Computers, Materials & Continua
DOI: 10.32604/cmc.2022.029161
Article

Tech Science Press

Trustworthiness Evaluation for Permissioned Blockchain-Enabled Applications

Shi-Cho Cha[1], Chuang-Ming Shiung[1], Wen-Wei Li[1], Chun-Neng Peng[1], Yi-Hsuan Hung[1] and Kuo-Hui Yeh[2,3,*]

[1]Department of Information Management, National Taiwan University of Science and Technology, Taiwan, China
[2]Department of Information Management, National Dong Hwa University, Hualien, Taiwan, China
[3]Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, China
*Corresponding Author: Kuo-Hui Yeh. Email: khyeh@gms.ndhu.edu.tw

Received: 26 February 2022; Accepted: 20 April 2022

Abstract: As permissioned blockchain becomes a common foundation of blockchain-based circumstances for current organizations, related stakeholders need a means to assess the trustworthiness of the applications involved within. It is extremely important to consider the potential impact brought by the Blockchain technology in terms of security and privacy. Therefore, this study proposes a rigorous security risk management framework for permissioned blockchain-enabled applications. The framework divides itself into different implementation domains, i.e., organization security, application security, consensus mechanism security, node management and network security, host security and perimeter security, and simultaneously provides guidelines to control the security risks of permissioned blockchain applications with respect to these security domains. In addition, a case study, including a security testing and risk evaluation on each stack of a specific organization, is demonstrated as an implementation instruction of our proposed risk management framework. According to the best of our knowledge, this study is one of the pioneer researches that provide a means to evaluate the security risks of permissioned blockchain applications from a holistic point of view. If users can trust the applications that adopted this framework, this study can contribute to the adoption of permissioned blockchain-enabled technologies. Furthermore, application providers can use the framework to perform gap analysis on their existing systems and controls and understand the risks of their applications.

Keywords: Permissioned blockchain; blockchain security; blockchain risk evaluation

1 Introduction

Due to the popularity of Bitcoin and other cryptocurrencies built on blockchain technology, blockchain technology is now at the center stage of the world. Several organizations have launched their blockchain applications. However, it is said that the water that bears the boat is the same that swallows it up. When the prices of Bitcoin and other cryptocurrencies crashed in Dec. 2017, the experts were reconsidering the value of blockchain applications [1]. Currently, blockchain application providers may need to convince their clients that they are not going blockchain for blockchain's sake. In addition, researches have demonstrated criteria to decide whether applications are suitable to use blockchain technology. For example, McAbee et al. mentioned some critical factors to determine the adoption of blockchain technology in the military intelligence process [2]. In this study, we refer to a blockchain application as an application found on blockchain networks. A blockchain network is composed of several nodes (or participants). The application can send a request to a node in a blockchain network and delegate the node to execute the request on behalf of the application. The node further propagates the request or execution results to other nodes. Afterward, the nodes achieve consensus on the execution result of the request collaboratively. The blockchain networks can be classified into public blockchains and permissioned blockchains [3]. In a permissioned blockchain network, only permitted nodes can join the network. Comparatively, a public blockchain network has no restriction on who can participate in the network.

This study focuses on the applications that rely on permissioned blockchain networks. If organizations establish applications on a public blockchain network, the application providers or the application providers or users may not be capable of affording the transaction fees in return for rewarding the node owners of the network to process the requests of the applications. Moreover, in the public blockchain networks, as nodes of the network spread around the world, the spreading needs a significant amount of time periods to achieve consensus on the block data. Consequently, in addition to the applications related to cryptocurrency exchanges, organizations usually deploy their blockchain applications based on permissioned blockchain. To avoid a blockchain application from the criticism of blockchain for blockchain's sake, the involved parties of people could dive into the key features of a blockchain network, and they can judge a blockchain application by evaluating whether the applications utilize the features of blockchain technology. From a technical perspective, comparing the blockchain technology with existing technologies such as PKI (Public Key Infrastructure), distributed database, and high availability architecture, this study advocates that a permissioned blockchain network should at least have the following features: (1) having a friendly means for data verification; (2) letting more than one party of authority to keep data replication and to endorse data integrity; (3) being able to tolerate a certain degree of failure.

When a blockchain application claims that it utilizes the above features of its blockchain network, users may be curious about whether the application provider manages its blockchain network properly. For example, a natural disaster may disable a blockchain network if all nodes are located in the same facility. Enabling users to trust that a blockchain application is managed appropriately is particularly important to permissioned blockchain applications. Comparatively, a public blockchain application usually assumes that each node of the associated blockchain is untrustworthy. Therefore, users usually judge the blockchain with its algorithm and number of nodes in the blockchain. For example, in addition to regulation risks and market related risks, Muller et al. propose a framework to evaluate risks of crypto tokens with the underlying technology, such as consensus protocols, cryptographic algorithms, and countermeasures to address cybersecurity attacks [4]. Islam et al. propose to assess the sustainability of blockchain networks on their mining schemes [5]. The number of nodes in a permissioned blockchain network is usually much less than the number of nodes in a public blockchain network. For example, attackers could just control a few nodes in a permissioned blockchain to influence data integrity [6]. Therefore, a security risk management scheme needs to be in place to help permissioned blockchain application providers to estimate the security risks of their applications and adopt measures to control the risks [7].

