *A*'s account
2. Steps 1 to 3
3. Depositing
4. *A*'s account
5. Block
6. Withdrawing

[20] sgornick, "Vector76 Double Spend Attack?," Reddit, https://www.reddit.com/r/Bitcoin/comments/2e7bfa/vector76_double_spend_attack. (See paragraphs starting with the sentence, "The attack would be carried out as follows.")

7. Most of nodes regard *TR_AA* as valid
   TR_dep becomes invalid
8. Steps 4 to 6

=============

6.1.4.2. Countermeasures

The following countermeasures are proposed against this attack:

- Do not accept a payment with one confirmation only. However, waiting for two or more confirmations takes about 20 minutes, the circumstances in which *V* can use this countermeasure are limited.
- Do not use transactions sent from a node that is connected inward.[21] Although this countermeasure requires the Bitcoin client to be upgraded, it should be easily implemented.
- Do not use a static IP address. As this countermeasure can be deployed by simply changing *V*'s network settings, it should be easily implemented.

6.1.4.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 2 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 2 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Narrow ･･･ The potential targets are exchanges, Bitcoin mixers, and other entities providing accounting services off-chain.
  - Severity: Low ･･･ The potential targets are exchanges, Bitcoin mixers, and other entities providing accounting services off-chain.
- Impact on financial trading systems
  - Scope: Narrow ･･･ The number of potentially targeted exchanges is small.
  - Severity: Low ･･･ Damage to the systems would be minor.
- Occurrence probability
  - Ease of attack: Low ･･･ The attacker must possess substantial capabilities.
  - Incentive: Medium ･･･ Proceeds would be limited.

[21] A node that is connected to *V* from another party can be a node that *A* installs for the purpose of directly connecting to *V*, as is the case of *N_A*.

6.1.5. >50% Hashpower (or Goldfinger or Majority) Attack

6.1.5.1. Outline

Backed by more than 50% of the network's total hashpower, this attack is aimed chiefly at the double spending of coins and the monopolization of mining. The attack targets the entire blockchain network including miners, exchanges, and users.

The attacker (*A*) needs to obtain more than 50% of the network's total hashpower, which can be achieved through an oligopoly in or collusion between mining pools and a large capital injection by government or major corporations. If *A* attains more than 50% of the network's total hashpower, it becomes able to have many blocks added after the block that contains a fraudulent transaction. As a result, *A* acquires the ability to cause double payments and impede legitimate transactions, as well as to monopolize mining, thereby gaining all rewards.

This attack exploits the consensus formation algorithm in Bitcoin transactions that is based on the majority of hashpower.

6.1.5.2. Countermeasure

It is believed that, as long as blockchains adopt a proof-of-work protocol, there is no countermeasure against this attack. The only deterrent to this attack is that, as the miner or mining pool that has mined each block is published in the Bitcoin network, the occurrence of this attack would be immediately known to the public, with the result of the Bitcoin system losing reliability and the value of Bitcoin plummeting. As such developments would be detrimental to *A*'s interests too, it is presumed that there is virtually no incentive for *A* to carry out this attack unless *A*'s motive is to destroy the Bitcoin network.

Also, using a proof-of-stake algorithm in lieu of the proof-of-work protocol is expected to mitigate the risk of this attack further. Under the proof-of-work protocol, the greater the hashpower, the higher the probability of success in having the block accepted. Under the proof-of-stake algorithm, however, that probability increases in proportion to the amount of coins held or the length of the period during which they have been held; therefore, if *A* is to increase the probability of success of the attack, it would have to hold a large amount of coins, which makes the attack costly. In addition, even if succeeding in the attack, the coins in *A*'s possession would decrease in value. On account of these factors, the proof-of-stake algorithm is believed to be robust against this attack.[22]

6.1.5.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 3 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted by this attack.
  - Severity: Medium ・・・ The financial damage would be limited and depends on the situation.

- Impact on financial trading systems
  - Scope: Broad ・・・ This attack, especially the Goldfinger one, might disable these systems.
  - Severity: High ・・・ This attack, especially the Goldfinger one, might disable these systems.

- Occurrence probability
  - Ease of attack: Low ・・・ Acquiring more than 50% hashpower requires the attacker to have considerable resources.
  - Incentive: Low ・・・ The occurrence of attack becoming public knowledge would decrease the value of Bitcoin, thus reducing the attacker's proceeds.

[22] The proof-of-stake algorithm also has problems. Firstly, as the probability of success in having the block accepted increases in proportion to the amount of coins held or the length of the period during which they have been held, many people would endeavor to increase currency holdings, leading to a decline in the liquidity of the currency. Secondly, as this algorithm enables currency holders to earn rewards by just holding them without having hashpower, it would lower the incentive to add blocks to the longest blockchain, thereby hindering the convergence of blockchains. Another problem is unfairness that exists its mechanism whereby early adopters are able to obtain a greater amount of coins.

6.1.6.1. Outline

This attack is aimed at earning a more than fair reward by having honest miners and mining pools waste their computational resources and thereby facilitating the adoption of the attacker's block verification results. The attacker keeps the blocks it has mined unpublished to honest miners; then publishes these blocks at an opportune time, such as immediately after an honest miner mined a block, with the aim of increasing the possibility for the blocks the attacker has mined to be adopted.

As such, the attack targets honest miners and mining pools.

Since the probability of the attacker (*A*) earning a more than fair reward increases in proportion to *A*'s computational power, *A* is usually a mining pool rather than a single node.

This attack is conducted following the steps below. (See Figure 6-5: Schematic overview of block discarding or selfish mining.)

A keeps a private blockchain, apart from the public one on which honest miners and mining pools mine. *A* continues to mine on this private blockchain, keeping any blocks it has discovered private, without immediately publishing them. *A* determines its strategy according to the state of the network as described below.

State 0: This is the initial state in which the state of *A*'s private blockchain is the same as that of the public one. When α denotes *A*'s hashpower as a proportion of the total network hashpower, the following two developments can occur:

- With probability α, *A* discovers a block, the system transitioning to State 1 (in which *A*'s private blockchain is one block longer).
- With probability $1–\alpha$, another miner discovers a block and the public blockchain becomes one block longer, whereupon *A* replicates the public blockchain as its private blockchain.

State 1: This is a state in which *A*'s private blockchain is one block longer, and the following two developments can occur:

- With probability α, *A* discovers a block, the system transitioning to State 2 (in which *A*'s private blockchain is two blocks longer).
- With probability $1–\alpha$, another miner discovers a block, and the public blockchain reaches the same length as *A*'s private blockchain, the system transitioning to State 0'.

State 0': *A* immediately publishes its private blockchain. In this state, therefore, the public blockchain comprises two competing branches: the public branch and *A*'s forked branch, both consisting of one block. When γ denotes the percentage of the aggregate hashpower controlled by the nodes mining on *A*'s branch in the network's total hashpower, the following three developments can occur; and in any case the system goes back to State 0:

- With probability α, *A* discovers a block and publishes it. In consequence, the forked branch consisting of the two blocks *A* has discovered becomes the longer one in the public blockchain, resulting in *A*'s branch being adopted and *A* earning a reward for the two blocks.
- With probability $\gamma(1–\alpha)$, another miner discovers a block that is added to *A*'s branch, and said miner and *A* earn a reward for one block, respectively.
- With probability $(1–\gamma)(1–\alpha)$, another miner discovers a block that is added to the public branch, and said miner earns a reward for two blocks.

State 2: This is a state in which *A*'s private blockchain is two blocks longer, and the following two developments can occur:

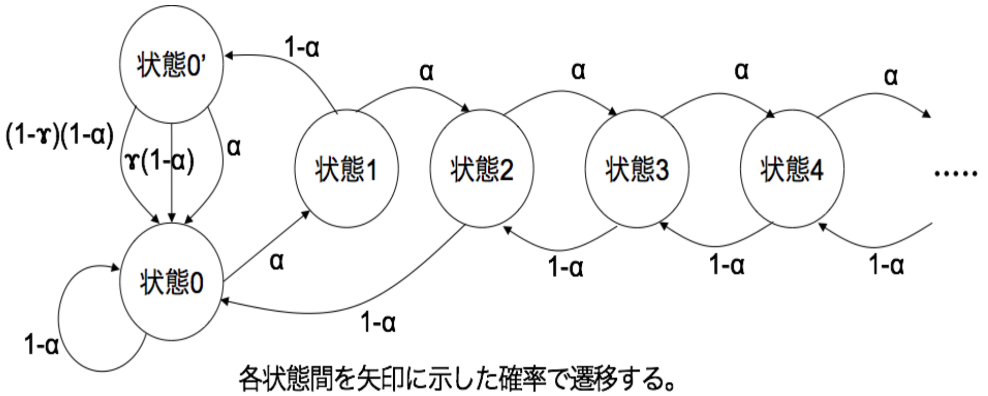
- With probability α, *A* discovers a block, the system transitioning to State 3 (in which *A*'s private blockchain is three blocks longer).
- With probability $1–\alpha$, another miner discovers a block; and when the public blockchain becomes one block shorter than *A*'s private blockchain, *A* publishes its private two blocks. As these two blocks published by *A* are adopted, *A* earns a reward for the two blocks, the system transitioning to State 0.

State *n* (n>2): This is a state in which *A*'s private blockchain is *n* blocks longer, and the following two developments can occur:

- With probability α, *A* discovers a block, the system transitioning to State n+1 (in which *A*'s private blockchain is three blocks longer). *A* earns a reward for one block.
- With probability 1–α, another miner discovers a block, the system transitioning to State n–1.

This attack exploits the Bitcoin network's vulnerability that allows a miner having found a legitimate block to keep it private without immediately publishing it.

Reference: V. Buterin, "Selfish Mining: A 25% Attack against the Bitcoin Network," *Bitcoin Magazine*, https://bitcoinmagazine.com/articles/selfish-mining-a-25-attack-against-the-bitcoin-network-1383578440.



各状態間を矢印に示した確率で遷移する。
α :Aのハッシュパワーの割合
γ：フォークが発生した場合、Aのブロックに対してマイニングを行うノードの全ハッシュパワーの割合

Figure 6-5: Schematic overview of block discarding or selfish mining

=============

図 6-5 の訳

1. State 0'
2. State 1
3. State 2
4. State 3
5. State 4
6. State 0
7. The system transitions from one state to another with the probability indicated near each arrow, where:
   α denotes the percentage of *A*'s hashpower; and
   γ denotes the percentage of the aggregate hashpower controlled by the nodes mining on *A*'s forked branch.

=============

6.1.6.2. Countermeasure

In the case of γ =1, *A* can earn a more than fair reward by employing the strategies above, regardless of the value of α. In the case of γ =0, on the other hand, for *A* to earn a more than fair reward, the value of α needs to be 1/3 or larger,[23] which makes the attack more difficult to carry out. However, γ =0 is hard to realize because, when the public blockchain is forked, honest miners are unable to recognize which branch is *A*'s one. Thus, as a feasible countermeasure it is proposed that the algorithm be changed in such a way that, when an honest miner learns of two competing branches, the miner propagates both of them and chooses which one to mine on uniformly at random. This would yield γ =1/2, meaning that in order for *A* to earn a more than fair reward, the value of γ must be 1/4 or larger. Given that what this countermeasure requires is a change in the mining algorithm only, it should be implemented with relative ease.

[23] I. Eyal and E. G. Sirer, *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*, Figure 3.

6.1.6.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 2 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Narrow ・・・ The potential targets are honest miners in a mining pool.
  - Severity: Low ・・・ As the potential targets are honest miners in a mining pool, ordinary users would not be affected.
- Impact on financial trading systems
  - Scope: Narrow ・・・ Given mining is successfully completed, the systems would not be affected.
  - Severity: Medium ・・・ A miner succeeding in mining could not earn a fair reward.
- Occurrence probability
  - Ease of attack: Low ・・・ The attacker must have mining capabilities.
  - Incentive: Medium ・・・ Rewards for successful mining would be limited to moderate amounts.

6.1.7. Block Withholding

6.1.7.1. Outline

This attack is aimed primarily at damaging a mining pool and unjustly obtaining proceeds by withholding a block the attacker has mined and discovered. The attack targets the operator and participants of a mining pool. Carrying out this attack requires the attacker (*A*) to be a participant in the mining pool. The greater *A*'s hashpower, the larger the impact of the attack.

This attack is categorized into two types based on the tactic used: sabotage and lie in wait.

In a sabotage attack, *A*'s tactic is simply to not submit any blocks it has mined to the operator of the mining pool. Consequently, when the mining pool adopts a pay-per-last-N-shares (PPLNS) reward system,[24] the operator does not suffer a loss but the rewards for the participants, including *A*, decrease by an amount corresponding to the percentage of *A*'s hashpower in the pool's total hashpower. In contrast, when the mining pool employs a pay-per-share reward system,[25] in which each participant is rewarded according to the participant's degree of contribution to mining even if no block is discovered, the operator suffers a loss.

The tactic of a lie in wait attack, on the other hand, allows *A* to gain proceeds. To carry out this attack, *A* first conducts mining concurrently in multiple mining pools that employ the PPLNS reward system. When discovering a block in one of these mining pools, *A* keeps it private; and, to concentrate its entire hashpower on that mining pool, *A* stops mining in the other pools. Then, *A* submits the block after a certain period of time from the discovery (*T*), increasing its reward by an amount corresponding to the portion of its hashpower that was previously allocated to the other mining pools. If *T* is excessively long, however, the probability of another participant in the mining pool finding a block first increases while the expected value of the reward to be earned by *A* decreases. The value of *T* that yields the maximum expected value of said reward can be obtained by the following formula: $T = (m–1)/(2m–1)T_0$, where T_0 represents the average time to find a block, and *m* represents the number of mining pools in which *A* conducts mining.[26]

This attack exploits the Bitcoin network's vulnerability that allows a miner having found a legitimate block to keep it private without immediately publishing it.

6.1.7.2. Countermeasure

As a countermeasure against this attack, it is proposed that the Bitcoin protocol be modified in such a way that any block discovered by a miner in a mining pool cannot be identified as a valid one until it is submitted to the operator of the mining pool (i.e., the miner cannot choose to keep the block private). The modification entails the following:

(1) Every block will have three additional fields: SecretSeed, ExtraHash, and SecretHash.

ExtraHash will be the hash of SecretSeed.

(2) ExtraHash will be a part of the block header and one of the fields used in calculating the block hash.

(3) SecretHash will be the hash of the concatenation of the block hash and SecretSeed.

(4) The requirement for a block to be valid will be changed from the current one that mandates the block hash be less than $2^{256}/(2^{32}D)$;[27] instead, the new requirement will dictate that the block hash be less than $2^{256}/2^{32}$ and SecretHash be less than $2^{256}/D$ (where *D* denotes the degree of difficulty).[28]

(5) The mining pool operator will choose SecretSeed and keep it secret, while calculating ExtraHash and providing it and the other fields to the miners. The miners will calculate the block hash to see if it is less than $2^{256}/2^{32}$ and, if that is the case, will submit it as a share without knowing whether that is a valid block. The mining pool operator, with the knowledge of SecretSeed, will calculate SecretHash and, if that is less than $2^{256}/D$, will broadcast it as a valid block to the network.

As this countermeasure necessitates a Bitcoin soft fork, it is considered relatively difficult to implement.

[24] A method under which a reward to a pool participant is determined based on the participant's degree of contribution to a certain computation amount prior to the discovery of the block.

[25] A method under which a reward to a miner is determined simply based on the miner's hashpower and amount of time spent on mining.

[26] M. Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*, Chapter 6.2.2.

[27] This is because a block hash is 256 bits long and is set in a way that, when D=1, enables one block to be mined per 2^{32} hash calculations.

[28] The frequency with which a mining pool mines a block remains the same before and after the change.

6.1.7.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 2 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Narrow ・・・ The potential targets are honest miners in a mining pool or its operator.
  - Severity: Low ・・・ As the potential targets are honest miners in a mining pool or its operator, ordinary users would not be affected.
- Impact on financial trading systems
  - Scope: Narrow ・・・ Given mining is successfully completed, the systems would not be affected.
  - Severity: Medium ・・・ A miner succeeding in mining could not earn a fair reward.
- Occurrence probability
  - Ease of attack: Low ・・・ The attacker must have mining capabilities.
  - Incentive: Medium ・・・ Rewards for successful mining would be limited.

6.1.8. Bribery Attack

6.1.8.1. Outline

This attack is aimed at acquiring a large volume of computational resources in a short period of time, and thereby succeeding in a double-spending or block withholding attack. The attack targets honest miners and vendors.

Carrying out this attack requires the attacker (*A*) to provide miners with bribes, which in turn necessitates having funds available for that purpose.

This attack begins with bribing miners in any of the three ways described below. Through these means, *A* acquires a large volume of computational resources in a short period of time, launching a double-spending or block withholding attack.

(1) Directly pay bribes in either cryptocurrency or fiat currency, which can easily be done by using a cloud mining provider.

(2) Establish a mining pool that pays above-market rewards in an attempt to lure miners.

(3) Offer bribes to miners to have them work on a blockchain's fork that *A* wants to extend. This can be achieved, for example, by simply broadcasting a high-fee transaction to the fork *A* wants to extend.

This attack exploits the consensus formation algorithm in Bitcoin transactions that is based on the majority of hashpower.

6.1.8.2. Countermeasure

Preventing this attack demands that the reward for mining a block be equal to or greater than the total amount transacted in the block, but that is impractical.

In lieu of countermeasures, the following factors that may mitigate the risk of this attack are suggested:

- Even if miners are offered bribes or higher rewards, they may be unable or unwilling to rent their computational resources or change pools, or may be unaware of a blockchain's fork that contains a bribe. On the other hand, as miners behave in a more economically rational manner and become more technically capable, they are more likely to act in a way beneficial to them; in that case, the aforementioned possibility is less likely to be true (meaning that the chance of miners accepting bribes increases).
- To profit from this attack, *A* must create a very large transaction and have funds available for that purpose. In the event of the attack failing, *A* does not necessarily lose the value of that transaction but cannot recover the bribes paid.
- In the case of *A* trying to obtaining a product through double spending, the shipment thereof may be reversed. Also, a double-spending transaction may require *A* to pay transaction fees[29] to the counterparty, or may not provide sufficient proceeds to make the relative cost of bribes negligible.
- Recipients are presumed to demand a greater number of confirmation blocks for larger transactions, increasing *A*'s bribery costs. However, it is also possible for *A* to create many smaller transactions and attempt to double-spend all of them. Hence, this factor is not believed to substantially mitigate the risk of the bribery attack.
- The target may try to counteract the attack by bribing miners (the situation escalating into a bribe war).
- If *A* intends to carry out this attack, the intention is promptly known to other miners. And, miners are aware that, if successful, the attack would reduce the value of Bitcoin, which means that accepting bribes might be profitable in the short term but detrimental in the long term. Hence the possibility for them to accept bribes is low.