In light of this, this study proposes a security risk management framework for permissioned blockchain applications. Based on the implementation stacks of blockchain networks, the framework classifies security safeguards to protect permissioned blockchain applications into 6 categories. After collecting current information security practices and guidelines, such as ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) 27001, ISO/IEC 27002, PCI DSS (Payment Card Industry Data Security Standard), and CIS Controls (Critical Security Controls), this study maps and practices to the categories and includes controls specific to permissioned blockchain applications. With the framework, application providers can evaluate the security risks of permissioned blockchain applications by determining whether the applications adopt appropriate controls to protect the applications. Moreover, interested parties can delegate auditors to use the framework to ensure permissioned blockchain applications have implemented the controls. Therefore, the framework can improve the trustworthiness of permissioned blockchain applications. If people can trust the applications that adopted the framework, the paper can hopefully contribute to the adoption of permissioned blockchain technologies.

2 Preliminary

2.1 Implementation Stacks of Blockchain Application

This study follows the blockchain network implementation stack proposed by Wang et al. [8], which is summarized from the model proposed by Duan et al. [9], to provide background knowledge of blockchain applications. As illustrated in Fig. 1, each node in a blockchain network follows the data organization protocol to store data. Simply speaking, blockchain technology organizes data in the form of blocks chained together in sequential order. Each block comprises a block header and block contents. The contents of a block consist of a set of transactions. Each transaction is issued by a person and is signed with the person's private key with digital signature technologies. People can check the authenticity of a transaction with the associated digital signature.

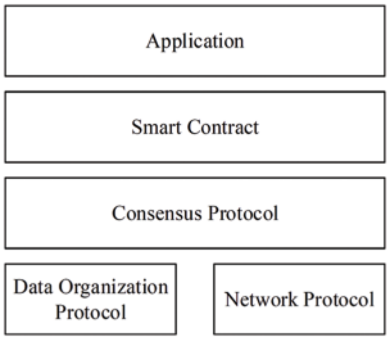

Figure 1: The blockchain network implementation stacks proposed by Wang et al. [8]

To prevent the composition of block contents from being tampered with, a signature is generated from the transactions of a block based on a hash function or other signature generators. In addition to the signature of the block contents, a block header includes a serial number, block generation time, and other block verification information. People usually called the first block in a blockchain as the genesis block. Suppose that the serial number of the genesis block of a blockchain is 0. The serial number of the second block of the blockchain is 1 and so on. Therefore, the blocks are chained logically with the serial numbers. To protect data integrity, the values of the block header are hashed. Note that except for the genesis block, each block includes the hash value of the previous block in its block header.

Therefore, if a malicious intent person wishes to modify a transaction in i-th block in a blockchain, the block owner may need to update the hash value of the block's block header. Then, modification of the headers of the (i + 1)-th block to the latest block is necessary. Furthermore, current blockchain technologies may request block generators to solve some kinds of cryptographic puzzles based on the values of the block header. A malicious person may need a huge amount of computing power to tamper with block data.

The network protocol maintains the connectivity of nodes in a blockchain network. Upon receiving a transaction, a node propagates the transaction to other nodes. Hence, nodes in a blockchain network can validate the transaction collaboratively. A consensus is achieved by the nodes on the block data with the consensus protocol. The consensus protocol can further be divided into three sub-processes. First, in the block generation sub-process, it is assumed that nodes in a blockchain network have achieved consensus on the first i - 1 blocks. One or more nodes generate the candidate i-th block based on transactions that have not been encapsulated in existing blocks. Second, in the agreement achievement sub-process, the nodes elect one block from the candidates as the i-th block and keep the data in their local storage. Third, if a node finds that its block data are different from others' data, the node may initiate the conflict resolution sub-process to ensure data consistency.

Blockchain technology was first introduced to operate with cryptocurrencies like Bitcoin. The smart contract technology enables people to enforce a blockchain network to execute autonomous computer programs [10,11]. Simply speaking, users can deploy programs (or smart contracts) and initial state valuables as transaction data in blockchain networks. Nodes in a blockchain network may launch virtual machines to execute the smart contracts. When a user sends a request to a smart contract, the virtual machine in a node fetches the instructions of the smart contract along with current values of state valuables of the smart contract from the blockchain network. The virtual machine then executes the smart contract and stores the execution results as transaction data in the blockchain network. Consequently, any nodes equipped with the virtual machines can extract the smart contracts and associate versions of state variables and re-execute the smart contract to verify the execution results. Finally, current blockchain networks usually have their own APIs (Application Programming Interfaces). People can develop applications to send requests to the blockchain networks via the APIs.

2.2 Major Characteristics of a Blockchain Network

This study compares blockchain technologies with similar technologies to identify the major characteristics of a blockchain network in this sub-section. First, a blockchain network can be viewed as a special purposed distributed database. As described in previous subsection, a blockchain network has a linked list structure like block data. In addition, digests are embedded in the blocks by design for tamper proof. Second, users may challenge the blockchain technology as they can simply use digital signature technologies to ensure data integrity. In this case, the blockchain technology also adopts the digital signature technologies. Moreover, the blockchain technology spreads signed data around the blockchain networks. As the nodes of blockchain networks are managed by different parties, the nodes can enhance data integrity collaboratively. For example, when a person shows logs with digital signatures, we can only make sure that the logs are signed by the same person. However, the person may change the time in the logs and sign the logs with his or her private key. The person can therefore pretend that he or she has done something to the logs. In this case, if the logs are stored in a blockchain network as each log entry is generated, the person may need to collude with majority node managers to replace block data in the managers' nodes. Obviously, it is more difficult for people to counterfeit past log entries and to store the log entries in the blockchain as time goes by.