[29] These are not rewards to miners, but general transaction fees outside the Bitcoin network.

6.1.8.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 4 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Medium ･･･ The potential targets are miners and vendors.
  - Severity: Medium ･･･ The damage would be similar to that caused by a double-spending (outlined in Sections 2.4.1 to 2.4.3) or block withholding attack.
- Impact on financial trading systems
  - Scope: Medium ･･･ Double spending might cause a fork.
  - Severity: Medium ･･･ Double spending might cause a fork.
- Occurrence probability
  - Ease of attack: Low ･･･ The attacker must have financial resources.
  - Incentive: Medium ･･･ The incentive would be similar to that for a double-spending (outlined in Sections 2.4.1 to 2.4.3) or block withholding attack.

6.1.9. Refund Attack

6.1.9.1. Outline

This attack is aimed at gaining illicit proceeds by taking advantage of refund policies and denying involvement in the transaction. The attack is thought to be further categorized into two types: a Silkroad attack that highlights an authentication vulnerability in Bitcoin Improvement Proposal (BIP) 70; and a Marketplace Trader attack that misuses the refund policies of existing payment processors.[30]

There is no particular prerequisite for the refund attack, each type of which is conducted following the steps below.

6.1.9.1.1 Silkroad Attack (See Figure 6-6: Schematic overview of Silkroad attack.)

A typical situation is that a customer who wishes to purchase an illicit product from a trader (*T*) carries out this attack to deny its involvement in the transaction, taking advantage of a vendor (*V*) that is a third party. Here, the customer is the attacker (*A*) while the target is *V*.

(1) *A* downloads from *T*'s website a payment request message that contains *T*'s address (A_T), price (*B*), and *T*'s public key (σ_T).

(2) *A* searches for *V* that sells a product at a price equal to or higher than that of the illicit one. Once *V* is found, *A* starts the payment process for *V*'s product, downloading a payment request message that contains *V*'s address (A_V), price (*B*), and *V*'s public key (σ_V).

(3) *A*'s wallet acknowledges the payment transaction and inserts A_T in the payment request message as the refund address.

(4) Upon receipt of the payment acknowledgement message from *V*, *A* cancels the order and requests a refund from *V*.

(5) If *V* complies with BIP 70, the refunded coins are sent to *T*.[31]

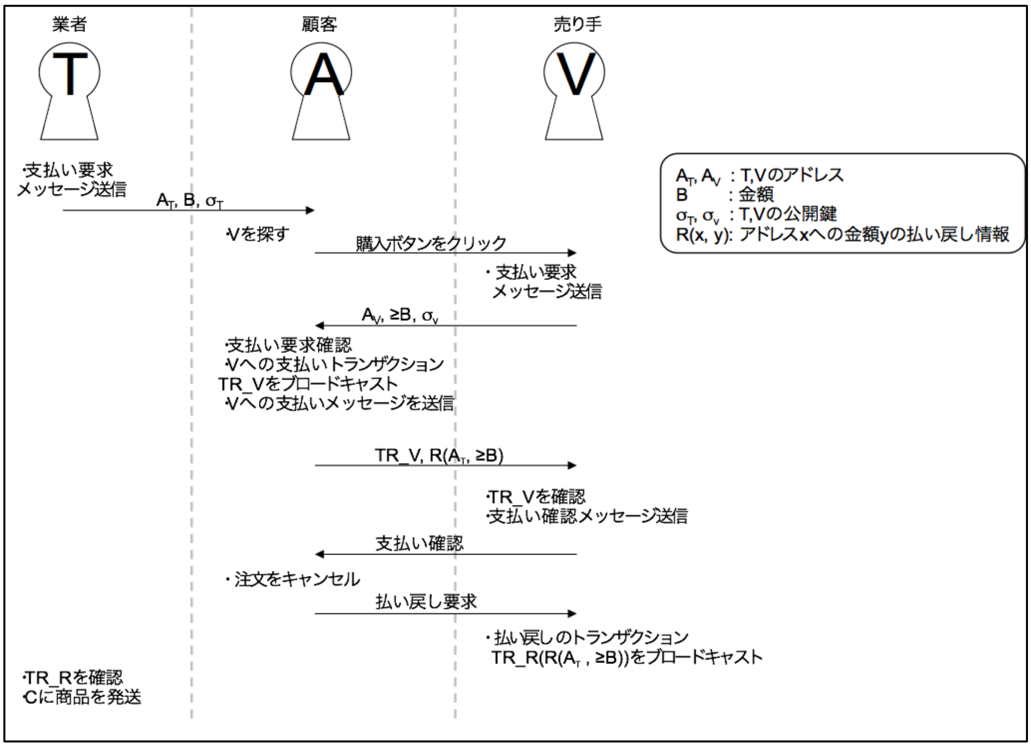


Figure 6-6: Schematic overview of Silkroad attack

=============

図 6-6 の訳

9. Trader
10. Customer
11. Vendor

[30] P. McCorry, S. F. Shahandashti, and F. Hao, *Refund Attacks on Bitcoin's Payment Protocol*.

[31] The reason is that, under the payment protocol introduced by BIP 70, while a customer informs the vendor of the Bitcoin address for refund purposes at the time of payment, the vendor has no way of confirming whether that address actually belongs to the customer.

12. Send payment request message.
13. Search for *V*.
14. Click Purchase button.
15. A_T, A_V: Addresses of *T* and *V*
    B: Price
    σ_V, σ_T: Public keys of *T* and *V*
    R(x, y): Information on refund of *y* amount to *x* address
16. Send payment request message.
17. Acknowledge payment request.
    Broadcast payment transaction to *V* (*TR_V*).
    Send payment message to *V*.
18. Acknowledge *TR_V*.
    Send payment acknowledgement message.
19. Acknowledge payment.
20. Cancel order.
21. Request refund.
22. Broadcast refund transaction *TR_R*($R(A_T, \geq B)$).
23. Acknowledge *TR_R*.
    Ship product to *A*.

=============

6.1.9.1.2 Marketplace Trader Attack (See Figure 6-7: Schematic overview of Marketplace Trader attack.)

Under the Marketplace Trader attack, a rogue merchant that is the attacker creates a website on which the latest products and other merchandise are sold at below-market prices, and advertises that payments can be made through a reliable major retailer to reassure potential customers. Once a customer is tricked into purchasing an item on the website, *A* cancels the order and gains proceeds by having the retailer refund to *A* the Bitcoins paid by the customer. In this attack, the merchant is the attacker (*A*) while the target is the customer (*V*).

(1) *A* creates a website on which products are sold below the market prices, and misrepresents itself as a trustworthy merchant by conducting transactions through a reliable retailer (*T*) like CeX.

(2) Once *V* decides to buy *A*'s product and clicks the Purchase button on *A*'s website, *A* automatically obtains from *T*'s website a payment request message that contains *T*'s address (A_T), price (*B*), and *T*'s public key (σ_T), forwarding it to *V*.

(3) *V*'s wallet opens the genuine payment request message that displays the name of *T*, a reliable retailer, along with the amount to be paid. This leads *V* to trust *A*, send the payment message to *T*, and broadcast the payment transaction to *T* (*TR_T*).

(4) When *A*'s website detects *TR_T* on the network, *A* refreshes *V*'s web browser to display a fake confirmation page.

(5) *A* informs *T* of the cancellation of *V*'s order, and sends *T* an email that contains the refund address (A_A) and price (*B*).

(6) In accordance with the policy that permits authentication by email, coins worth *B* are sent to A_T.

The Marketplace Trader attack exploits the fact that a refund of a Bitcoin transaction can be received by a third party that misrepresents itself as the purchaser of the product.

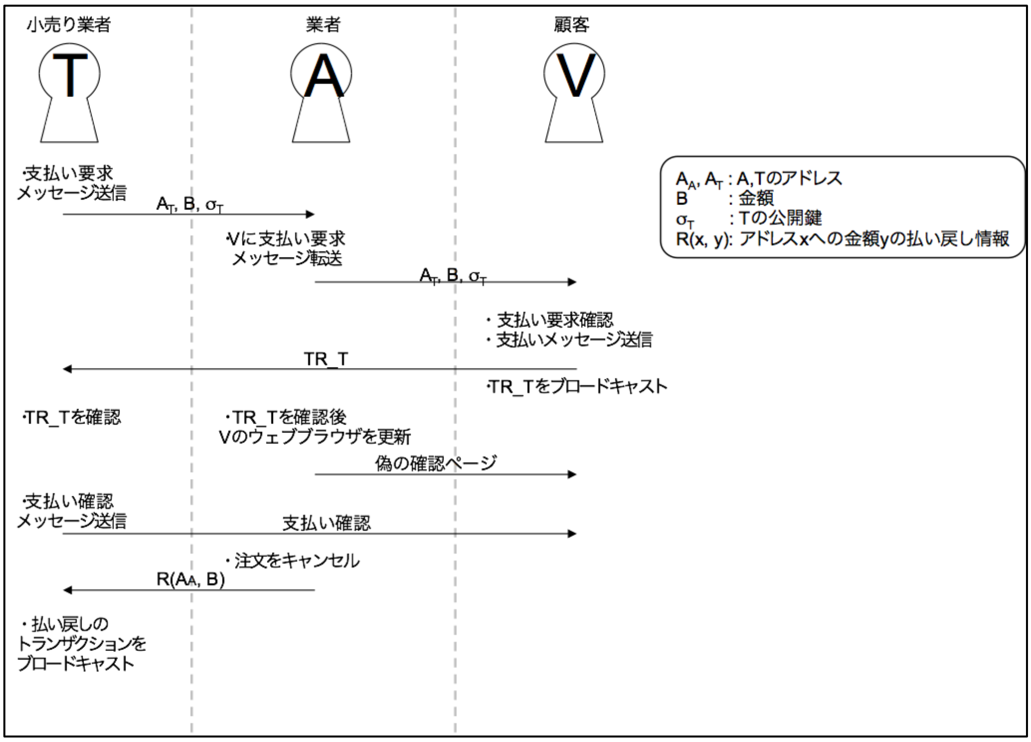


Figure 6-7: Schematic overview of Marketplace Trader attack

=============

図 6-7 の訳

1. Retailer
2. Merchant
3. Customer
4. Send payment request message.
5. Forward payment request message to *V*.
6. Acknowledge payment request.
   Send payment message.
7. Broadcast *TR_T*.
8. Detect *TR_T*.
9. Upon detection of *TR_T*, refresh *V*'s web browser.
10. Display fake confirmation page.
11. Send payment acknowledgement message.
12. Acknowledge payment.
13. Cancel order.
14. Broadcast refund transaction.
15. A_A, A_T: Addresses of *A* and *T*
    B: Price
    σ_T: *T*'s public key
    R(x, y): Information on refund of *y* amount to *x* address

=============

6.1.9.2. Countermeasure

What is proposed as a countermeasure against the refund attack is to provide a seller with publicly verifiable evidence that can cryptographically prove the refund address received was endorsed by the very purchaser who authorized the payment. The idea is to prevent the attack by having the purchaser endorse its refund address using the key that authenticated the transaction.

This countermeasure is considered relatively easy to implement, and has actually been introduced by some exchanges.

6.1.9.3. Assessment

The risk of the refund attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 6 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 2 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted by this attack.
  - Severity: Medium ・・・ The financial damage would be limited.
- Impact on financial trading systems
  - Scope: Narrow ・・・ There would be no impact on systems.
  - Severity: Low ・・・ There would be no impact on systems.
- Occurrence probability
  - Ease of attack: Low ・・・ This attack exploits a specific protocol that is BIP 70.
  - Incentive: Medium ・・・ Proceeds would be limited.

6.1.10. Punitive and Feather Forking

6.1.10.1. Outline

This attack is aimed, for example, at getting the target's transactions blacklisted to have its assets frozen. The attack targets customers, vendors, and other ordinary users.

Carrying out this attack requires the attacker (*A*) to have mining capabilities. Additionally, the greater *A*'s hashpower, the larger the impact of the attack.

Suppose that *A* wants a transaction from the target to be blacklisted. In that case, *A* carries out this attack by creating a fork of the blockchain that contains the transaction, mining on this fork as opposed to mining on the main blockchain, and announcing its intent to take these actions. If *A* has only small hashpower and carries out this attack by itself, the probability for the attack to succeed is low. However, if *A* has certain hashpower and announces the intent to carry out this attack, other miners become aware of the greater chance that they may not be able to obtain rewards from mining on the main blockchain. This increases the number of miners working on the fork created by *A*, thereby raising the probability of the success of this attack.

This attack exploits the consensus formation algorithm in Bitcoin transactions that is based on the majority of hashpower.

6.1.10.2. Countermeasure

No measures to counter this attack are proposed.

6.1.10.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 1 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ･･･ Anyone could be targeted by this attack.
  - Severity: High ･･･ Cryptocurrencies owned by users would become permanently unusable.
- Impact on financial trading systems
  - Scope: Narrow ･･･ The systems would be sustained.
  - Severity: Low ･･･ The systems would be sustained.
- Occurrence probability
  - Ease of attack: Low ･･･ The attacker must have mining capabilities.
  - Incentive: Low ･･･ The attacker would not gain any direct proceeds.

6.1.11. Wallet Theft

6.1.11.1. Outline

This attack is aimed at illicitly acquiring cryptocurrencies in possession of users by hacking into their wallets. As such, the attack targets cryptocurrency users.

The attacker (*A*) needs to somehow obtain the target user's private key or the login ID and password for the user's web or other wallet.

Wallets are roughly divided into hot wallets and cold wallets. Hot wallets, which are online, include web wallets, desktop wallets, and mobile wallets. Examples of cold wallets, which are off-line, are paper wallets that are primarily pieces of paper on which private keys and addresses are printed for storage purposes, as well as hardware wallets that preserve private keys and other information in the form of special USB devices.

The way *A* obtains a user's private key and address data from the cold wallet is virtually limited to a physical or digital theft.

As for web wallets, however, users log on to these wallets by entering their ID and password while their private keys are managed by cryptocurrency exchange and other service providers. This provides *A* with the possibility to obtain their IDs and passwords by means of hacking, virus infection, and other malicious methods. Compared to web wallets, desktop and mobile wallets are securer in that the users manage their private keys using their computers or mobile devices; still it is not impossible for *A* to obtain those private keys by means of hacking, virus infection, and other malicious methods.[32] In addition, it is possible to obtain those keys by carrying out a collision attack on a hash function.

The vulnerability this attack exploits lies in not the Bitcoin system but computer security and cryptographic techniques in general.

6.1.11.2. Countermeasures

To counter this attack, the following measures are proposed:

The US government has launched its own Bitcoin networks with multifactor security that incorporates fingerprint biometrics for wallet protection. (See "Biometric Tech Secures Bitcoin Wallet," *Biometric Technology Today*, Vol. 2015, Issue 6, 2015.)

As another means of protecting Bitcoin private keys, the BlueWallet hardware token is proposed for use in authenticating transactions. Communicating with devices that create transactions via Bluetooth Low Energy,[33] this token displays the details of each transaction (the payee's address, amount to be paid, and fee) before the user signing it, allowing the user to check whether the transaction is authentic. The private key stored in a BlueWallet is unlocked only when the user enters the correct PIN. (See T. Bamert, C. Decker, R. Wattenhofer, and S. Welten, *BlueWallet: The Secure Bitcoin Wallet*, Springer International Publishing, 2014, pp. 65–80.)

The Bitcoin system offers a multi-signature scheme that assigns multiple private keys to one address. Under this scheme, the theft of one of the necessary private keys alone would not enable the coins to be stolen unless the rest of the necessary keys are stolen as well.

These countermeasures are considered relatively easy to implement, and have actually been introduced by some entities.

[32] In August 2016, the Hong Kong-based cryptocurrency exchange Bitfinex was hacked and stolen about 120,000 bitcoins (worth approximately 6.6 billion yen at the time) in spite of adopting a highly secure multi-signature scheme that requires multiple private keys for authentication.

[33] A power-conserving variant of the Bluetooth wireless personal area network technology, added when Bluetooth 4.0 was introduced.

6.1.11.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 9 (High)
- Assessment score (Impact on financial trading systems × Occurrence probability): 3 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted by this attack.
  - Severity: High ・・・ The assets of individual users could be wiped out.
- Impact on financial trading systems
  - Scope: Narrow ・・・ There would be no impact on systems.
  - Severity: Low ・・・ There would be no impact on systems.
- Occurrence probability
  - Ease of attack: Medium ・・・ This depends on the level of security of each wallet.
  - Incentive: High ・・・ The attacker could access the target’s assets at will.

6.1.12. Transaction Malleability

6.1.12.1. Outline

This attack is aimed at gaining illicit proceeds or disrupting the network by having users or exchanges repeatedly send the same transaction. It exploits the vulnerability that renders hashes—by which each transaction is identified, as explained later—malleable.

The attack targets those ordinary users and exchanges that identify transactions based solely on hashes without checking the contents of the transactions.

The attacker (*A*) does not need to know the private key of its target (*V*), nor does it need to meet particular conditions to modify transactions sent from the target. Hence, there is no special requirement for *A*.

This attack is conducted following the steps below. (See Figure 6-8: Schematic overview of transaction malleability.)

(1) *A* has *V* (normally, exchanges and similar entities) create a payment transaction to *B* (*TR_VB*). If *A*'s purpose is to gain proceeds, *B* needs to be an address under *A*'s control; but that is not the case if *A*'s purpose is simply to disrupt the network.

(2) *A* waits for *TR_VB* created by *V* to be sent to the network.

(3) Upon receipt of *TR_VB*, *A* reproduces and falsifies it to create another transaction *TR_VB'* that has a different hash value while maintaining the validity of the digital signature. There are multiple ways to create *TR_VB'*. One is to create a new, different signature if *A* knows *V*'s private key. Even without that knowledge, *A* still can create *TR_VB'*, for example by modifying a portion of the script of *TR_VB* only to the extent that the script's outcome remains unchanged.

(4) *A* sends *TR_VB'* to the network.

(5) If *TR_VB'* is accepted, the attack succeeds. In cases where *V* identifies transactions based solely on hashes, the fact that the hash of *TR_VB* is not contained in the blockchain misleads *V* into thinking that the payment to *B* failed and makes it resend the same transaction.

This attack exploits two of Bitcoin's vulnerabilities: one is the possibility to create a transaction that is semantically identical (that is, in terms of the payer, payee, and amount) to the original transaction but has a different hash value by changing the way it is scripted; the other lies in Bitcoin software that identifies a transaction created in the aforementioned manner as a separate transaction.

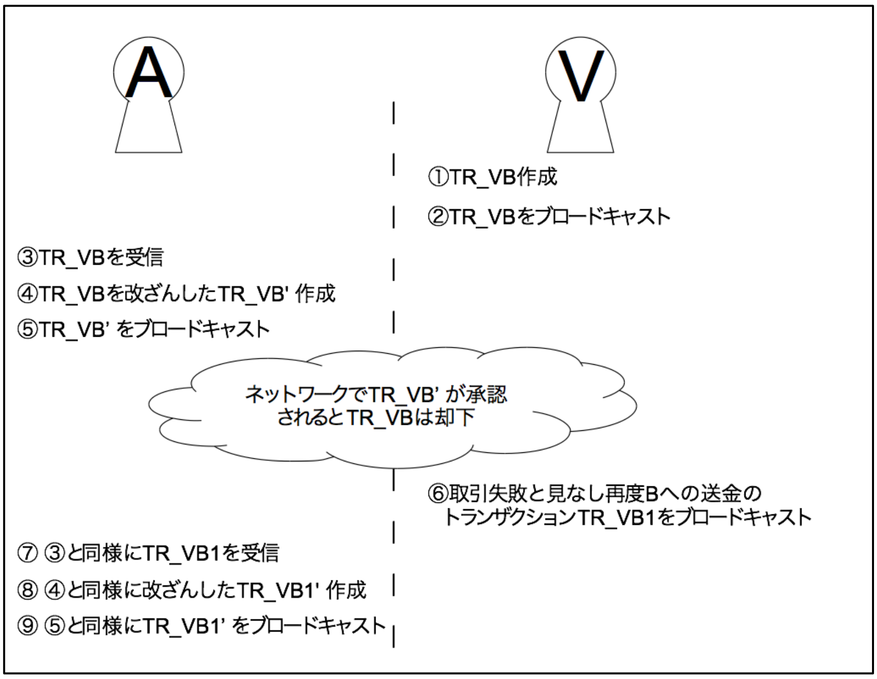

A
V
①TR_VB作成
②TR_VBをブロードキャスト
③TR_VBを受信
④TR_VBを改ざんしたTR_VB' 作成
⑤TR_VB' をブロードキャスト
ネットワークでTR_VB' が承認
されるとTR_VBは却下
⑥取引失敗と見なし再度Bへの送金の
トランザクションTR_VB1をブロードキャスト
⑦ ③と同様にTR_VB1を受信
⑧ ④と同様に改ざんしたTR_VB1' 作成
⑨ ⑤と同様にTR_VB1' をブロードキャスト

Figure 6-8: Schematic overview of transaction malleability

=============

図 6-8 の訳

1. (1) Create *TR_VB*
   (2) Broadcast *TR_VB*
2. (3) Receive *TR_VB*
   (4) Create *TR_VB'* by falsifying *TR_VB*
   (5) Broadcast *TR_VB'*
3. *TR_VB'* is accepted by the network, causing *TR_VB* to be rejected
4. (6) Regard it as a failed transaction, rebroadcasting a payment transaction to *B* (*TR_VB1*)
5. (7) Receive *TR_VB1* similarly to (3) above
   (8) Create *TR_VB1'* by falsifying *TR_VB1*
   (9) Broadcast *TR_VB1'* similarly to (5) above

=============

6.1.12.2. Countermeasures

To counter this attack, the following measures are proposed:

- Restrict the syntax of the Bitcoin scripting language so that different scripts will not produce the same outcome. This proposal, however, is an ad-hoc, heuristic one, not subjected to any formal argument. As such, it is unclear whether this measure is effective to all kinds of syntax.
- Change the way to calculate hash values, from the current method whereby a hash value is calculated based on the entire transaction, to the method that calculates a hash value based on the portion of the transaction that excludes the input script.
- Introduce a new instruction code "OP_CHECKLOCKTIMEVERIFY" to the Bitcoin scripting language, which renders a transaction output unspendable until some point in the future, making it impossible for *A* to immediately withdraw coins. This measure was already introduced in October 2015.
- In August 2017, SegWit was activated in the Bitcoin network.[34] In consequence, the conventional method, whereby a hash value is calculated based on the entire transaction, has been superseded by the method that calculates a hash value based on the input and output data excluding the digital signature. This precludes falsifying a transaction to create another that has a different hash value while maintaining the validity of the digital signature, as described in Step 3 above. As a result of this measure coupled with the introduction of OP_CHECKLOCKTIMEVERIFY mentioned above, no transaction malleability is expected to occur in the future.

6.1.12.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 6 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 3 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Medium ・・・ The potential targets are ordinary users and exchanges that identify transactions based solely on hashes.
  - Severity: Medium ・・・ The financial damage would be limited.
- Impact on financial trading systems
  - Scope: Narrow ・・・ There would be no impact on systems.
  - Severity: Low ・・・ The number of the potentially targeted exchanges is small.
- Occurrence probability
  - Ease of attack: High ・・・ What is needed is just to modify transactions.
  - Incentive: Medium ・・・ Proceeds would be limited.

[34] Thereafter, SegWit2x was proposed to increase the size of each block in the blockchain network from one megabyte to two megabytes, but its implementation was called off, due in part to SegWit2x requiring a hard fork and having no protection against replay attacks (a form of attack whereby, following a hard fork, a transaction on the original blockchain is also executed on the forked blockchain without the user's intent in the event of the latter not employing a signature specific thereto). In contrast, SegWit mentioned here is implemented via a soft fork as opposed to a hard fork, thus entailing no replay attack risk.

6.1.13. Time Jacking

6.1.13.1. Outline

This attack is aimed primarily at successfully making double-spending transactions, draining another miner's computational resources, or slowing down the speed of transactions being accepted, by illicitly altering a blockchain node's network time counter. The attack targets ordinary users and miners.

The attacker (*A*) needs to know the IP address of its target (*V*).

Every node on the Bitcoin network internally maintains a counter that represents the network time. When a node connects to the network for the first time, the median time of its peers is sent in the version message. If that median time deviates from the system time by more than 70 minutes, however, the network time counter reverts to the system time.

The network time is used to verify new blocks. Any block whose timestamp is ahead of the current network time by more than 120 minutes is rejected by nodes. A block whose timestamp is earlier than the median time of the past 11 blocks is also rejected. This verification puts upper and lower bounds on the acceptable range of block timestamps.

This attack is conducted following the steps below.[35] (See Figure 6-9: Schematic overview of time jacking.)

(1) *A* has multiple nodes connect to the network and report inaccurate timestamps. This makes *V*'s timestamp behind of the real time by a maximum of 70 minutes and a majority of the other nodes ahead by a maximum of 70 minutes. Even if the number of nodes under *A*'s control is small, *A* is still able to send enough version messages to alter the clocks of the connected peers by falsifying the sender's IP addresses using Tor and various other methods.

(2) *A* creates a block (*B*) whose timestamp is set 190 minutes ahead of the real time.

(3) *V*'s node rejects *B* because *B*'s timestamp appears to be 260 minutes ahead of *V*'s network time.

(4) On the other hand, a majority of the other nodes, to which *B*'s timestamp appears to be 120 minutes ahead, accept *B*.

(5) *V*'s node becomes isolated from the network's normal transaction processing as it rejects all blocks created by the other miners due to their timestamps appearing to be 140 minutes ahead of *V*'s network time. This makes it easier to conduct a double-spending transaction against *V*,[36] drain *V*'s computational resources,[37] and carry out other attacks.

As described above, every Bitcoin node's network time is determined based on the timestamp of the version message sent from its peers upon connection to the network. This makes it possible to speed up or slow down a particular node's network time counter by sending a fraudulent version message. That is the vulnerability the time jacking attack exploits.

[35] corbixgwelt, "Timejacking & Bitcoin," http://culubas.blogspot.jp/2011/05/timejacking-bitcoin_802.html.

[36] The reason is that, even if the double-spending transaction has been accepted by a majority of the other miners, *V* does not notice the transaction as it has rejected that block.

[37] From the perspective of *V* that is being attacked, the blocks accepted by a majority of the other nodes on the network appear to be invalid. Thus, *V* is tricked into mining on a block to be added in front of those blocks.

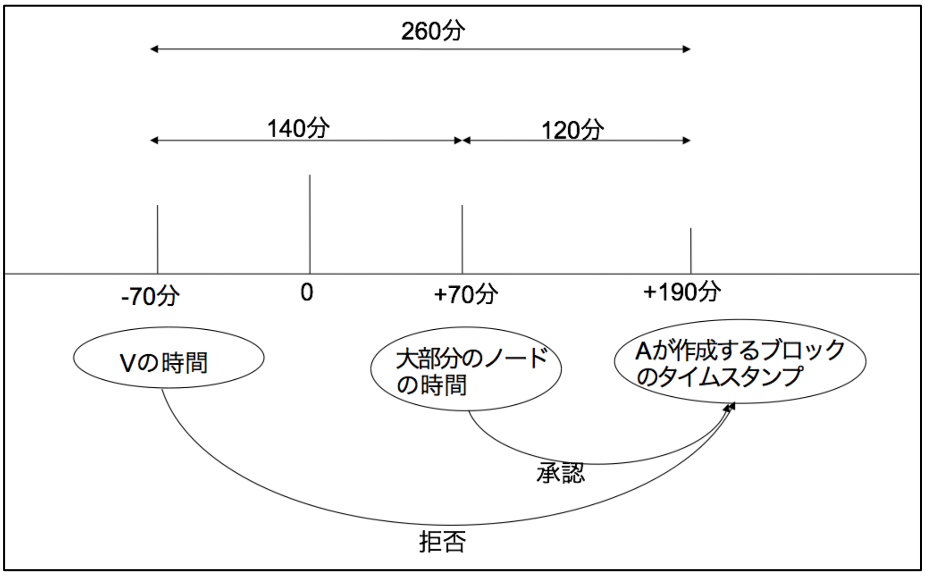


Figure 6-9: Schematic overview of time jacking

=============

図 6-9 の訳

1. 260 minutes
2. 140 minutes
3. 120 minutes
4. 70 minutes behind
5. 70 minutes ahead
6. 190 minutes ahead
7. *V*'s network time
8. Most nodes' network time
9. Timestamps of a block *A* creates
10. Accept
11. Reject

=============

6.1.13.2. Countermeasures

To counter this attack, the following measures are proposed:

- Use the node's system time instead of the network time, to determine block timestamps and the upper limit thereof. This, however, requires the clock to be calibrated on a regular basis; a disagreement among nodes of only a few seconds or so could allow *A* to split the network or isolate a node, for instance.
- Tighten the acceptable range of deviation of the node's network time. The range can be reduced from the current 70 minutes to 30 minutes, which would not completely prevent this attack though.
- Use trusted peer nodes only. This measure, however, could make those nodes less secure since a small number of trusted peers may be subverted. Furthermore, this would negate the benefit of decentralized systems.
- Monitor network health and shut down the network if there is a suspicious activity. This is an effective measure but would not resolve this attack automatically.
- Require more confirmations before accepting a transaction. This could prevent double spending, but other goals of this attack, such as draining miners' computational resources, might still be attained.
- Use the value computed based on the median time of past blocks—not only for the lower limit of block timestamps but also for the upper limit thereof, instead of 120 minutes—when verifying blocks. This is a complete solution to this attack, and is considered relatively easy to implement, despite requiring revisions to the Bitcoin client rules and software upgrades.
- Retain blocks with excessive timestamps in memory and recheck them later. This, too, requires revisions to the Bitcoin client rules and software upgrades, but is considered relatively easy to implement.

6.1.13.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 6 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Narrow ・・・ The potential targets are miners.
  - Severity: Low ・・・ The potential targets are miners.
- Impact on financial trading systems
  - Scope: Medium ・・・ Clocks within the systems would become asynchronous.
  - Severity: Medium ・・・ Clocks within the systems would become asynchronous.
- Occurrence probability
  - Ease of attack: High ・・・ What is needed is just to create blocks with inaccurate timestamps.
  - Incentive: Medium ・・・ Proceeds would be limited.

6.1.14. Sybil

6.1.14.1. Outline

This attack is aimed at succeeding in time jacking, DDoS, and double-spending attacks by creating and using multiple identities. It targets ordinary users and miners.

The attacker (*A*) is required to install many nodes in the network, which in turn necessitates many clients or helpers. By installing helper nodes in the network or creating identities, *A* aims to isolate its target from the network, disconnect transactions initiated by the target, and force the target to choose only those blocks governed by *A*.

This attack exploits Bitcoin's vulnerability that allows a single individual to create multiple identities and collude with helper nodes.

6.1.14.2. Countermeasure

A protocol proposed to counter this attack is Xim, a mixing protocol for two anonymous nodes to exchange coins. This protocol dictates that, when paring nodes, the miner receive a fee from the both nodes. Accordingly, creating numerous identities would incur *A* costs for paring nodes, which is expected to deter *A* from perpetrating this attack.

6.1.14.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 6 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 6 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ･･･ In a worst case scenario, anyone could be targeted by this attack.
  - Severity: Medium ･･･ The financial damage would be limited although that depends on the purpose of the attack.
- Impact on financial trading systems
  - Scope: Broad ･･･ In a worst case scenario, a DDoS attack might disable the systems.
  - Severity: Medium ･･･ In a worst case scenario, a DDoS attack might disable the systems.
- Occurrence probability
  - Ease of attack: Medium ･･･ Succeeding in this attack requires substantial resources.
  - Incentive: Medium ･･･ Proceeds would be limited.

6.1.15. DDoS

6.1.15.1. Outline

This attack is aimed at disabling the network by flooding it with a vast amount of invalid data. It targets the entire network ranging from exchanges and mining pools to e-wallets.

The attacker (*A*) must have a large number of nodes to broadcast a vast amount of data in a short period of time. By broadcasting a vast amount of invalid data like fake blocks and transactions from numerous clients, *A* aims to exhaust the network resources, preclude users from accessing the network, and have miners reject blocks from ordinary users.

This attack exploits the Bitcoin network's vulnerability that allows anyone to send invalid data without incurring much cost.

6.1.15.2. Countermeasure

What is proposed is the use of a proof-of-activity (PoA) algorithm that is robust against this attack.[38] Under the PoA algorithm, *N* number of stakeholders create a block. As a first step, a miner generates an empty block header in accordance with the proof-of-work protocol, and invokes a subroutine called "follow-the-satoshi," whereby stakeholders are selected randomly based on the amount of coins held. *N–1* number of miners so selected sign the hash of this empty block header and broadcast it. Lastly, the *N*th stakeholder adds transactions to this empty block header and signs the entire block. The PoA algorithm makes it easy to verify that the block contains the signatures of *N* number of stakeholders and to reject receiving invalid blocks, thereby mitigating the risk of the DDoS attack. However, as it is impossible to reject receiving invalid transactions, the PoA algorithm is not a perfect countermeasure.

6.1.15.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 6 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 9 (High)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Medium ･･･ The attack would impact the users connecting to nodes.
  - Severity: Low ･･･ The systems would be rendered unusable, but assets would not be lost.
- Impact on financial trading systems
  - Scope: Broad ･･･ The systems might be disabled.
  - Severity: High ･･･ The systems might be disabled.
- Occurrence probability
  - Ease of attack: High ･･･ The attack could be carried out with necessary tools.
  - Incentive: Medium ･･･ Proceeds would be limited.

[38] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [extended abstract]," *ACM SIGMETRICS Performance Evaluation Review*, Vol. 42, Issue 3 (December 2014), pp. 34–37.

6.1.16. Eclipse (or Netsplit) Attack

6.1.16.1. Outline

This attack is aimed at isolating the target from the network by monopolizing all connections to and from it. The attack targets honest miners and users.

As the attacker (*A*) is required to have many IP addresses, it must have botnets or a large number of nodes. In addition, *A* needs to possess the capability to have the target (*V*)'s blockchain client restart via DDoS and other attacks. *A* without this capability must wait until *V*'s blockchain client restarts to execute this attack. As such, the goal of this attack is to control *V*'s data flow and make *V* accept the data of *A*'s choosing only by controlling *V*'s routing.

This attack is conducted following the steps below. (See Figure 6-10: Schematic overview of eclipse [or netsplit] attack.)

(1) *A* has the IP addresses of the nodes under its control inserted into *V*'s "tried" table (which stores the IP addresses for peers to whom the node has successfully established a connection). Upon establishment of connection with a peer, the Bitcoin node automatically stores in its tried table the peer's IP address. Thus, all *A* has to do is connect to *V* from nodes *A* controls.

(2) *A* overwrites the IP addresses in *V*'s "new" table (which stores the IP addresses for peers to whom the node has not yet initiated a successful connection) with IP addresses not existing in the Bitcoin network. A Bitcoin node unconditionally inserts into its new table up to 1,000 IP addresses contained in ADDR messages sent from other nodes. *A* thus can complete this step by sending an ADDR message containing nonexistent IP addresses to *V* to which *A* has established a connection in Step 1 above.

(3) *A* makes *V*'s client restart in some way or waits until it restarts.

(4) Upon restart, *V*'s client selects IP addresses to which it is connecting, from its tried and new tables. Given that all the IP addresses stored in the new table are nonexistent, the IP addresses to which *V* are connecting are limited to the ones of the nodes *A* controls that are stored in the tried table.

(5) *A* also monopolizes all inbound connections to *V*'s node.