As the blockchain networks replicate block data on nodes managed by different parties for tamper-proof, the users may still challenge that they can manage data in a server and request the server to publish data signatures periodically. Accordingly, interested people can keep the published data and use the data to discover data manipulation. However, the centralized server may become a single point of failure. When the centralized server is attacked or crashed, nobody can access the data. Comparatively, the application based on a blockchain network may survive when one or more nodes are unavailable. To sum up, people usually adopt blockchain networks to achieve the following features:

- *Verifiable block relationship:* Blocks in a blockchain are ordered. Also, each block includes a block digest. Except for the genesis block, the digest of a block is generated with the digest of its previous block. Manipulation of a block will change its digest and broke the relationship with its next block. The feature make a person validate block data in a blockchain more easily. Moreover, as the "order" of blocks "is" guaranteed, people can assert that a transaction in a block is earlier than transactions in the next block.
- *Relying participants to achieve data immutability:* A blockchain network should have more than one participating node. A node can check data received from other nodes and discover the abnormal. Therefore, the node can take appropriate steps to eliminate the abnormal data for data immutability.
- *Sustainable under partial failures:* As a blockchain network is a distributed system per se, a blockchain should tolerate partial failures. In addition, even a participating node crashes or meets network-partition failures. Once the node makes connection to other healthy nodes, the node can obtain necessary data from other healthy nodes and recover to the correct state.

A blockchain network should enable people to trust that it satisfies the above features. As anybody can join a public blockchain network, people usually cannot evaluate the trustworthiness of a public network chain based on its participants. In this case, people usually can just evaluate the consensus algorithm of a public blockchain network to decide whether the blockchain can achieve the above features. As a reminder, this research focuses on the permissioned blockchain applications. In addition to evaluating the consensus algorithm of a permissioned blockchain network, the framework proposed in this study request people to assess the security risks of participants in the blockchain to determine the trustworthiness of the blockchain.

2.3 Model for Permissioned Blockchain Applications

This study specifies the model of a permissioned blockchain application in this article. As depicted in Fig. 2, a permissioned blockchain application is built on a permissioned blockchain network. And a permissioned blockchain network is maintained by one or more specified blockchain network providers. Note that an organization may have two blockchain network providers if the organization ensures members of the two blockchain network providers are independent.

Each blockchain network provider contributes one or more participating nodes to the blockchain network. A participating node is executed on a computing resource, such as a computer, a VM (Virtual Machine) instance, etc., and is administrated by a network provider. A participating node stores block data and achieve consensus on the block data with other participating nodes. Also, a participating node may include identity and access management information to authenticate the user or the application service that sends requests to the node and determine the privileges of the user. Furthermore, the blockchain network may have the capacity to handle smart contracts. Therefore, the nodes achieve consensus on block data and the states of smart contracts.

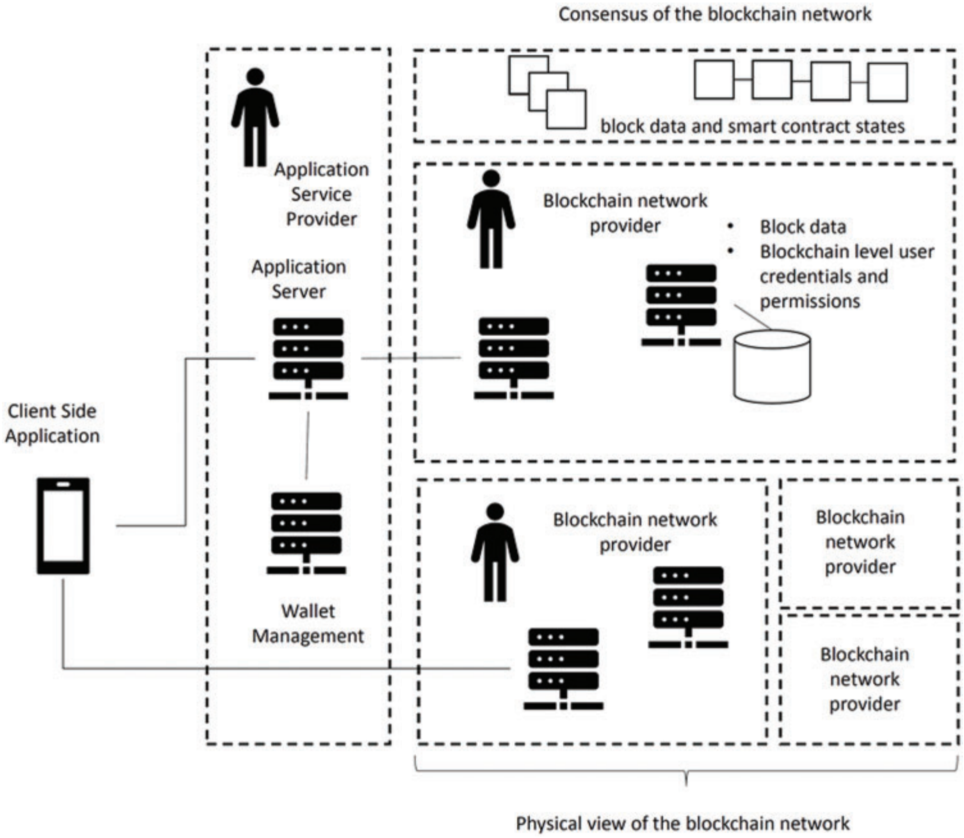
Consensus of the blockchain network
block data and smart contract states
Application Service Provider
Application Server
Blockchain network provider
Block data
Blockchain level user credentials and permissions
Client Side Application
Wallet Management
Blockchain network provider
Blockchain network provider
Blockchain network provider
Physical view of the blockchain network

Figure 2: Model of a permissioned blockchain application in this study

The application service provider may develop and deploy smart contracts of the applications in the blockchain network. In addition, the application service provider may also deploy instances of application services. An application service instance connects to a participating node in the blockchain network and sends transactions to the blockchain network via the node. On the other hand, the application service provider may provide client-side applications to users as well. Therefore, users can use the client-side applications to access the application service or send requests to the blockchain network directly. Finally, the users may delegate the application service provider to manage their wallets of the blockchain network.