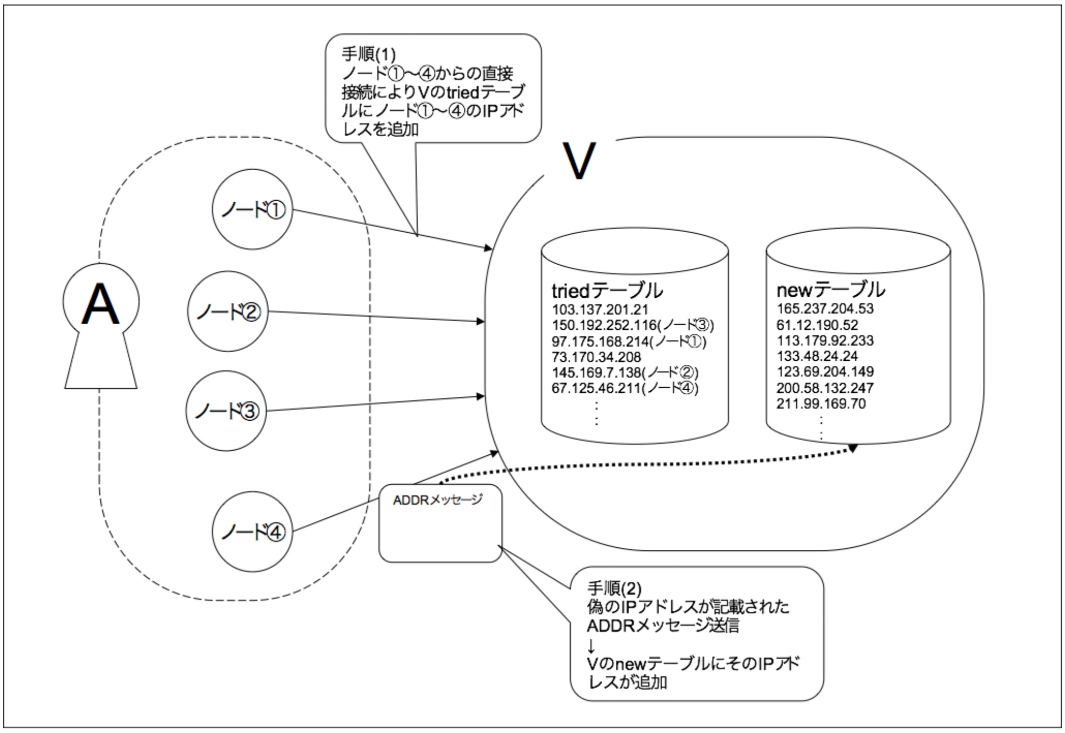


Figure 6-10: Schematic overview of eclipse (or netsplit) attack

=============

図 6-10 の訳

1. Step 1:
   Nodes 1 to 4 directly connect to *V*, thereby inserting their IP addresses into *V*'s tried table
2. Tried table
   103.137.201.21
   150.192.252.116 (Node 3)
   97.175.168.214 (Node 1)
   73.170.34.208
   145.169.7.138 (Node 2)
   67.125.46.211 (Node 4)
3. New table
4. ADDR message
5. Step2:
   An ADDR message containing nonexistent IP addresses is sent
   ↓
   These IP addresses are inserted into *V*'s new table
6. Node 1
   Node 2
   Node 3
   Node 4

============

6.1.16.2. Countermeasures

To counter this attack, the below-listed measures are proposed. Of these, the ones outlined in (a), (b), and (f) have been incorporated into the Bitcoin network.

(a) The current algorithm dictates that, if the tried table of a node just connected to by another has no more vacant slot, one of the IP addresses stored in the tried table be chosen randomly and replaced with the connecting node's IP address. Therefore, making repeated connections to *V* from the same IP address enables *A* to fill *V*'s tried table with *A*'s IP address. To prevent this, change the algorithm to ensure that a slot in which the connecting node's IP address is stored within the connected node's tried table is determined not randomly but based solely on the hash of that IP address.

(b) The current algorithm features a heavy bias toward forming outbound connections to IP addresses with fresh timestamps in the tried table. Replace this with the method of selecting IP addresses randomly from the tried and new tables.

(c) Before overwriting an older IP address stored in the tried table with new one, first attempt to connect to the older IP address, and store the new IP address only if the connection fails.

(d) Test a connection to a random IP address in the new table. If the connection succeeds, move the IP address to the tried table; otherwise, delete it from the new table.

(e) Add an "anchor" table that records IP addresses of current connections and the time of first connection to each address. Upon restart, connect to the oldest two IP addresses recorded in the anchor table.

(f) Increase the size of a tried table, which should be one of the most effective countermeasures. This would require *A* to have a greater number of nodes to increase the fraction of the IP addresses of nodes under its control in *V*'s tried table. However, if there are not many valid IP addresses in the tried table in the first place, this countermeasure would not be able to prevent *A* from increasing the fraction of the IP addresses of nodes under its control in the tried table. Thus, this countermeasure should be accompanied by another countermeasure that increases the number of valid IP addresses in the tried table.

(g) Raise the number of a node's outbound connections so as to increase the number of botnets and nodes *A* must have in place.

(h) Reject receiving an ADDR message that contains more than 10 IP addresses, and solicit ADDR messages only from outbound connections when the node's new table is too empty.

(i) Restrict the number of inbound connections from the same IP address, given that a Bitcoin node can currently have all of its inbound connections come from the same IP address, making it easy for *A* to monopolize *V*'s inbound connections.

(j) Introduce a system that detects anomalies, such as a flurry of short-lived inbound TCP connections from diverse IP addresses, large ADDR messages containing invalid IP addresses, and a spike in the number of inbound connections to a Bitcoin network node.

6.1.16.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 4 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Medium ･･･ The potential targets are users with public IP addresses to which the attacker can directly connect.
  - Severity: Medium ･･･ While the target’s data flow would be hampered, the financial damage would be limited.
- Impact on financial trading systems
  - Scope: Medium ･･･ The potential targets include miners with public IP addresses to which the attacker can directly connect.
  - Severity: Medium ･･･ While miners’ data flow would also be hampered, the financial damage would still be limited.
- Occurrence probability
  - Ease of attack: Low ･･･ Falsifying routing tables is difficult.
  - Incentive: Medium ･･･ The financial damage would be limited.

6.1.17. Tampering Attack

6.1.17.1. Outline

This attack is aimed at delaying a particular node's delivery of transactions or blocks in order to use it for a DDoS or double-spending attack, or enhance the attacker's mining capabilities correspondingly. The attack targets honest miners and users.

The attacker (*A*) must know the IP address of the target (*V*) so as to connect directly to *V*.

By abusing Bitcoin bandwidth optimization techniques and the measures in place to tolerate network delays and congestion, *A* can delay the delivery of transactions or blocks to *V*'s nodes without changing the network configuration.

This attack is conducted following the steps below. (See Figure 6-11: Schematic overview of tampering attack.)

(1) Upon receiving a transaction or block, *A* immediately relays it to *V* without verification. As neighboring nodes are still verifying that transaction or block, A becomes the first node that relays the data to *V*. To minimize the network's bandwidth consumption, Bitcoin nodes request a given piece of data only from a single peer. Hence, *A*'s neighboring nodes do not accept requests from any other node for a certain period of time. In a case where *A* created that data, all *A* needs to do is send the data to *V* before broadcasting it to the network.

(2) Before sending the main body of data, a Bitcoin node sends a hash of the data using an *inv* message; and, only if the recipient node has not previously received the data advertised by the *inv* message, it requests that the main body of the data be sent, using a *getdata* message. If *inv* messages for the same data are sent from multiple nodes, the recipient node inserts those nodes in a first-in first-out (FIFO) buffer and repeats a receipt of the data from each of those nodes setting a certain length of timeout. The probability of the success of this attack correlates with the length of time *V* waits until it receives the requested data from *A* after sending a *getdata* message to *A*. Thus, *A* sends back-to-back *inv* messages to *V*. As a timeout for requesting the delivery of a transaction is set at two minutes, *A* sending *x* number of back-to-back *inv* messages makes the timeout 2*x* minutes.

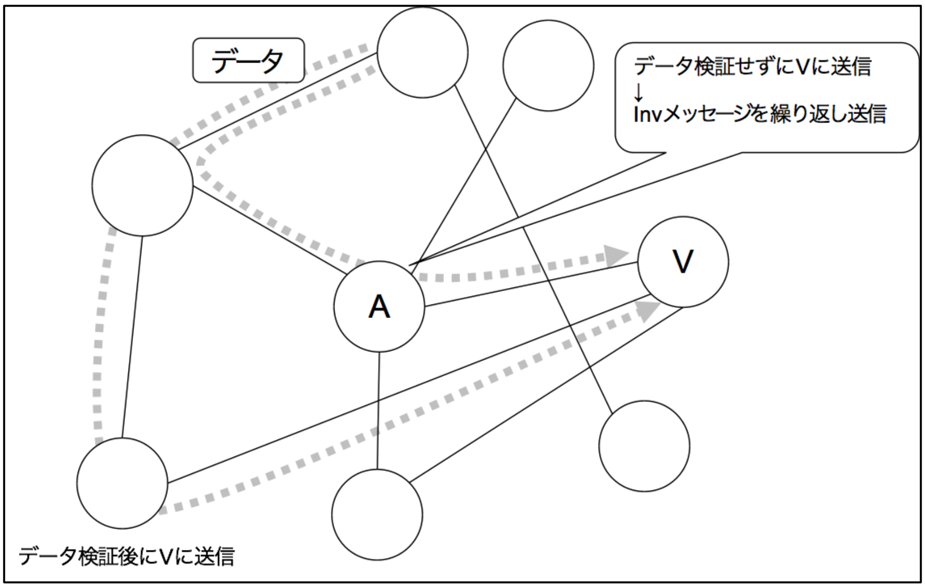


Figure 6-11: Schematic overview of tampering attack

=============

図 6-11 の訳

1. Data
2. Relay data to *V* without verifying it
   ↓
   Send back-to-back *inv* messages
3. Verify data before sending it to *V*

============

6.1.17.2. Countermeasures

To counter this attack, the following measures are proposed:

- Change to a system that allows each node to dynamically set the timeout value according to message sizes and other factors. This would enable the detection of multiple *inv* messages sent from the same node.
- Replace *inv* messages with the method of sending block headers only.
- Accept only one *inv* message of the same transaction per IP address, or randomly choose an incremental number of peers to contact if the first peer did not answer to the *getdata* message.
- Penalize nodes that constantly delay data delivery after a *getdata* message, and disconnect them from the network.

6.1.17.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 4 (Medium)
- Assessment score (Impact on financial trading systems × Occurrence probability): 4 (Medium)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Medium ・・・ The attack would impact the users of the target's node.
  - Severity: Medium ・・・ While the target's data flow would be hampered, the financial damage would be limited.
- Impact on financial trading systems
  - Scope: Medium ・・・The attack would impact the miners working on the target's node.
  - Severity: Medium ・・・ While miners' data flow would also be hampered, the financial damage would still be limited.
- Occurrence probability
  - Ease of attack: Low ・・・ The attacker must be a full node.
  - Incentive: Medium ・・・ The financial damage would be limited.

6.1.18. Deanonymization

6.1.18.1. Outline

This attack is aimed at invading users' privacy by linking their IP addresses to their Bitcoin or private key addresses. As such, it targets Bitcoin users.

As the attacker (*A*) is required to connect to numerous servers in the Bitcoin network, it must have botnets or a large number of nodes. The goal of this attack is to link the hash value of a user's public key to the user's IP address, thereby obtaining information on their past payments, receipts, and other transactions. This attack is conducted following the steps below.

(1) *A* either disables Tor in the Bitcoin network or targets users not utilizing Tor only. One way to disable Tor is to have a Tor node send malformed messages to the Bitcoin network, causing the network to ban connecting to that node for 24 hours.

(2) *A* compiles a list (*S*) of all servers on the Bitcoin network, by sending GETADDR messages to known nodes and collating addresses contained in the ADDR messages sent in response thereto. This list is updated on a periodic basis.

(3) A compiles a list (*C*) of the IP addresses of Bitcoin clients whose public-key hash values *A* wants to obtain through this attack.

(4) By connecting to servers listed in *S* and collecting the addresses that a client (*V*) listed in *C* sends upon connection to the Bitcoin network, *A* identifies pairs (*E'P*) of *V*'s addresses and nodes *V* connects to.

(5) *A* gathers *inv* messages from severs listed in *S* and collects a set (*RT*) of the addresses of nodes that relay the same *inv* message.

(6) Comparing *E'P* with *RT*, *A* obtains pairs of *V*'s IP addresses and corresponding transactions.

6.1.18.2. Countermeasures

One measure proposed to counter this attack is to change a client's IP address after every transaction and add some random delay after the transaction. While this would make it impossible for *A* to link the client's IP address to its transactions, the risk of *V*'s internet service provider learned by *A* could not be mitigated. Another possible way to prevent this attack is to use a mixing service that mixes multiple transactions before redistributing them among ultimate payees.

6.1.18.3. Assessment

The risk of this attack is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 3 (Low)
- Assessment score (Impact on financial trading systems × Occurrence probability): 1 (Low)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ･･･ Anyone could carry out this attack.
  - Severity: Low ･･･ There would be no financial damage.
- Impact on financial trading systems
  - Scope: Narrow ･･･ There would be no impact on systems.
  - Severity: Low ･･･ There would be no impact on systems.
- Occurrence probability
  - Ease of attack: Low ･･･ The attacker needs to listen to the server's responses.
  - Incentive: Low ･･･ The attacker would not gain any direct proceeds.

6.1.19. Compromise of Underlying Cryptographic Algorithms

6.1.19.1. Outline

The blockchain architecture does not take into account migrations from the underlying cryptographic algorithms, the compromise of which might therefore lead to the loss of the entire value of financial trading relying on them. As an example, take digital signatures that are provided to secure the authenticity of the transactions constituting blockchains (in other words, to guarantee that the contents of the transactions have not been falsified). A compromise of a digital signature algorithm would allow transactions to be falsified, making it impossible to secure the authenticity thereof. This would render trading data contained in those transactions unreliable, resulting in the loss of the value of financial trading.

There are two ways in which cryptographic algorithm compromise materializes. One is the computational cryptanalysis of a cryptographic algorithm, made possible by increases in computing power. The other is the discovery of an efficient cryptanalysis method against a cryptographic algorithm. While the former risk is bound to materialize every several decades, the latter is an imminent risk. Actual examples of these two will be given in Appendix A below.

As these points suggest, the long-term stable operation of blockchain-based financial trading requires a system capable of addressing the risk of cryptographic algorithm compromise, which in turn demands the use of blockchains that enable the underlying cryptographic algorithms to be changed. By doing so, even in the event that a previously used cryptographic algorithm is compromised, the authenticity of the blockchains that have been processed up to that point can be secured.

6.1.19.2. Countermeasure

One measure proposed to counter the risk of cryptographic algorithm compromise is to apply a long-term signature scheme and thereby switch to another cryptographic algorithm before the current one is compromised. Under this proposal, during a period in which both the old and new cryptographic algorithms are in use, the validity of the financial trading conducted up to that point would be guaranteed using the new algorithm.

To achieve this, a paper we have studied[39] suggests the following two methods:

- Transition within the original chain. (See the paper's Figure 6 shown below.)
- Transition with a support chain. (See the paper's Figure 7 shown below.)

[39] M. Sato and S. Matsuo, *Long-term Public blockchain: Resilience against Compromise of Underlying Cryptography*," 2017 International Conference on Computer Communication and Networks.

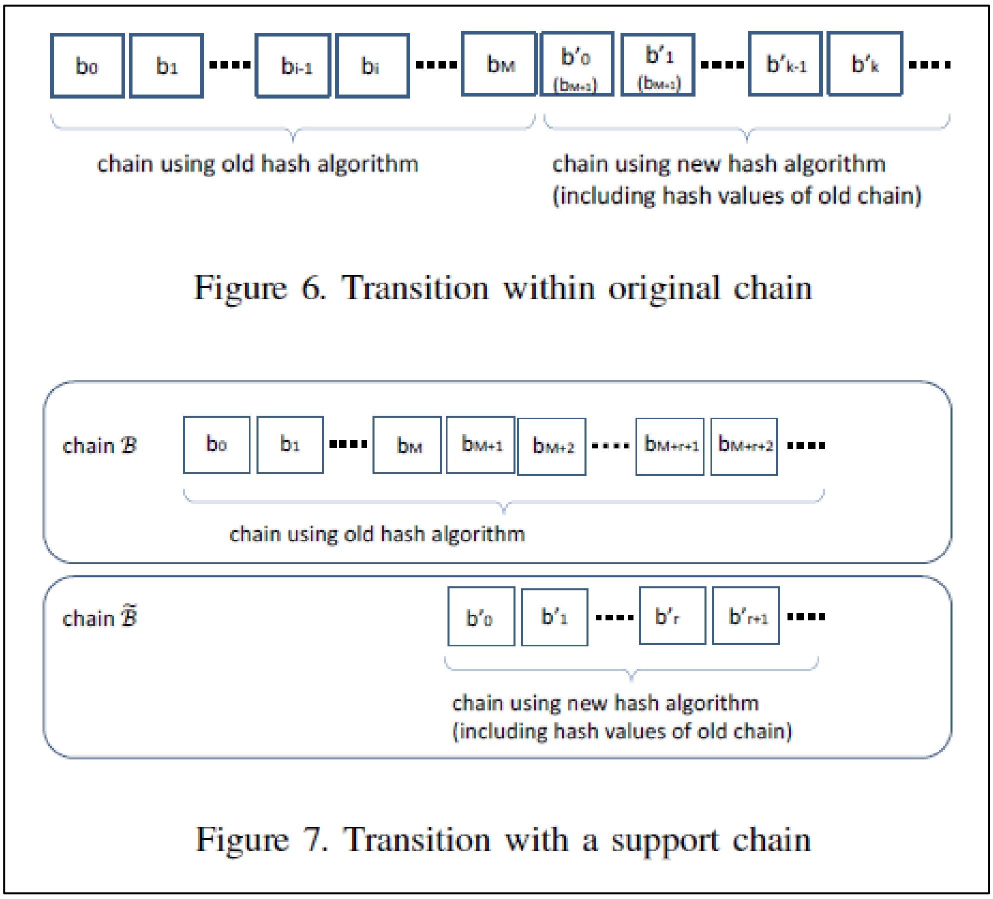

b_0
b_1
b_{i-1}
b_i
b_M
b'_0 (b_{M+1})
b'_1 (b_{M+1})
b'_{k-1}
b'_k
chain using old hash algorithm
chain using new hash algorithm (including hash values of old chain)
Figure 6. Transition within original chain
chain $\mathcal{B}$
b_0
b_1
b_M
b_{M+1}
b_{M+2}
b_{M+r+1}
b_{M+r+2}
chain using old hash algorithm
chain $\widetilde{\mathcal{B}}$
b'_0
b'_1
b'_r
b'_{r+1}
chain using new hash algorithm (including hash values of old chain)
Figure 7. Transition with a support chain

Figure 6-12: Methods to implement blockchains using the long-term signature scheme (Source: Reproduced from the aforementioned paper.)[39]

Implementing blockchains using the long-term signature scheme requires attention to be given to points listed below.