3 The Proposed Framework

3.1 Overview

The proposed framework provides security controls in different implementation stacks of permissioned blockchain applications. As depicted in Fig. 3, this study groups the controls for different implementation stacks into categories. First, the proposed framework requests permissioned blockchain application services and blockchain network providers (or simply application providers) to deploy perimeter security controls as the first line of defense. Permissioned blockchain application providers should identify resources, such as hardware, software, and data, which are used for the permissioned blockchain application's logical and physical means to access the resources. The providers can then define protected areas based on the location of the resources. As a result, providers could deploy physical and logical checkpoints to prevent unauthorized people from accessing the protected areas and controlling the flow of the resources. Second, the host security category seeks for the necessary defensive protection measure of the operating environment, including not just the physical operating device but also the status quo of the blockchain data within the permissioned chain. Access control plays a crucial role in this control, both in the pairing of keys and the monitoring of authorized access between the nodes and permissioned blockchain application providers. Third, in the node management and network security category, the behaviors of nodes are monitored, and joint-decision organization is involved to help maintain the control of the nodes. Fourth, the consensus mechanism security category includes controls to request the application providers to provide information about the permissioned blockchain. In addition to the provided information to prove the validity of the consensus mechanism, the application providers should provide other information to help with evaluating fault tolerance ability of target blockchains, and they should know how the application ensures privacy and data confidentiality. Fifth, the application security category requests the application system security to help reduce security breaches based on faulty application systems. By code inspection, security awareness, and security testing, this control mainly seeks to identify potential major threats and prevent security breaches. Finally, the organizational security category asks application providers to define procedures to enforce the security of their applications.

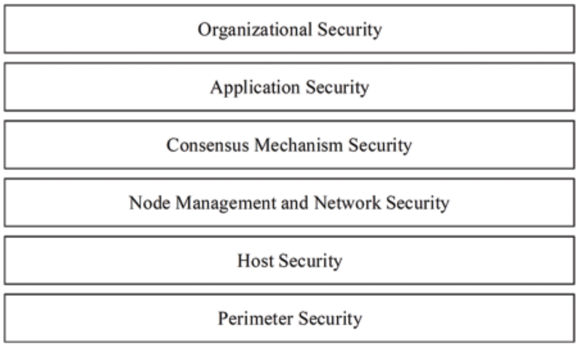

Figure 3: Major components of the proposed framework

3.2 Perimeter Security (PS)

Similar to the NIST (National Institute of Standards and Technology) cybersecurity framework [12], this study maps the controls with existing security guidelines, including PCI-DSS v3.2.1, CIS Controls v7.1, ISO/IEC 27001:2013, and ISO/IEC 27002:2013. As listed in Tab. 1, this study provides controls with associated guidelines as references. The main objective of this category is to request the permissioned blockchain application providers to ensure perimeter security. The category includes two controls:

- Application providers should deploy logical perimeter security mechanisms, such as firewalls and other network access control mechanisms, to prevent unauthorized users from accessing protected resources (Control PS-1).
- Application providers should prevent unauthorized people from entering protected areas and prevent protected resources from being taken out of protection without permission (Control PS-2). As illustrated in Tab. 1, application providers can apply existing guidelines or best practices to implement the controls.

Table 1: List of controls in perimeter security category

ID	Control	Informative references
PS-1	Logical perimeter security	PCI-DSS 3.2.1 Requirement 1 CIS Controls v7.1 2.10 9 12 15 ISO/IEC 27002:2013 13.1.2 13.1.3
PS-2	Physical perimeter security	PCI-DSS 3.2.1 9.1 9.2 9.3 9.4 9.6 9.8 CIS Controls v7.1 2.10 ISO/IEC 27002:2013 8.3 11.1 11.2.5 11.2.7

3.3 Host Security (HS)

The Host Security category requests application providers to adopt appropriate host-level safeguards to protect participating nodes and associated components. As listed in Tab. 2, the category includes the following controls:

- Application providers should implement common security protection mechanisms, such as antivirus software, software firewall, backups, identity management, access control, etc., on participating nodes and associated devices (Control HS-1). Moreover, application providers can deploy a host-based monitoring scheme to log and identify malicious behaviors.
- Application providers should protect block storage from being tampered with or unauthorized access (Control HS-2). The control is specific in blockchain applications. Application providers should identify the location of block data (including backups of the data) and adopt safeguards to protect the data. Note that each participant could have credentials for node identification and settings of permissioned nodes. Application providers should identify such sensitive data and ensure data protection.
- Application providers should adopt an appropriate cryptographic algorithm and key management scheme (Control HS-3). As blockchain applications are triggered by users' private keys or wallets, blockchain applications should provide security mechanisms to help users to protect their credentials. In addition, to prevent the users from losing their keys, users may delegate application providers to manage their wallets. In this case, application providers could use HSMs (Hardware Security Modules), MPC (Multi-party Computing), and other advanced security protection mechanisms to protect the wallets.
- Application providers should physically protect the participating nodes and associated components (Control HS-4).