(1) Management of different client versions is needed.

- A version using the old algorithm only.
- A version to which the new algorithm is added (the old algorithm still being secure).
- A version accepting the new algorithm only (blocks using the old algorithm needing to undergo additional verification).

(2) It is necessary to consider how to reconcile the transactions in the support chain method (such as whether rewards to miners should be included to one of them only).

(3) Implementation via a soft fork calls for changes to the block headers (because the old algorithm is not equipped with a mechanism that disregards the added fields).

(4) In the case of the old and new algorithms coexisting, blocks created using the new algorithm cannot be verified under the old algorithm. This raises questions of how to motivate miners to work on unverifiable blocks and how to migrate from the old algorithm.

Appendix A: Actual Examples of Cryptographic Algorithm Compromise

(1) Compromise attributable to increases in computational power

Examples of compromise attributable to increases in computational power include the holding of Data Encryption Standard (DES) Challenges and the establishment of the Advanced Encryption Standard (AES). Published in 1977 by NIST as a FIPS, DES is a block cryptographic algorithm with a key length of 64 bits. In spite of a quinquennial review, however, the security of DES degraded significantly following the discovery of linear cryptanalysis and increases in computational power. In the DES Challenge cryptanalysis competition, which was held for the first time in 1997, DES was eventually broken in 22 hours and 15 minutes in 1999. To address this situation, NIST proposed Triple DES, which applies the DES algorithm three times to each data block so as to make the use of DES securer. At the same time, NIST kicked off a project to develop a block cryptographic algorithm standard that would replace DES, which culminated in the 2001 publication as a FIPS of AES that is a block cryptographic algorithm with a key length of 128 bits. In 2005, the FIPS dictating the specifications of DES was withdrawn, which essentially banned the use of DES.

(2) Compromise attributable to the discovery of cryptanalysis methods

Cases of compromise attributable to the discovery of a cryptanalysis method include the cryptanalysis of Message-Digest Algorithm 5 (MD5), and the resultant ban on the use of the Authenticated Post Office Protocol (APOP).[40][41] MD5 is a hash function developed in 1991 that produces a 128-bit hash value, while APOP is a protocol for authenticating incoming email using MD5. How APOP password authentication works is illustrated in Figure 6-13 below.

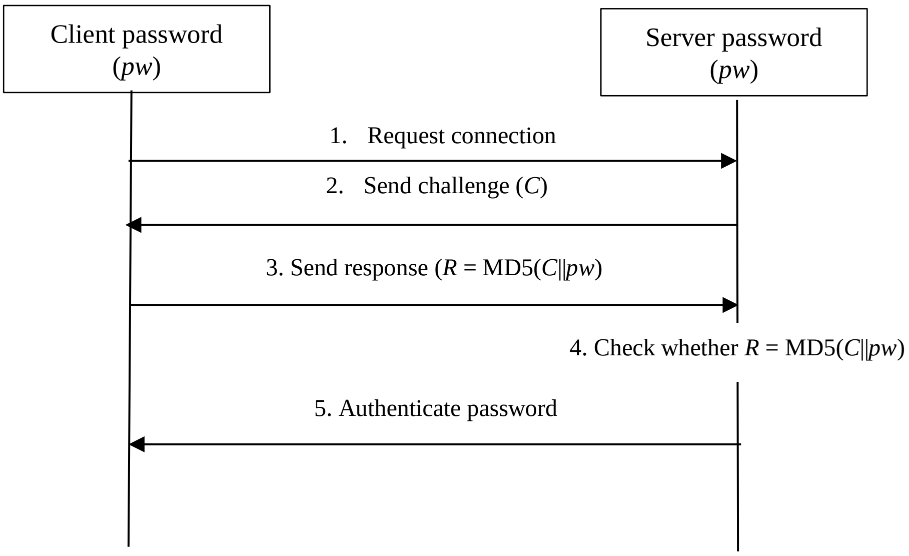


Figure 6-13: APOP password authentication

In 2004, researchers demonstrated a collision attack against MD5 able to be implemented in practical time, which was followed by a 2007 report on the application of this attack to APOP that would enable the recovery of the first 31 characters of APOP password in practical time. As a result of the APOP vulnerability discovered by this attack, a recommendation was issued that the use of APOP be avoided.

The blockchain's cryptographic algorithms that were eventually broken include the Curl hash function used in the IOTA cryptocurrency.[42] This incident is believed to be attributable in part to the use in IOTA of a hash function that was designed by the IOTA project rather than the one whose security has been thoroughly scrutinized. In the wake of the discovery of this vulnerability, IOTA underwent a hard fork to replace Curl with a hash function called Keccak (also known as Secure Hash Algorithm 3 [SHA-3]). This process led to an about three-day stoppage of trading in this cryptocurrency.

[40] IPA, *Vulnerability Notice: Security Alert for APOP Vulnerability* [in Japanese], https://www.ipa.go.jp/security/vuln/200704_APOP.html.

[41] IPA, *A Study on the Limitation of MD5 Security* [in Japanese], July 2008, https://www.ipa.go.jp/files/000013897.pdf.

[42] *IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency*, https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md.

6.1.19.3. Assessment

The risk of this factor is assessed as outlined below.

- Assessment score (Impact on users × Occurrence probability): 9 (High)
- Assessment score (Impact on financial trading systems × Occurrence probability): 9 (High)

The assessment is based on the grounds below.

- Impact on users
  - Scope: Broad ・・・ Anyone could be targeted.
  - Severity: High ・・・ The value of assets might be wiped out.
- Impact on financial trading systems
  - Scope: Broad ・・・ The systems might be disabled.
  - Severity: High ・・・ The systems might be disabled.
- Occurrence probability
  - Ease of attack: Low ・・・ The cryptographic algorithms currently in use are secure.
  - Incentive: High ・・・ Proceeds could be obtained by falsifying transactions.

6.2. Glossary

Fast payment

A scheme that is used primarily for small-value transactions at ordinary stores. Under this scheme, a payment is accepted without the store waiting for the transaction to be incorporated into a blockchain and for more than one block to be added to the end of the chain. The payment is deemed completed when the store having received the transaction verifies the signature and confirms that the coins with which the payment is made are unused.

Address

An identifier, like an account number, that is needed when using Bitcoin. A Bitcoin address consists of 27 to 34 alphanumeric characters that begin with the number 1 or 3, generated from the public key.

P2P

The acronym for peer to peer, referring to an architecture that allows nodes to communicate directly with each other rather than through a central server and such.

Node

Individual programs that participate in the Bitcoin network, which include mining nodes, wallet nodes, and simplified payment verification (SPV) nodes.

IP address

A numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

Broadcast

Sending the same data simultaneously to an unspecified large number of nodes. In a Bitcoin transaction, the transaction information is broadcast to all the nodes on the network; then, upon confirmation of the transaction by a node, the confirmed transaction data (block) is rebroadcast to all the nodes on the network.

Fork

A divergence in a blockchain that occurs typically when two or more miners find blocks at nearly the same time and add those blocks to the end of the same chain.

Bitcoin mixer

A tool designed to increase anonymity for Bitcoin use.

Off-chain

Conducting the payments and other transactions outside the blockchain that ought to be made on the blockchain.

Static IP address

A fixed IP address.

Mining pool

A protocol for a group of miners to work together.

Block confirmation

A state in which a Bitcoin transaction is incorporated into a block on the blockchain.

Cloud mining

Investing in mining services.

BIP 70

A protocol that has been developed to prevent a man-in-the-middle attack against the payment process using Bitcoin between a merchant and its customer.

Private key

A key kept secret that is paired with a public key in public-key cryptography.

Soft fork

A divergence in a blockchain that occurs by tightening the verification protocol. This is a backward compatible update.

Hard fork

A divergence in a blockchain that occurs by relaxing the verification protocol. This is a non-backward compatible update that splits the blockchain into two incompatible paths.

Mixing

A system that enhances Bitcoin anonymity by making it impossible for third parties to trace when certain coins were obtained. This is said to be classified into three generations: centralized mixing services, peer-to-peer mixers, and anonymous altcoins.

Botnet

A network of numerous zombie computers hijacked by cybercriminals using Trojan horses and other malicious programs.

DDoS attack

A DoS attack against one service that occurs from more than one source at the same time.

DoS attack

A denial of service attack, which attempts to prevent legitimate users from accessing a web service by intentionally imposing an excessive burden on or exploiting a vulnerability in the server, network, or other resources providing the service.

Routing

Selecting the optimal path for, and forwarding, a packet to the destination host.

TCP connection

A connection according to the Transmission Control Protocol, which is one of the most commonly used internet protocols, like IP.

Inbound/outbound connections

A method of connection in the Bitcoin network. An outbound connection means that a node makes a connection to another, while an inbound connection refers to the node receiving a connection from another.

Hash value

A hash referring to a cryptographic function to identify a message. It is used by a recipient to make sure that the message has not been falsified on the communication route and the data received has not been broken.

Tor

A name of a technology, or software to realize the technology, that enables anonymous communication by encrypting it and connecting multiple nodes through a proxy.

Public key

A key known to all users that is paired with a private key in public-key cryptography.

Public-key cryptography

A class of cryptographic systems in which separate keys (protocols) are used for encryption and decryption, and the encryption key can be made public.

Collision attack

Primarily an attack on cryptographic hash functions that attempts to find any two inputs producing the same hash value (i.e., a hash collision).

Preimage attack

An attack attempting to find a message that has a specific hash value. This is classified into two forms: a first preimage attack that, given a hash value h, attempts to find a message m such that hash(m) = h; and a second preimage attack that, given a fixed message m_1, attempts to find a different message m_2 such that hash(m_2) = hash(m_1).

Digital signature

An element of public-key cryptography, which is used to replicate the security of handwritten signatures on paper.

Cryptocurrency

A currency that has no official issuer or administrator (like central banks), and can be used to pay for goods and services on the internet among an unspecified large number of people.

Distributed database

A database in which a database management system controls storage devices that are attached to multiple computers.

Supply chain

A series of activities ranging from raw material production to the delivery of finished products to consumers.

Memory pool

A region of memory that is allocated in advance by software (in this context, a Bitcoin client).

6.3. Bibliography

6.3.1. Method of Selecting Papers to Study

The papers referred to for this research study have been selected in the following manner:

(1) Identified existing attack methods (risk factors) and the papers pertaining thereto from the survey paper listed in Item Number 0.

(2) Besides those identified in (1) above, picked out the pertinent papers from among those presented at relevant conferences: namely, Financial Cryptography and Data Security, an international forum for the application of cryptography in the financial field, and the associated workshops; and Crypto, Eurocrypt, Asiacrypt, and other major international conferences organized by the International Association for Cryptologic Research.

(3) Ultimately narrowed these articles down to 27 papers including the survey paper listed in Item Number 0.

Note that, during the selection process, we focused on papers available online gratis to the exclusion of paid papers, in order for readers to freely peruse the papers that interest them.

Table 6-1: List of Studied Papers

Item no.	Category	Risk factor	Studied paper			
			Author(s) and title	Media published	Available at:	Type of publication
0	Survey paper	—	●M. Conti, S. Kumar E, C. Lal, and S. Ruj, *A Survey on Security and Privacy Issues of Bitcoin*.	●arXiv	https://arxiv.org/pdf/1706.00916	Article
1	Double-spending attacks	Double-spending (or race) attack	●G. O. Karame, E. Androulaki, and S. Capkun, *Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin*.	●2012 Association for Computing Machinery (ACM) Conference on Computer and Communications Security (CCS)	https://eprint.iacr.org/2012/248.pdf	Article
2		Finney attack	●H. Finney, "Re: Best Practice for Fast Transaction Acceptance—How High Is the Risk?"	●Bitcoin Forum	https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384	Website
3		Brute-force attack	●M. Rosenfeld, *Analysis of Hashrate-Based Double-Spending*.	●arXiv	https://arxiv.org/pdf/1402.2009.pdf	Article
4		Vector 76 (or one-confirmation) attack	●vector76, "Re: Fake Bitcoins?" ●sgornick, "Vector76 Double Spend Attack?"	●Bitcoin Forum ●reddit.com	https://bitcointalk.org/index.php?topic=36788.msg463391#msg463391 https://www.reddit.com/r/Bitcoin/comments/2e7bfa/vector76_double_spend_attack/	Website
5		>50% hashpower (or Goldfinger or majority) attack	●J. A. Kroll, I. C. Davey, and E. W. Felten, *The Economics of Bitcoin Mining or, Bitcoin in the Presence of Adversaries*.	●Workshop on the Economics of Information Security 2013	http://www.thebitcoin.fr/wp-content/uploads/2014/01/The-Economics-of-Bitcoin-Mining-or-Bitcoin-in-the-Presence-of-Adversaries.pdf	Article

Item no.	Category	Risk factor	Studied paper			
			Author(s) and title	Media published	Available at:	Type of publication
6	Mining pool attacks	Block discarding or selfish mining	●N. T. Courtois and L. Bahack, *On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency*. ●L. Bahack, *Theoretical Bitcoin Attacks with Less than Half of the Computational Power*. ●I. Eyal and E. G. Sirer, *Majority Is Not Enough: Bitcoin Mining Is Vulnerable*.	●Computing Research Repository	https://allquantor.at/blockchainbib/pdf/courtois2014subversive.pdf https://eprint.iacr.org/2013/868.pdf https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf	Article
7	Mining pool attacks	Block withholding	●M. Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*.	●arXiv	https://bitcoil.co.il/pool_analysis.pdf	Article
8		Bribery attack	●J. Bonneau, *Why Buy When You Can Rent? Bribery Attacks on Bitcoin-style Consensus*.	●Third Workshop on Bitcoin and Blockchain Research	http://fc16.ifca.ai/bitcoin/papers/Bon16b.pdf	Article
9		Refund attack	●P. McCorry, S. F. Shahandashti, and F. Hao, *Refund Attacks on Bitcoin's Payment Protocol*.	●Cryptology ePrint Archive	https://eprint.iacr.org/2016/024.pdf	Article
10		Punitive and feather forking	●A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies*. ●A. Miller, "Feather-Forks: Enforcing a Blacklist with Sub-50% Hash Power."	●Princeton University Press ●Bitcoin Forum	https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf https://bitcointalk.org/index.php?topic=312668.0	•Book •Website
11	Client security threat	Wallet theft	• "Storing Bitcoins."	●Bitcoin Wiki	https://en.bitcoin.it/wiki/Securing_your_wallet	Website
12	Attacks against Bitcoin protocols or network infrastructure	Transaction malleability	• M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, *On the Malleability of Bitcoin Transactions*. • C. Decker and R. Wattenhofer, *Bitcoin Transaction Malleability and MtGox*.	●Financial Cryptography and Data Security 2015 ●European Symposium on Research in Computer Security (ESORICS) 2014	https://ai2-s2-pdfs.s3.amazonaws.com/c276/84f2fe5a85fe2871f693edc46061d0ecb20d.pdf https://www.tik.ee.ethz.ch/file/7e4a7f3f2991784786037285f4876f5c/malleability.pdf	Article
13		Time jacking	• corbixgwelt, "Timejacking & Bitcoin."	●culubas	http://culubas.blogspot.jp/2011/05/timejacking-bitcoin_802.html	Website
14		Sybil	• J. R. Douceur, *The Sybil Attack*.	●First International Workshop on Peer-to-Peer Systems	http://www.divms.uiowa.edu/~ghosh/sybil.pdf	Article
15		DDoS	• M. Vasek, M. Thornton, and T. Moore, *Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem*.	●Financial Cryptography and Data Security 2014	http://fc14.ifca.ai/bitcoin/papers/bitcoin14_submission_17.pdf	Article

Item no.	Category	Risk factor	Studied paper			
			Author(s) and title	Media published	Available at:	Type of publication
16		Eclipse (or netsplit) attack	• E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, *Eclipse Attacks on Bitcoin's Peer-to-Peer Network*.	● Advanced Computing Systems Association (USENIX) Security Symposium 2015	https://eprint.iacr.org/2015/263.pdf	Article
17		Tampering attack	• A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, *Tampering with the Delivery of Blocks and Transactions in Bitcoin*.	● ACM CCS 2015	https://scalingbitcoin.org/hongkong2015/presentations/DAY1/3_block_propagation_2_gervais.pdf	Article
18		Deanonymization	• P. Koshy, D. Koshy, and P. McDaniel, *An Analysis of Anonymity in Bitcoin Using P2P Network Traffic*. • A. Biryukov, D. Khovratovich, and I. Pustogarov, *Deanonymisation of Clients in Bitcoin P2P Network*.	● Financial Cryptography and Data Security 2014 • ACM CCS 2014	https://pdfs.semanticscholar.org/c277/62257f068fdbb2ad34e8f787d8af13fac7d1.pdf https://orbilu.uni.lu/bitstream/10993/18679/1/Ccsfp614s-biryukovATS.pdf	Article
19		Compromise of underlying cryptographic algorithms	• I. Giechaskiel, C. Cremers, and K. B. Rasmussen, *On Bitcoin Security in the Presence of Broken Crypto Primitives*. • M. Sato and S. Matsuo, *Long-term Public Blockchain: Resilience against Compromise of Underlying Cryptography*.	• ESORICS 2016 • International Conference on Computer Communications and Networks (ICCCN) 2017	https://eprint.iacr.org/2016/167.pdf ICCCN_Blockchain_matsuo_cameraready_8p.pdf	Article

6.4. Details of Preparation for Proof of Concept

6.4.1. Implementation of Cryptographic Algorithm Compromise Countermeasures on BSafe.network's Bitcoin

We added the following modifications to BSafe.network's Bitcoin environment:

6.4.1.1. Implementing archiveHash

We implemented the computation and verification capabilities of archiveHash in accordance with the policies below.

- Update the consensus rules and carry out a hard fork (the result being that nodes not supporting archiveHash would be left in the original chain).
- Add the following three parameters to the consensus rules:
    - ➢ The height of a chain at which archiveHash begins to be added.
    - ➢ The number of blocks archived per archiveHash.
    - ➢ The number of archiveHash added.
- Implement the following to support SPV nodes:
    - ➢ Split archiveHash into archiveHeaderHash and archiveTransactionHash.
    - ➢ Separate archiveWitnessHash from archiveTransactionHash to ensure consistency with SegWit.

Specifically, archiveHash was implemented in the following ways:

- Add to a block header the three hash values mentioned above—namely, archiveHeaderHash, archiveTransactionHash, and archiveWitnessHash (as a result of which the block header would increase by 96 bytes).
- Add to a block header's version field a flag that indicates whether it has archiveHash or not.
- In cases where a node with the archiveHash setting in its consensus rules receives a new block, ascertain that the archiveHash-addition flag is off if the height of the block is outside the range defined by the archiveHash setting; conversely, if the height is within the range, ascertain that the flag is on, identify a corresponding block to be archived from the blockchain, and verify whether the three hash values (in the case of an SPV node, archiveHeaderHash only) computed are identical to the hash values of the new block.
- Compute respective hash values in the manners outlined below.
    - ➢ archiveHeaderHash
        - Compute this hash by combining the byte sequence representation of each block header to be archived with the hash value of the last block header.
    - ➢ archiveTransactionHash
        - Compute a hash value of a transaction contained in each block to be archived, using a Merkle tree.
        - Note that, for SPV nodes, a better approach to be adopted in the future is to verify transactions to be archived using a partial Merkle tree (which was not implemented in this proof of concept).
    - ➢ archiveWitnessHash
        - Compute a witness hash value of a transaction contained in each block to be archived, using a Merkle tree.

6.4.1.2. Implementing SHA-512 Hash Function

We replaced SHA-256, a hash function used in the proof-of-work protocol and a transaction Merkle tree, with SHA-512. This was implemented according to the policies, and with attention given to the points, described below.

- The current Bitcoin implementation takes no account whatsoever of the possibility of changing the bit length of a hash function, as its code is premised on the hash value being 256 bits. This makes it difficult to implement a switch to a hash function that involves a change of the bit length, as the impact on code would be far-reaching. In this proof of concept, therefore, we decided to reduce the output length of

SHA-512, the new hash function, to 256 bits.

- Replace the hash function used in computing the hash value of a block header (proof-of-work), a transaction Merkle tree, and archiveHash with the new hash function.
- Continue to use the existing RIPEMD-160 hash function for computing transaction IDs.
- Update the consensus rules and carry out a hard fork (the result being that nodes not supporting the new proof-of-work would be left in the original chain).
- Add to the consensus rules a new parameter concerning the height of a chain migrating to the new proof-of-work.

Specifically, SHA-512 was implemented in the following ways:

- Reduce the output length of the SHA-512 that has already been included in the Bitcoin implementation to 256 bits to implement it as a new hash function.
- Add to a block header’s version field a flag that indicates whether to use the new proof-of-work.
- When the height of a chain about to create a block is equal to or greater than the height required by the consensus rules, create a block with the new proof-of-work flag being on and compute its Merkle tree using the new hash function.
- When computing the hash value of a block header, check the status of the flag and switch hash functions accordingly.
- Verify blocks in the manners outlined below.
  - In cases where a node with the new proof-of-work setting in its consensus rules receives a new block, ascertain that the block’s flag is off if the height of the chain is less than the required value; conversely, if the height of the chain is equal to or greater than the required value, ascertain that the flag is on.
  - Continue to use the existing hash function for verifying the remaining portion of blocks, except for switching hash functions according to the status of the flag.

6.4.1.3. Changing ECDSA Key Length

We changed the ECDSA key length from the current 256 bits to 384 bits, which in turn necessitated the change of the curve in use. These were implemented in accordance with the policies below.

- In addition to the current signature algorithm that is ECDSA using the secp256k1 curve, enable the use of the secp384r1 curve.
- Use OpenSSL for implementing the secp384r1 curve.
- Add commands that transfer the balance of a wallet to a new address.

Specifically, the change in the ECDSA key length was implemented in the following ways:

- Change the private key length from being fixed at 32 bytes to being configurable at either 32 or 48 bytes, managing it using a flag.
- Store private keys in the DER format, similarly to the present Bitcoin system.
- When restarting a node, load a wallet in the format being used in the Bitcoin system; if that fails, load the private key having 48 bytes.
- As for the public key length, add 49 bytes (compressed representation) to the current 33 bytes (compressed representation) and 65 bytes (non-compressed).
- Add a new flag to determine the type of a signature algorithm at the first byte of the public key.
- Record signatures in the DER format similarly to the current practice.
- Check the status of the private key flag at the time of signing; and in the case of a 48-byte private key, generate the signature by invoking the OpenSSL ECDSA library.
- Verify signatures either by using the current Bitcoin method or by invoking the OpenSSL ECDSA library, depending on the number of bytes of the public key.

- When verifying a signature, parse the DER format using the current Bitcoin method; if that succeeds, verify the signature using the current Bitcoin method; otherwise, verify the signature invoking the OpenSSL ECDSA library.
- For Bitcoin addressees, continue to use P2PKH addresses with the public key of 160-bit hash value.
- Add commands that generate a key using the new method and add the key to a wallet, the address of which is designated as the default recipient. This makes it possible to transfer the entire funds to the new address, thereby migrating to the ECDSA key having the new length within the wallet.
- With regard to an address at which change is received, replace the method of generating a new address each time with the method of using the address designated as the default recipient.

6.4.2. Deployment in BSafe.network

We deployed nodes of the following two types in the BSafe.network environment:

- Full nodes; and
- SPV nodes.

These two types of nodes were deployed in networks corresponding to each of the five scenarios indicated in 4.3.3.

6.4.2.1. Deploying Full Nodes

A program with the capabilities outlined in 6.4.1 was installed on each of the full nodes deployed in the BSafe.network's Bitcoin environment, following the steps below.

(1) Deploy the source code obtained from GitHub.

```
$ git clone https://github.com/BSafe-network/LongTermBlockchain
$ cd LongTermBlockchain
$ git checkout bsafe-long-term-20180130 # Tags corresponding to Scenarios 1 to 3: bsafe-long-term-20180130, Tags corresponding to Scenarios 4 and 5: bsafe-long-term-20180215
$ ./autogen.sh
$ ./configure
$ make
$ make install # optional
```

(2) Create a directory for each scenario.

```
$ mkdir bsafenetlt1
```

(3) Set bitcoin.conf in the directories created.

```
$ vi bsafenetlt1/bitcoin.conf
bsafenetlt1=1    # or bsafenetlt12 or bsafenetlt3 or bsafenetlt4 of bsafenetlt5
dnsseed=0
upnp=0

server=1
rpcallowip=0.0.0.0/0

rpcuser=user
rpcpassword=password
seednode=202.16.211.119 # the seed node’s IP address (to be changed depending on the environment)
```

(4) Start bitcoind.

```
$ LongTermBlockchain/src/bitcoind -datadir=bsafenetlt1
```

(5) Execute a command using bitcoin-cli to conduct mining.

```
$ LongTermBlockchain/src/bitcoin-cli -datadir=bsafenetlt1 generate 1
```

6.4.3. Data Used for Proof of Concept

To test Scenarios 1 to 3 of this proof of concept, we transmitted blocks as data, which were created by executing bitcoin-cli, a mining command.

On the other hand, Scenarios 4 and 5 were tested by transmitting transactions as data. The reason is that, as ECDSA is used for digital signatures affixed to transactions, monitoring the impact of the change in the ECDSA key length necessitates transmitting transactions.

For this proof of concept, it was necessary to transmit in the BSafe.network environment the same level of transactions as those flowing on the actual Bitcoin network (“mainnet”). To achieve this, we developed a tool (“txrelay.jar”) to convert transactions flowing on the mainnet into a form acceptable to the BSafe.network environment. When converting a transaction, the payee’s public key and the payment amount were used unchanged. All payments were made from one payer that was a wallet in the BSafe.network environment.

Details of txrelay.jar are given below. First, we connected to a mainnet node and converted received transactions in the following ways:

- Change the mainnet’s magic numbers (network identifiers) in the payees’ addresses to those of BSafe.network.
- Use the payment amounts unchanged.

It was implemented as follows:

```
Map<String, Double> amounts = new HashMap<String, Double>();
for (TransactionOutput output: transaction.getOutputs()) {
        Address address = output.getAddressFromP2PKHScript(params);
        if (address == null) {
                address = output.getAddressFromP2SH(params);
        }
        if (address != null) {
                if (address.isP2SHAddress()) {
                        address = new Address(dstParams, dstParams.getP2SHHeader(),
                        address.getHash160());
                } else {
                        address = new Address(dstParams, dstParams.getAddressHeader(),
                        address.getHash160());
                }
                amounts.put(address.toBase58(), (double)output.getValue().getValue() /
        Coin.COIN.getValue());
        }
}
```

Upon compilation of the list of payees and payment amounts, we sent an RPC command to a BSafe.network node to make payments from the node’s wallet. That was implemented as follows:

```
JSONObject json = sendmany(amounts);
System.err.println(json.toJSONString());

post.setEntity(new StringEntity(json.toJSONString(), StandardCharsets.UTF_8));

try (CloseableHttpResponse response = client.execute(post)) {
   System.err.println("response: " + response);
   System.err.println("response: " + EntityUtils.toString(response.getEntity()));
}
```

6.5. Details of Findings from Proof of Concept

In this proof of concept, we conducted measurement at the full nodes only. Unequipped with measurement capabilities, the SPV nodes were unable to be used for measurement purposes and thus utilized solely to check connections with the full nodes.

6.5.1. Impact on Data Traffic

6.5.1.1. Number of Bytes Sent and Received per Block

We measured the number of bytes sent and received per block in each of the scenarios below to compare their averages.

- Scenario 1 (“bsafenetlt1”)
- Scenario 2 (“bsafenetlt2”)
- Scenario 3 (“bsafenetlt3”)

The nodes were placed in the institutions below.

- Node 1: Toho University, Japan (“node1”)
- Node 2: Keio University, Japan (“node2”)
- Node 3: University of British Columbia, Canada (“node3”)

The network configuration for this proof of concept happened to become a star topology in which two nodes (Nodes 2 and 3) connected to Node 1. In consequence, the measurement results show that traffic was heavily concentrated on Node 1. We infer that this concentration was attributable to the small number of nodes used (only three), and that a greater number of nodes would create an ideal topology for a P2P network, evening out both incoming and outgoing traffic.
Another point of note is that, as test data was entered and mining conducted at Node 1, traffic mainly consisted of data sent from Node 1 to the other nodes.

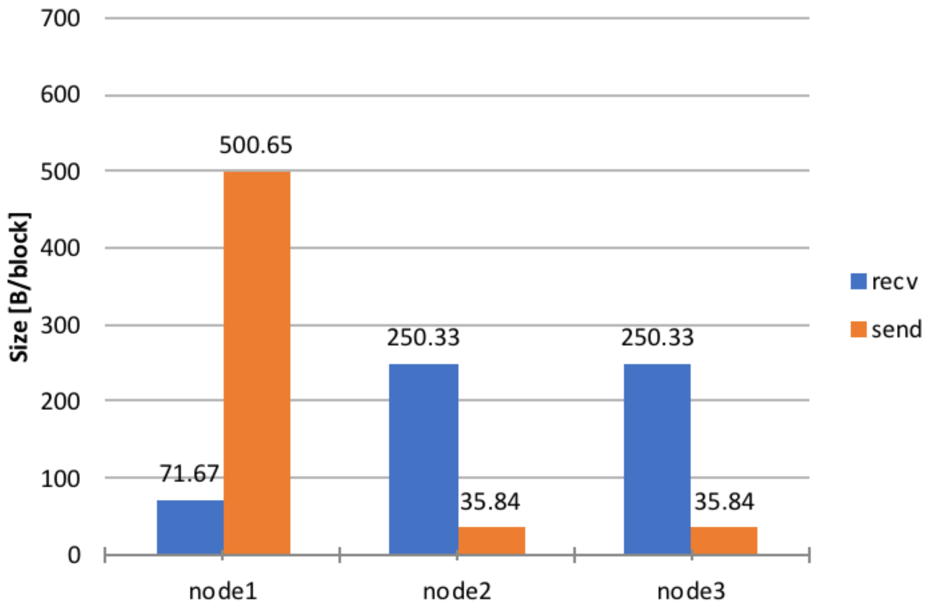


Chart 6-1: Average number of bytes sent and received—Scenario 1

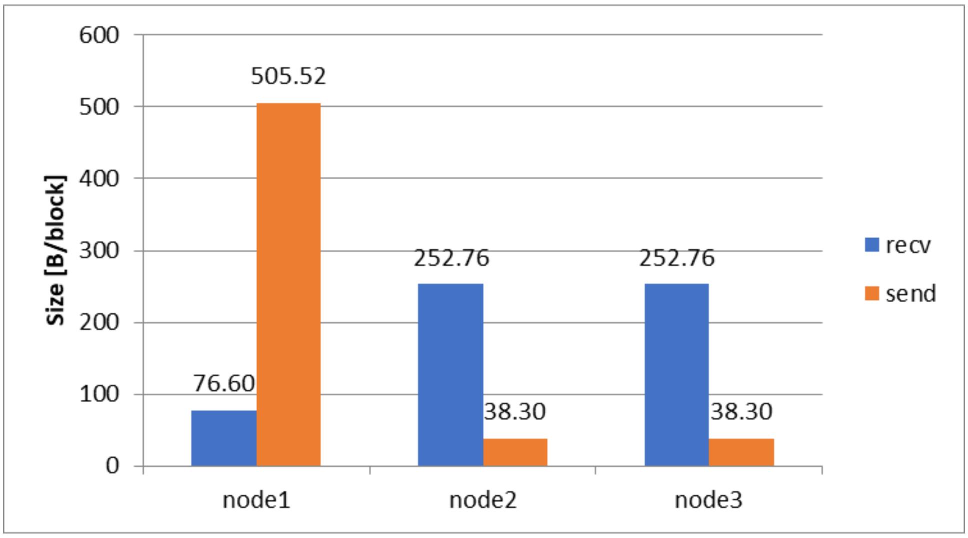


Chart 6-2: Average number of bytes sent and received—Scenario 2

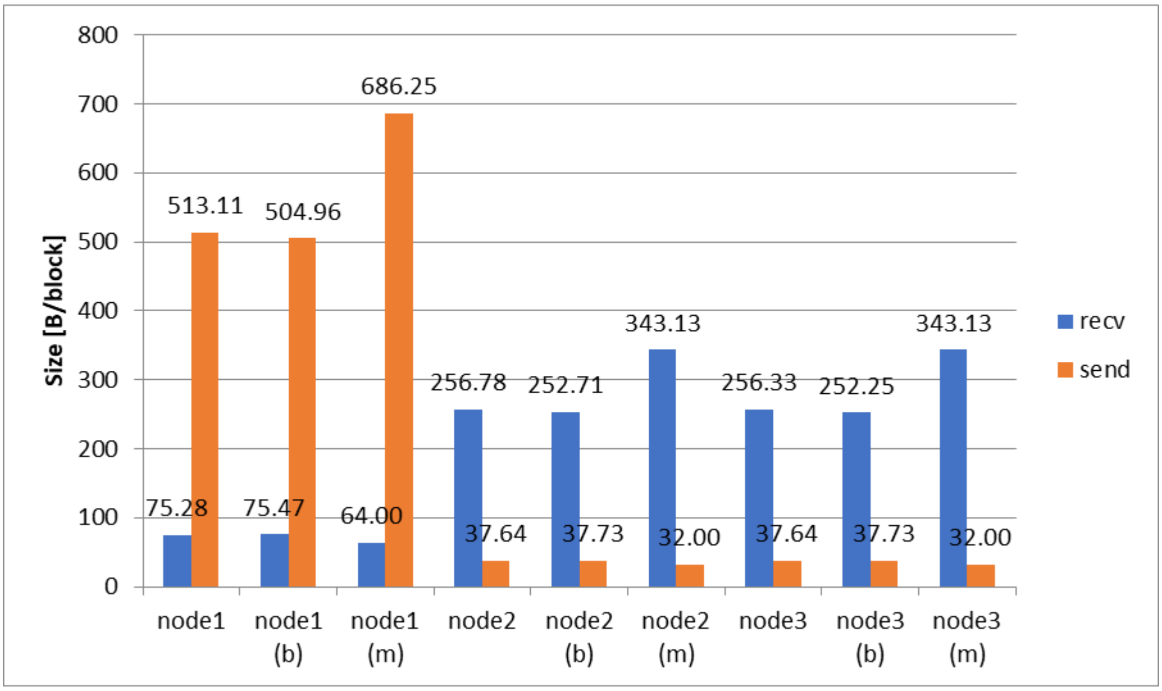


Chart 6-3: Average number of bytes sent and received—Scenario 3
Note: "(b)" denotes data without, and "(m)" denotes data with, archiveHash; the rest of the figures are the overall averages.

In Scenarios 1 to 3, the nodes transmitted not transactions but blocks only. Hence, the impact on block size was directly reflected in the results.

In Scenarios 1 and 2, approximately 250 bytes were sent and received per connected peer. Although these two scenarios differed in terms of hash functions used for the proof-of-work, no substantial difference was observed in the number of bytes sent and received because the output length remained unchanged (see 6.4.1.2 Implementing SHA-512 Hash Function).