Table 2: List of controls in host security category

ID	Control	Informative references
HS-1	Protect hosts and device security	PCI-DSS 3.2.1 Requirement 2 Requirement 5 Requirement 8 CIS Controls v7.1 1 2 3 4 5 8 9 14

(Continued)

Table 2: Continued

ID	Control	Informative references
		ISO/IEC 27002:2013 9.2.1 9.2.3 9.2.4 9.4.4 12.2 12.4 12.5.1 12.6
HS-2	Protect block data and node credentials	PCI-DSS 3.2.1 3.7 CIS Controls v7.1 13 14 ISO/IEC 27002:2013 9.1.1
HS-3	Protect user credentials	PCI-DSS 3.2.1 3.5 3.6 CIS Controls v7.1 10.4 13.1 14.4 14.8 16.4 16.5 18.5 ISO/IEC 27002:2013 9.2.4 10.1.1 10.1.2 11.2.7
HS-4	Guard the physical and environmental security	PCI-DSS 3.2.1 9.2 9.3 9.4 9.5 9.6 9.7 9.8 9.10 CIS Controls v7.1 10.4 ISO/IEC 27002:2013 6.2.1 8.3 11.1.3 11.1.4 11.1.5 11.2.1 11.2.2 11.2.3 11.2.4 11.2.5 11.2.7

3.4 Node Management and Network Security (NMNS)

In a permissioned blockchain network, only permitted nodes can join the network. Also, each node should only perform allowed operations. In the node management and network security category, application providers should deploy network-level controls to enforce the permission settings in their permissioned blockchain networks. As listed in Tab. 3, the category includes the following controls:

- Application providers should implement common security protection mechanisms, such as antivirus software, software firewall, backups, identity management, access control, etc., on participating nodes and associated devices (Control NMNS-1). Moreover, application providers can deploy a host-based monitoring scheme to log and identify malicious behaviors. Application providers should maintain lists of nodes in the permissioned blockchain network and prevent unauthorized nodes from joining the network. In addition, application providers should administrate permissions of the nodes appropriately. If a node performs malicious or unauthorized operations, the application providers should be capable of removing the node from their permissioned network.
- Application providers should monitor nodes or hosts in their permissioned blockchain applications to detect anomalies (Control NMNS-2). Compared to the control of logical perimeter security (Control PS-1), the control focuses on the unauthorized or suspicious operations of permitted nodes and the associated components.

Table 3: List of controls in node management and network security category

ID	Control	Informative References
NMNS-1	Participant management	PCI-DSS 3.2.1 1.3.2 1.3.3 12.3.6 CIS Controls v7.1 13.3
NMNS-2	Monitoring abnormal behavior	PCI-DSS 3.2.1 10.1 10.2 10.3 10.4 10.5 10.6 10.7 10.8 10.9 11.4 CIS Controls v7.1 12 ISO/IEC 27002:2013 12.4 13.1.2

3.5 Consensus Mechanism Security (CMS)

The consensus mechanism is one of the most important components in a blockchain network. The consensus mechanism security category includes a set of controls to enable users to trust the consensus mechanisms and to enable application providers to enhance security of their consensus mechanism. As consensus mechanism is a specific characteristic in blockchain networks, Tab. 4 does not have references associated with controls in the category.

- Application providers should identify the consensus algorithms used by their blockchain network and ensure the correctness of the algorithm (Control CMS-1). Researchers usually evaluate the correctness of a distributed consensus algorithm with whether the algorithm satisfies the agreement and validation requirements [13]. In terms of agreements, it is proved that blockchain algorithms under certain degrees of viable error, acceptance can still ensure that all nodes can reach a consensus for a transaction in the end. In terms of validation, it is ensured that all nodes make the same acknowledgment for a transaction then the consensus is the validation of this transaction.
- Application providers should determine their Byzantine fault tolerance abilities (Control CMS-2). A distributed consensus algorithm usually can tolerate a certain degree of Byzantine fault. For example, in the traditional Byzantine algorithm, if a blockchain network needs to tolerate m malicious nodes, the blockchain network should have at least $3m + 1$ participating nodes. The application providers should determine their target Byzantine fault tolerance abilities and deploy their blockchain networks based on the target.
- Application providers should identify the minimum resources required for maintaining their blockchain network operation (Control CMS-3). With the minimum resources requirement information, application providers can use their participating node deployment status to estimate the availability of their blockchain networks. The application providers can then re-arrange the nodes for better availability. Moreover, application providers can establish their business continuity management systems based on the information.
- Application providers should implement mechanisms and associated procedures to handle incidents of the consensus mechanism (Control CMS-4). For example, if a malicious node in a blockchain network sends a useless number of transactions to the blockchain network, other nodes may waste storage on storing the transactions. Therefore, the blockchain network can generate a fork to clean the useless transactions.
- Application providers should apply the rule of segregation of duties and delegate the tasks of node management to independent parties (Control CMS-5). That is, a blockchain network

should be maintained by independent blockchain network providers. Also, a blockchain network provider should not control nodes which numbers are more than a certain degree in the blockchain network.