In Scenario 3, which differed from Scenario 2 in that archiveHash was activated from a certain block onward, the difference attributable to archiveHash was observed. The number of bytes sent and received before the addition of archiveHash was approximately 250 per connected peer, which was similar to that in Scenarios 1 and 2. When archiveHash was effective, however, the number of bytes sent and received reached around 340 per connected peer.

In Scenario 3, the increase in the number of bytes sent and received while archiveHash was effective reflected a 96-byte increase in a header resulting from the addition of archiveHash. In particular, the transaction overhead recoding rewards to miners and the protocol overhead for the P2P layer were large in size relative to the overall block sent or received, resulting in an about 30% increase in the overall block size.

Next, we compared Scenario 4 with Scenario 5 in terms of the number of bytes sent and received per connected peer. Scenario 4 used BSafe.network's Bitcoin environment with no modifications. In Scenario 5, meanwhile, this environment was modified by replacing the hash function, activating archiveHash, and changing an ECDSA parameter (extending the key length).

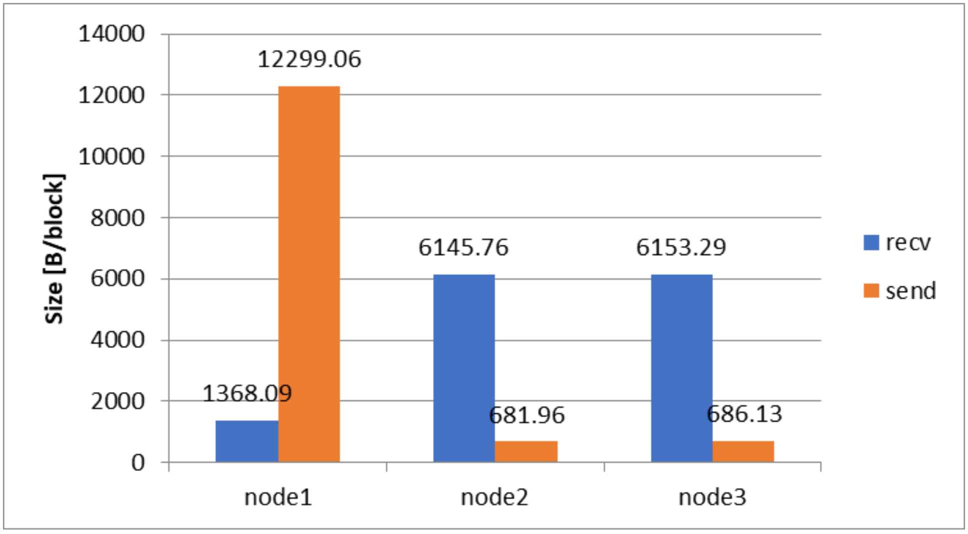


Chart 6-4: Average number of bytes sent and received—Scenario 4

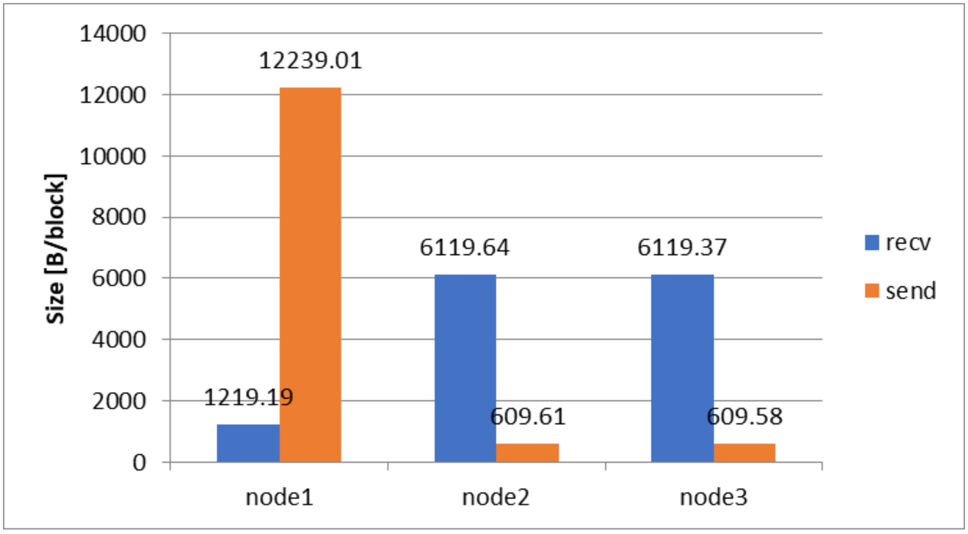


Chart 6-5: Average number of bytes sent and received—Scenario 5

In contrast to Scenarios 1 to 3 in which nodes transmitted blocks only, in Scenarios 4 and 5 both blocks and transactions were transmitted.

In Scenarios 4 and 5, approximately 6,000 bytes were sent and received per connected peer on average. A comparison between Scenarios 1 and 4 reveals that transactions composed about 96% of the number of bytes sent and received.

Consequently, although the modifications were made that affected the block size, virtually no difference was seen in the measurement results of Scenarios 4 and 5. This indicates that, in a near-production environment in which transactions are transmitted, transactions account for the majority of incoming and outgoing traffic on the network, and thus an increase in the signature size resulting from a change to the signature parameters has almost no impact on the network.

6.5.1.2. Analysis of Impact on Data Traffic

The following anticipated impacts were substantiated by the measurement results:

- The addition of archiveHash causes the block size to increase.
- Transactions account for the majority of data traffic.

On the other hand, the following anticipated impact was not substantiated by the measurement results.

- The signature (ECDSA) parameter change results in increased data traffic.
    - We infer that our hypothesis is correct, but the conditions under which this proof of concept was carried out were not ideal.
    - In this proof of concept, at the time of wallet migration we created a single address to which the entire funds were transferred, thereby carrying out migration to the new signature parameter.
    - As a result, each transaction contained only one payer address, making the number of signatures per transaction very small—just one (the average number of signatures per transaction being two).
    - In the real world, multiple addresses are contained in a Bitcoin wallet, from which multiple payer addresses are chosen for each transaction, requiring the corresponding number of signatures.
    - We deduce that the number of signatures per transaction that was incongruent with the real-world situation was the reason why the parameter change did not result in increased data traffic.
    - Another factor was the transaction size. Although we randomly selected transactions for use in measurement, the size of transactions used for measurement turned out to be 5,000 bytes, far larger than the 500 bytes considered to be the average transaction size. This diminished the impact of the increase by the signature parameter change.

6.5.2. Impact on Data Volume

6.5.2.1. Disk Usage per Block

To find disk usage, we measured the following on a node-by-node basis:

- An increase in disk usage from each previous block.
    - Measured using the “du -s --time $DIR” command.

Note, however, that measurement results need to be reviewed in light of the following features of the Bitcoin implementation:

- In the Bitcoin implementation, fairly large disk space is allocated initially.
- Then, when the allocated space becomes insufficient, once again large disk space is secured at one time.
- As large disk space is secured at one time, disk usage spikes occasionally.
- As large undo data is deleted on an irregular basis, disk usage plummets occasionally.

How disk usage increased from each previous block is illustrated in the graphs below, in which the unit of the vertical axis is kilobyte.

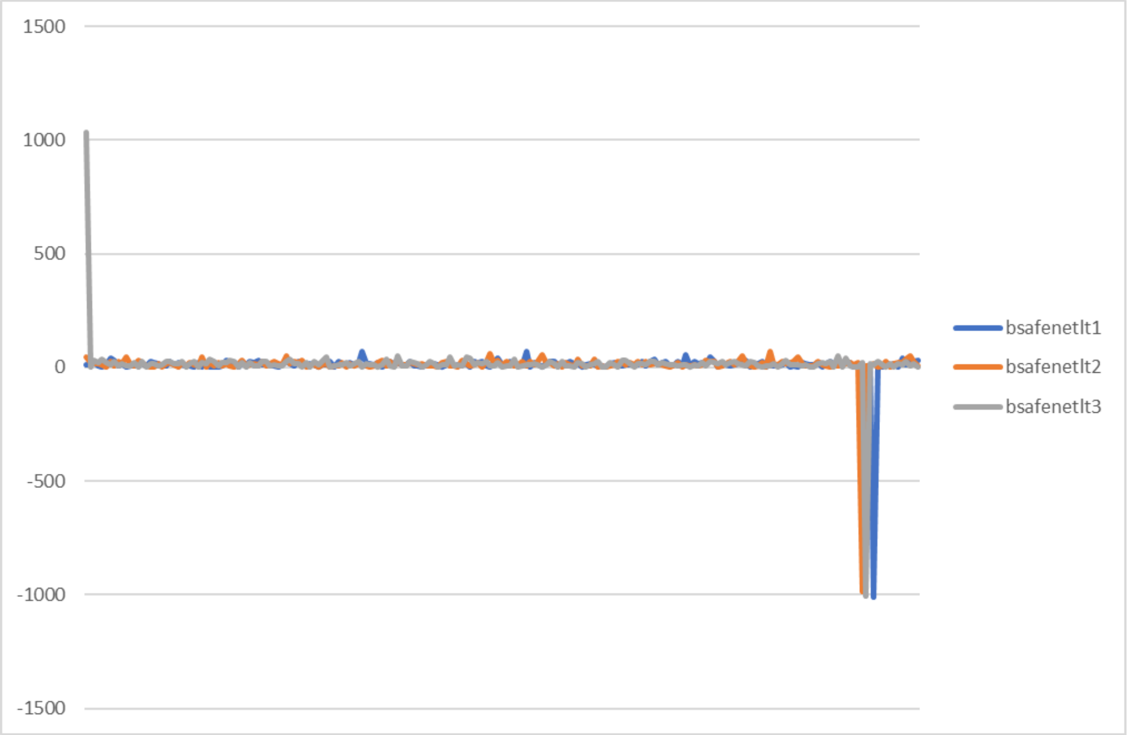


Chart 6-6: Changes in disk usage—Node 1

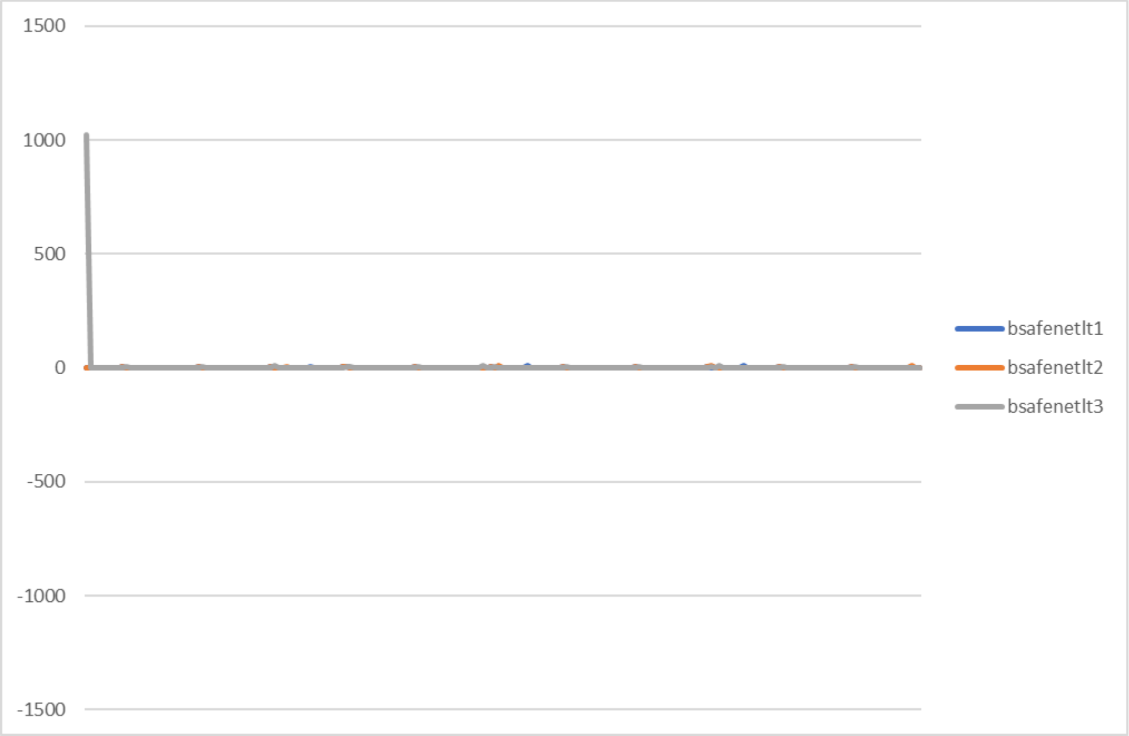


Chart 6-7: Changes in disk usage—Node 2

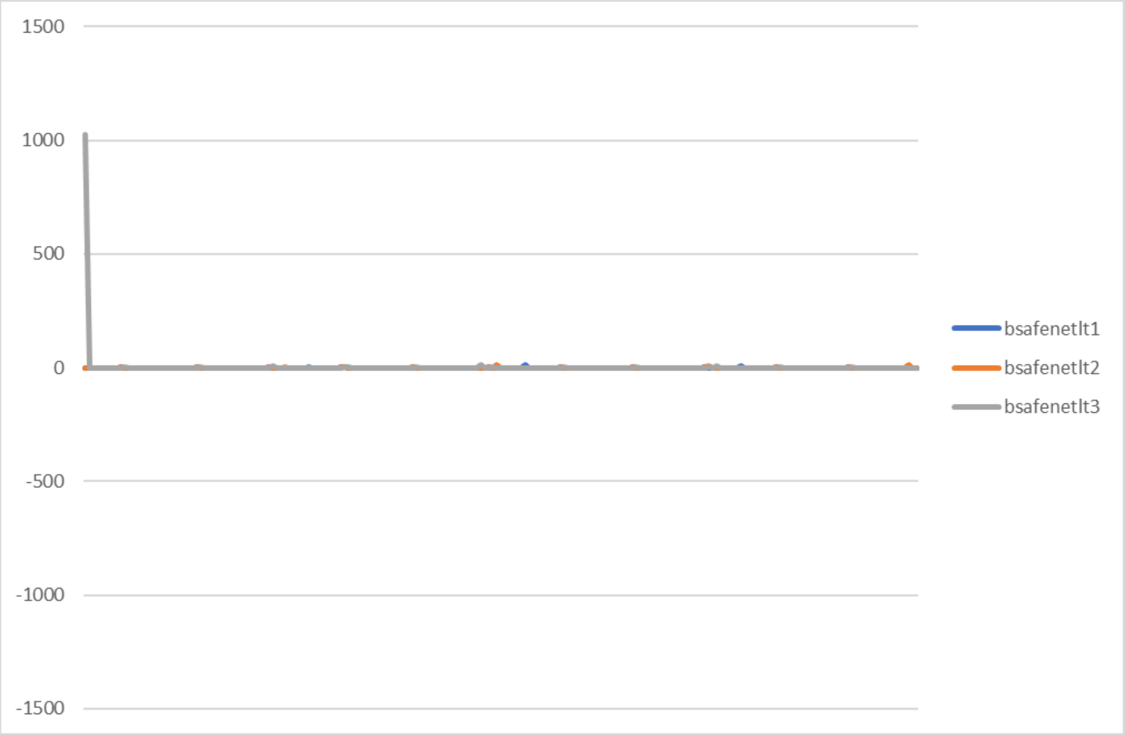


Chart 6-8: Changes in disk usage—Node 3

In Scenarios 1 to 3, no noticeable difference was observed in the disk usage due to the Bitcoin implementation in which fairly large disk space is allocated initially.