- If necessary, application providers disclose how their blockchain protects privacy and transaction confidentiality (Control CMS-6). Although blockchain technologies can protect data integrity and availability, they do not ensure data confidentiality and privacy [14]. For example, if a person submits a transaction to a blockchain, everybody who can access the blockchain can obtain the transaction content and identify the participants of the transaction. Even though people usually use pseudo-identities, which do not contain personally identifiable information, in a blockchain, researchers such as Biryukov et al. [15] have proposed IP (Internet Protocol) traffic analysis schemes to identify the IP addresses of transaction generators and to link these address to the pseudo-identities. To date, several pieces of researches have been dedicated to enhancing data confidentiality and privacy of blockchain systems. Therefore, if a permissioned blockchain network application has a privacy or data confidentiality requirement, the application provider should disclose how the application achieves the requirement.

Table 4: List of controls in consensus mechanism security category

ID	Control	Informative references
CMS-1	Consensus algorithm verification	NA
CMS-2	Determine the byzantine fault-tolerant capability	NA
CMS-3	Identify the minimum resource requirement	NA
CMS-4	Deal with incidents about the consensus mechanism	NA
CMS-5	Segregation of duties	NA
CMS-6	Disclose confidentiality and privacy protection mechanism	NA

3.6 Application Security (AS)

The application security category introduces controls to secure the application development cycle. In this case, several SSDLCs (Software Security Development Life Cycles) guidelines exist [16]. This study first adopts the touchpoints proposed by McGraw as candidate controls in this category. Among the seven touchpoints, this study moves the controls of risk analysis, penetrating testing, and security operations to the information security management policies category. Moreover, this study adopts the control of protecting development environment security mentioned in ISO/IEC 27001 and PCI-DSS in this category. Also, as users may not understand the concept of blockchain technologies, this study adds user notification control to reduce disputes between users and application providers. Finally, this study adopts the security update management control in IEC 62443-4-1. Note that application providers may outsource application development. The application providers should request the outsourced parties to adopt the controls. As listed in Tab. 5, the category includes the following controls:

- Application providers should perform threat modeling on their applications to identify risks to the applications (Control AS-1). In this case, we can follow the DFD (Data Flow Diagram)-based scheme to model the application and identify the potential attacks based on the STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) threat model [17,18]. For blockchain applications, Mallah and Farooq propose to evaluate the impact of potential attacks based on monetary loss, privacy, data integrity, and trust [19].

- Application providers should define security requirements before developing their applications or making modifications to their applications (Control AS-2). Simply speaking, the control AS-1 requires application providers to identify the threats that their applications should defend against with. Comparatively, control AS-2 requires application providers to identify the security functions the application should have. Application providers can reference the literature of security requirements on blockchain applications, such as the security and privacy requirements proposed by Zhang et al. [20], to define their security requirement.
- Application providers should notify the risks of using the applications and protection guidelines to users (Control AS-3). The control can be viewed as a special case of control AS-2. This study stresses the control because users may suffer from cryptocurrency fever and do not know the risks of using the blockchain applications.
- Application providers should implement manually reviewing the codes of applications or use static and dynamic tools to analyze the code to discover security weaknesses in the applications (Control AS-4). For permissioned blockchain applications, application providers should at least analyze codes of the following components: (1) smart contracts executed in the blockchain networks; (2) server-side programs (usually Web-based applications) used to receive user requests and to negotiate with the blockchain networks; (3) client-side applications for users to send requests to the server-side programs or the blockchain networks.
- Application providers should perform tests on the applications before the applications are released (Control AS-5). In the systems development life cycle, the security testing procedures can be separated into three stages: system development stage, system testing stage, acceptance stage, and deployment stage. In the system development stage, programmers should perform unit testing on their programs, and perform integration testing with different unit components, which can even be integrated with DevOps (Development and Operations) tools. In the system testing or acceptance stage, complete system testing should be conducted in the testing environment from vulnerability scanning to penetration testing. After the applications are online, application providers should perform vulnerability scans and penetration tests regularly on the operating environment.
- Application providers should provide a secured environment for application development (Control AS-6). Also, the application providers should separate the development, test, and production.
- Application providers should establish appropriate update management mechanisms (Control AS-7). Therefore, as the application providers discover vulnerabilities on their applications, they can update affected components to fix the vulnerabilities.

Table 5: List of controls in application security category

ID	Control	Informative references
AS-1	Threat modeling	PCI-DSS 3.2.1 6.1 12.2 ISO/IEC 27002:2013 14.1.1 14.2.5
AS-2	Identify security requirements	PCI-DSS 3.2.1 6.5

(Continued)

Table 5: Continued

ID	Control	Informative references
		CIS Controls v7.1 18.1 18.2 18.3 18.4 18.5 18.10 18.11 ISO/IEC 27002:2013 14.1 14.2.2
AS-3	User notification	ISO/IEC 27002:2013 9.3
AS-4	Code review	PCI-DSS 3.2.1 6.5 6.6 CIS Controls v7.1 18.7 ISO/IEC 27002:2013 14.2.1
AS-5	Perform security test	PCI-DSS 3.2.1 6.4 CIS Controls v7.1 20.5 ISO/IEC 27002:2013 14.2.1 14.2.2 14.2.3 14.2.4 14.2.7 14.2.8 14.2.9 14.3.1
AS-6	Ensure development environment security	PCI-DSS 3.2.1 6.4.1 6.4.2 6.4.4 CIS Controls v7.1 18.9 ISO/IEC 27002:2013 12.1.4
AS-7	Security update management	PCI-DSS 3.2.1 2.2 6.1 6.2 CIS Controls v7.1 18.3 18.4 18.8 ISO/IEC 27002:2013 6.1.4 12.6.1

3.7 Organizational Security (OS)

The concept of organizational security category is originated from the organizational controls in CIS Control v7.1 and the requirement 12 of PCI-DSS 3.2.1. It is worth noting that control OS-5 requests each permissioned blockchain application should establish a joint-decision making organization. The joint-decision making organization is formed by the participating parties collaboratively and defines procedures to achieve consensus on application operation. For example, the organization can request participating parties to generate forks to remove the tampered data generated from vulnerable smart contracts. With considering the legal requirements (Control OS-9), the joint decision-making organization of a permissioned blockchain application can gather the participating parties to define information security policies and procedures to request the participating parties to establish their information security management systems based on the policies:

- First, participating parties of a permissioned blockchain application should establish and maintain documented procedures of information security (Control OS-1). Documenting information security policies and procedures can reduce ambiguity among associated people and provide the foundation for continuous improvement.
- Second, the joint-decision-making organization of a permissioned blockchain application can request each participating party to designate an information security officer responsible for information security-related matters (Control OS-6). The security officer should have competent capabilities and authorities to ensure the enforcement of the information security in his/her party. In addition, a participating party should assign appropriate information security roles and responsibilities to its members (Control OS-2). Also, a participating party should

establish information security awareness, training, and education programs to make sure its members are capable of withholding their security responsibilities.

- Based on existing information security best practices and guidelines, this study selects some necessary procedures that the participant parties of a permissioned blockchain application should establish: (1) risk assessment and management procedures (Control OS-4); (2) change and release management procedures (Control OS-7); (3) incident management and business continuity procedures (Control OS-8). Interested people can see the associated standards listed in Tab. 6.
- To complete the PDCA (Plan-Do-Check-Act) cycle, participating parties of a permissioned blockchain application should perform vulnerability scanning, penetrating testing, and even social engineering testing regularly to discover deficiencies of the parties (Control OS-11). Also, the parties can execute self-checking or build an internal audit program to ensure compliance with the security policies and procedures (Control OS-10). Finally, participating parties of a permissioned blockchain application should learn from past incidents or deficiencies and improve their information management system continuously (Control OS-12).

Table 6: List of controls in organization security category

ID	Control	Informative references
OS-1	Establish and maintain documented procedures	PCI-DSS 3.2.1 12.1 12.8 CIS Controls v7.1 5.1 5.2 6.2 ISO/IEC 27002:2013 5.1.1 15.1.1
OS-2	Security roles and responsibilities	PCI-DSS 3.2.1 12.1 12.4 12.8.2 CIS Controls v7.1 18.3 18.4 18.8 ISO/IEC 27001:2013 5.3 ISO/IEC 27002:2013 6.1.1 7.1.2 7.2.1 7.2.2 9.3
OS-3	Information security awareness, training, and education	PCI-DSS 3.2.1 6.5 9.9.3 12.6 12.10.4 CIS Controls v7.1 17 ISO/IEC 27001:2013 7.2 7.3 ISO/IEC 27002:2013 7.2.2
OS-4	Risk management	PCI-DSS 3.2.1 12.2 ISO/IEC 27001:2013 6.1
OS-5	Joint decision-making	NA
OS-6	Designate an information security officer	PCI-DSS 3.2.1 12.4.1 12.5 CIS Controls v7.1 5.3

(Continued)

Table 6: Continued

ID	Control	Informative references
		ISO/IEC 27002:2013 6.1.1 7.2.1
OS-7	Change and release management	PCI-DSS 3.2.1 1.1.1 6.3.2 6.4 6.6 12.11 ISO/IEC 27001:2013 8.1 ISO/IEC 27002:2013 12.1.2 14.2.2 14.2.3 14.2.4 15.2.2
OS-8	Incident and business continuity management	PCI-DSS 3.2.1 9.5.1 11.1.2 12.5.3 12.10 CIS Controls v7.1 17.9 19 ISO/IEC 27002:2013 16 17
OS-9	Legal compliance	PCI-DSS 3.2.1 3.1 9.8 12.10.1 CIS Controls v7.1 4.2 ISO/IEC 27002:2013 18.1
OS-10	Internal auditing and self-check	PCI-DSS 3.2.1 9.1 9.2 ISO/IEC 27002:2013 12.7.1 15.2.1 18.2.1 18.2.2 18.2.3
OS-11	Penetration testing and vulnerability scanning	PCI-DSS 3.2.1 6.1 6.6 11.2 11.3.1 11.3.2 11.3.3 11.3.4 CIS Controls v7.1 3.1 3.2 3.6 15.2 20 ISO/IEC 27002:2013 18.2.3
OS-12	Continuous improvement	PCI-DSS 3.2.1 11.3.3 12.10.6 CIS Controls v7.1 9.3 10.1 10.2 ISO/IEC 27002:2013 16.1.6 18.2.1 18.2.2

4 Validation of the Proposed Framework

In this section, we demonstrate a case study to evaluate the validation of our proposed framework. In our case study, we first conduct a security check list and then utilize it as a risk evaluation of a specific organization. During the evaluation, the rate of attainment of domains 1 to 9, presented on the conducted security check list as shown in Tab. 7, are evaluated through the security controls involved with. After the evaluation, as shown in Tab. 8, the risk facing and the probability that services being unexpectedly interrupted (or terminated) are conducted in our experiment, respectively, based on the average percentage value as the rate of attainment of each security domain. The experiment environment is based on permissioned blockchain with PBFT (Practical Byzantine Fault Tolerance) algorithm in which the maximum tolerance of crashed/malicious nodes is one third. The evaluation criteria are chain-availability and chain-integrity. The violation of chain-availability/is identified as more than one third nodes of the experiment environment are interrupted or terminated in a unexpected manner and thus the PBFT algorithm is not workable. On the other hand, the chain-integrity is confirmed if, given a specific data block after the consensus of all nodes, more than two third data blocks maintained at their own nodes are not compromised.