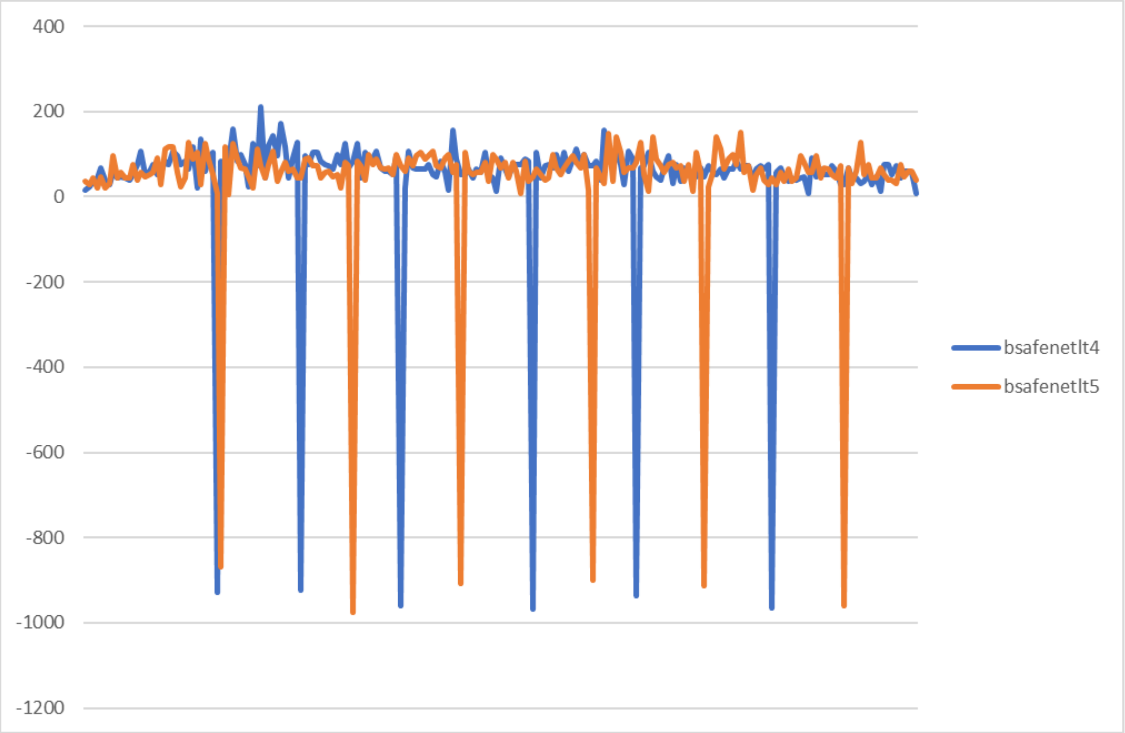


Chart 6-9: Changes in disk usage—Node 1

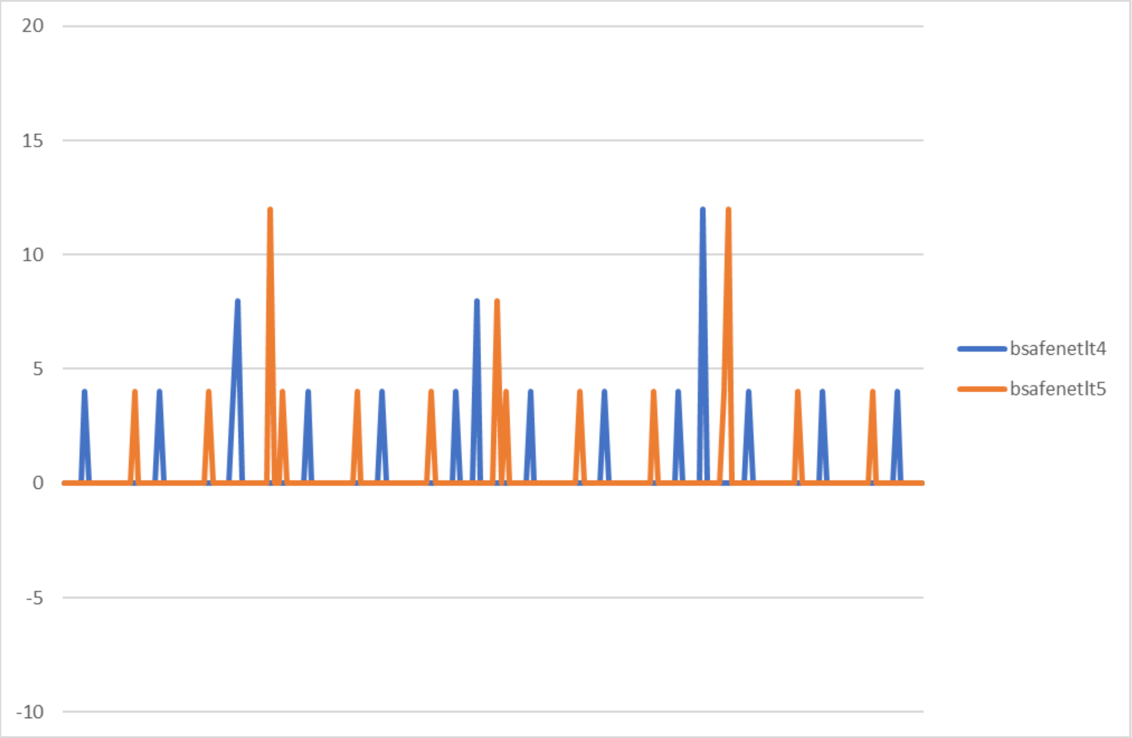


Chart 6-10: Changes in disk usage—Node 2

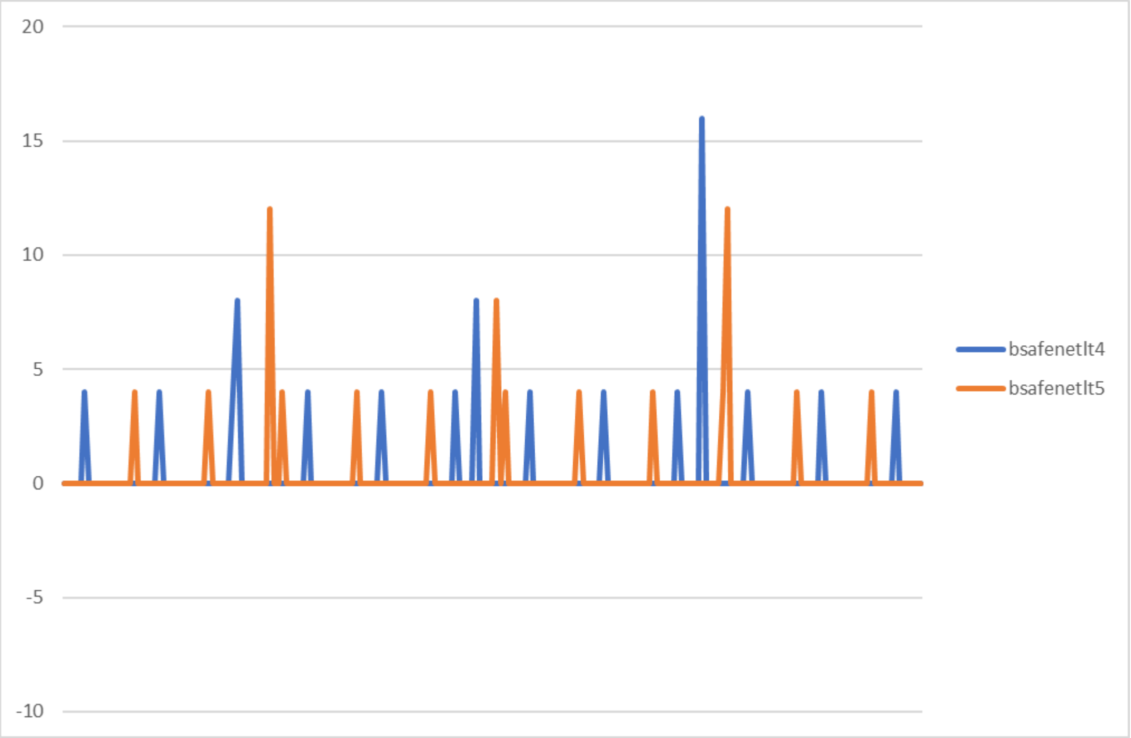


Chart 6-11: Changes in disk usage—Node 3

A comparison between Scenarios 4 and 5 does not also reveal any noticeable difference in changes in disk usage.

6.5.2.2. Size of Individual Blocks

Below are the measurement results concerning block size. The average block size in each scenario is shown in Figures 6-25 and 6-26, in which bsafenetIt1, bsafenetIt2, bsafenetIt3, bsafenetIt4, and bsafenetIt5 correspond to Scenarios 1 to 5, respectively.

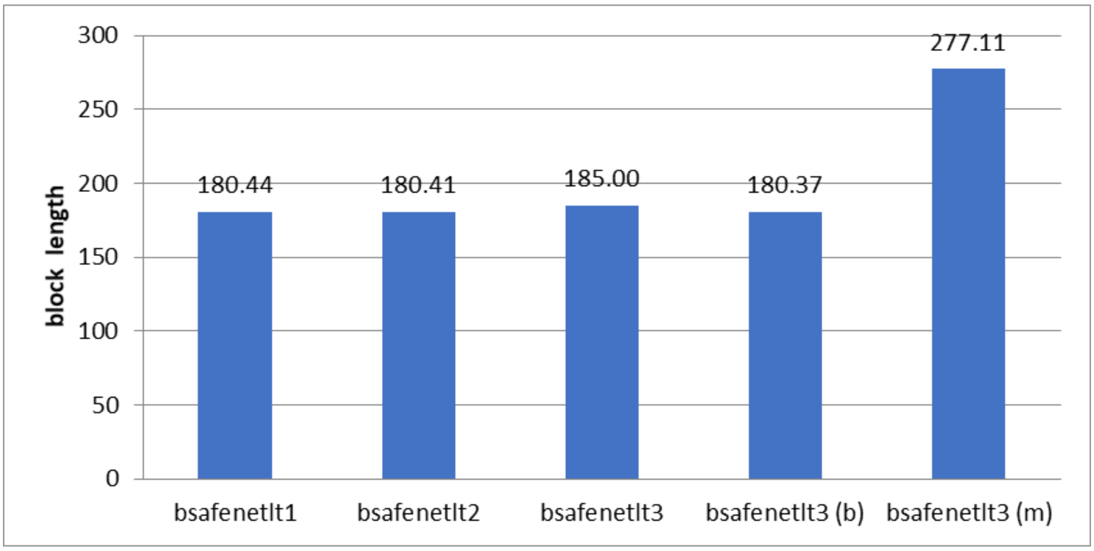


Chart 6-12: Average block size—Scenarios 1 to 3
Note: "(b)" denotes data without, and "(m)" denotes data with, archiveHash.

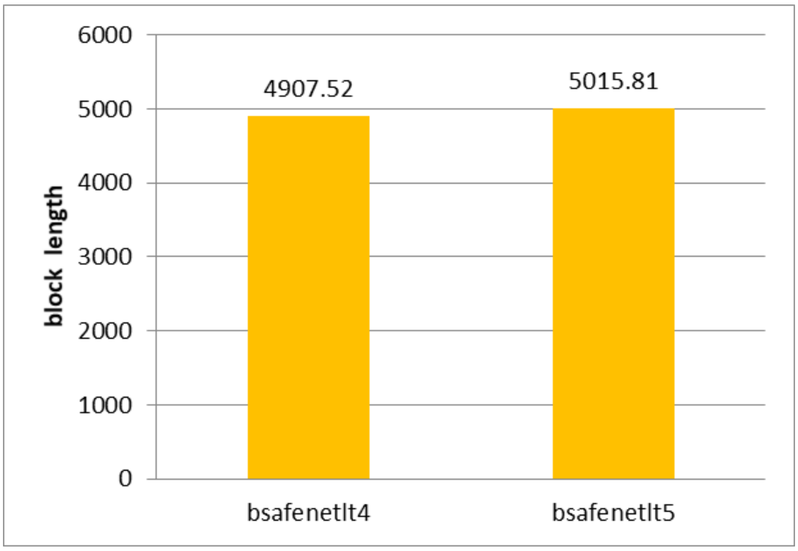


Chart 6-13: Average block size—Scenarios 4 and 5

In Scenarios 1 to 3 (bsafenetlt1, bsafenetlt2, and bsafenetlt3, respectively), the average block size increased by 97 bytes after archiveHash was added. This increase roughly corresponded to the increase in the header length.

A comparison between Scenarios 4 and 5 (bsafenetlt4 and bsafenetlt5, respectively) shows that, in this proof of concept's transaction flow, the average block size increased by around 100 bytes. This was affected by the increase in the signature size resulting from the change of the signature bit length from 256 bits to 384 bits. The increase in the signature size translated into a 48-byte increase per signature (for details, see 3.1. Estimates for Targeted Data Volume).Two signatures were affixed to one transaction in this proof of concept, resulting in an increase of around 100 bytes in total.

The size of a signature itself increased by about 50% from 256 bits to 384 bits. However, due to a signature accounting for only a small proportion of the transaction, the increase in the overall transaction size was limited to about 2% (from 4,907.52 bytes in Scenario 4 to 5,015.81 bytes in Scenario 5).
Figures 6-27 and 6-28 below illustrate changes in block size in each scenario.

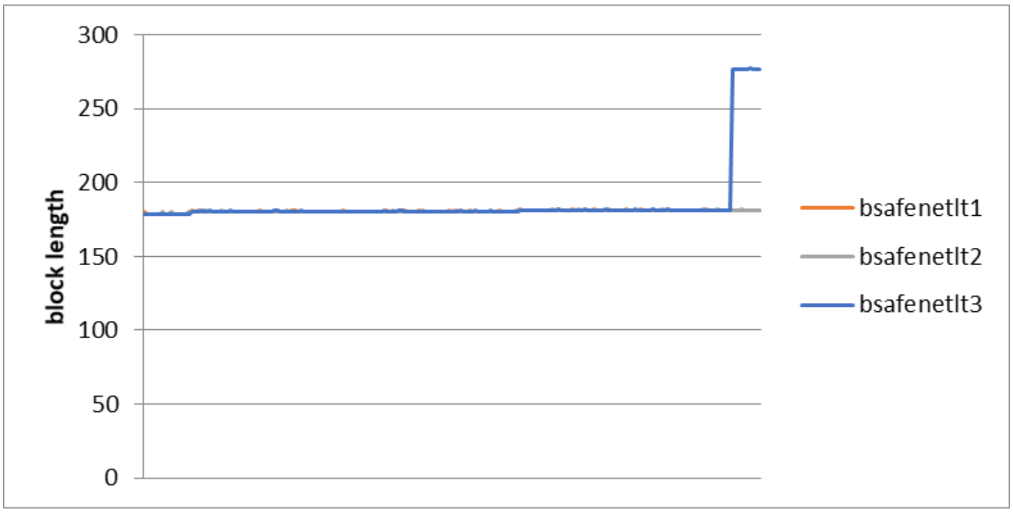


Chart 6-14: Changes in block size—Scenarios 1 to 3

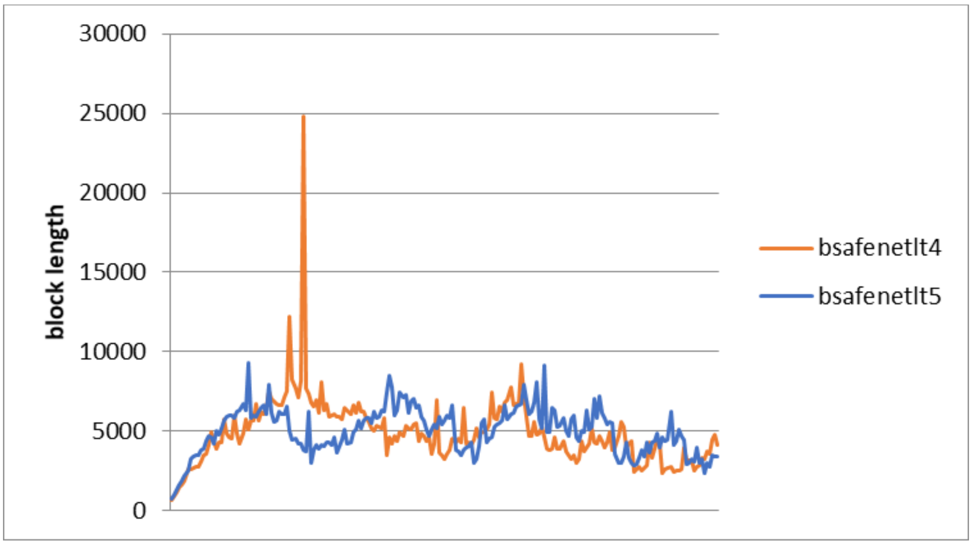


Chart 6-15: Changes in block size—Scenarios 4 and 5

6.5.2.3. Analysis of Impact on Data Volume

The following anticipated impacts were substantiated by the measurement results:

- On a block-by-block basis, the addition of archiveHash causes the block size to increase.
- On a block-by-block basis, the increase caused by the long-term signature scheme (archiveHash) is insignificant.

On the other hand, the following anticipated impacts were not substantiated by the measurement results.

- On a block-by-block basis, the signature (ECDSA) parameter change results in the increased block size.
  - ➢ The impact of the signature (ECDSA) parameter change is considered more substantial. (For details, see 6.5.1.2 Analysis of Impact on Data Traffic.)
- Neither archiveHash nor the signature parameter change has an impact on overall disk usage.
  - ➢ This was because, although a feature of the Bitcoin implementation dictates that large disk space be secured at one time, it was impossible to fully utilize this disk space during the proof of concept period.
- Measuring the impact on disk is likely to require a proof of concept that uses a disk space of about one gigabyte.
  - ➢ This is believed to have been affected by the less than ideal conditions under which the signature (ECDSA) parameter was changed in this proof of concept, similarly to the analysis on a block-by-block basis mentioned above.

6.5.3. Impact on Execution Speed

6.5.3.1. Computation Speed of Hash Functions

We measured the impact of changes of hash functions on computation speed, pertaining to the three hash functions below. Note that what was measured in each case was the time it took to compute the hash value of a block header only, and did not include the time it took to compute a Merkle tree.

- BlockHash
  - ➢ The time to compute the hash value of a block header, using the old proof-of-work hash function (unit: microsecond per block).
- BlockHashNew
  - ➢ The time to compute the hash value of a block header, using the new proof-of-work hash function (unit: microsecond per block).
- BlockHashArchive
  - ➢ The time to compute the hash value of a block header with archiveHash, using the new proof-of-work hash function (unit: microsecond per block).

The measurement results are shown in Table 6-2 below.

Table 6-2: Hash function computation time (unit: microsecond per block)

Hash function	Number of measurement	Minimum	Maximum	Average
BlockHash	1,179,648	0.846	0.983	0.888
BlockHashNew	1,310,720	0.733	0.983	0.809
BlockHashArchive	917,504	0.539	1.222	1.120

These measurement results led us to conclude that the impact of the change of hash functions (from BlockHash to BlockHashNew) on computation speed is insignificant.

As for BlockHashArchive, since its computation takes place only at the time of generating archiveHash and has little influence on hash rate verification, the impact of this hash function on the overall computation of the blockchain’s hash rate is considered insignificant.

6.5.3.2. Computation Speed of archiveHash

We measured the time it took to compute the hash value of a one-megabyte sequence, using the new hash function. This enabled us to measure the computation speed of archiveHash.

The measurement results are shown in Table 6-3 below.

Table 6-3: archiveHash computation time (unit: microsecond per block)

	Number of measurement	Minimum	Maximum	Average
HashArchive	448	1,237.076	2,797.548	2,584.982

6.5.3.3. Computation Speed of ECDSA

We measured the impact on computation speed of the change in the ECDSA key length, pertaining to the generation process of 256- and 384-bit ECDSA signatures.

The measurement results are shown in Table 6-4 below.

Table 6-4: ECDSA computation time (unit: microsecond)

	Number of measurement	Minimum	Maximum	Average
256-bit ECDSA	24,576	21.223	46.720	41.296
384-bit ECDSA	1,792	282.156	598.073	575.776

These results show that the change in the ECDSA key length slowed the signature generation speed down by a factor of about 10. We deduce that the slower speed was largely attributable to the use of an external library as a means of changing the key length (see 6.4.1.3), rather than to the increase of the key length itself.

As generating one signature does not take much time, lighter applications such as payment processing will not become an issue. However, a significant impact is expected on the verification of transaction signatures (ECDSA) at full nodes.

6.5.3.4. Analysis of Impact on Execution Speed

The following anticipated impacts were substantiated by the measurement results:

- Computation speed is not affected by the addition of archiveHash.
- Using the signature algorithm (ECDSA) in an external library has an impact on execution speed.

On the other hand, the following anticipated impact was not substantiated by the measurement results:

- None.