Table 7: A security check list of a case study with permissioned blockchain applications

Control domain	Control objective	Security control	Rate of attainment
Virtual boundary security	The objective is in order to manage and control the network, restrict external or unauthorized network access in virtual security zone defined by organization.	Define the network security boundary and set up firewall and IDS in network security boundary. Regularly check and properly manage the configuration of firewall and IDS Implementation of monitoring and recording of network for potential unsecure behavior. Management of wireless network access permissions.	90%
Physical boundary security	The objective is in order to manage and control the area, restrict external personnel or unauthorized physical access in physical security zone defined by organization.	Define the physical security boundary and establish the personnel access management system. Ensure that the physical boundary has sufficient physical protection against unauthorized entry and exit. Document personnel entry record and ensure the devices they bring in and out based on the requirement. Regularly review and update the access permissions of physical boundary Ensure that the confidential information is removed before critical blockchain system equipment is disposal or removed.	80%
Host security of private blockchain node	The objective is in order to protect the security of the private blockchain infrastructure, which ensures the availability of blockchain system machines.	Manage user accounts for registration and cancellation and appoint appropriate permissions. If the system has a privileged account, it is necessary to regularly check whether the user's qualifications are met to prevent the authority against unauthorized usage. Install anti-virus software on the host and regularly update and establish incident response procedures for malware attacks. Establish recovery mechanism to ensure the data can be recovered when the data is lost or invalid Identify the vulnerability of software or configuration and adopt proper security controls	75%

(Continued)

Table 7: Continued

Control domain	Control objective	Security control	Rate of attainment
Private blockchain block security	The objective is identifying the block storage position of nodes participating in blockchain consensus in order to eliminate the possibility of block data being tampered and ensure the integrity of block data.	Identify all the nodes participating in blockchain consensus Identify the storage position of all the nodes participating in blockchain consensus and adopt proper security controls. Establish access control mechanism for nodes participating in blockchain consensus	95%
Key security for private chain participants	The objective is in order to protect the private keys of all participants. Participants can utilize their own private key to send a transaction, and the private keys should be protected.	Adopt appropriate and valid cryptographic algorithm, and generate and retrieve the keys in a secure way. Establish and document a suite of requirements and procedures for key protection. Establish key security controls, including countermeasure for key lost or destroyed, to mitigate the availability risk cause by Protect keys with hardware secure module	95%
Private blockchain node security	The objective is in order to manage the permissions of the private blockchain nodes and prevent nodes against unauthorized actions.	Identify and document all the nodes participating in consensus Establish permission management system for all nodes to define the valid operation that each node can perform. Assign the corresponding responsibilities for each nodes and properly manage and set up the permissions. Restrict unauthorized nodes from sending transactions.	85%
Private blockchain network security	The objective is in order to protect the network environment of the private blockchain and monitor whether the private blockchain network is abnormal or not in order to ensure the availability of the blockchain network	Establish monitoring mechanism for the private blockchain network, and record and keep the operations in each node. Establish malformed behavior detection for the private blockchain network. If there are massed ineffective transactions sent by malicious nodes, it may lead to the blockchain network denial of service and the proper protection countermeasure should be adopted. Construct IDS or IPS to detect malformed network activities and take action for general network environment. Monitor the network traffic to detect the denial of service and abnormal connection for general network environment. Establish network log system and regularly review the log records.	75%

(Continued)

Table 7: Continued

Control domain	Control objective	Security control	Rate of attainment
		Establish network log protection system. Only a proper permission administrator can access the network logs to prevent the log information from being tampered	
Private blockchain consensus security	The objective is in order to protect the low-level consensus algorithm of the private blockchain, which ensures the correctness of blockchain operation.	The users should confirm the correctness of consensus algorithm adopted by the private blockchain with consensus result. The users should realize the fault tolerance of consensus algorithm adopted by the private blockchain. Taking PBFT as an example, if there are 3m+1 nodes in the blockchain system, it can tolerate at most m nodes as malicious node. The users should realize the minimum activation capability of consensus algorithm adopted by the private blockchain. Taking PBFT as an example, it requires more than 4 nodes to ensure the function of the blockchain system. The users should realize the error handling ability of the consensus adopted by the private blockchain.	80%
Private blockchain security policy	The objective is in order to develop a suite of security policy for regulation aspect from the organization and promote and implement other security control items in the policy.	Establish a set of information security policy managed to the organization's private blockchain application through the management and define the organization's information security objectives and principles.	90%

Table 8: Risk level of permissioned blockchain services

The rate of attainment of each security domain	Risk level	The probability that service being unexpectedly interrupted/terminated during session T	The probability that block being maliciously modified during session T
95% to 100%	High	1%	1%
90% to 95%	Medium	3%	3%
Under the 90%	Low	5%	5%

We have four experiments in which each experiment, containing four nodes, is performed 200 rounds. The first experiment scenario is to investigate the chain availability under the assumption that nodes are independent with each other at a PBFT-based permissioned blockchain. That is, the fault, such as unexpectedly interruption or termination, occurred at a node will not influence the other nodes in the distributed network. The results are as shown in Fig. 4. Note that x-axis is the experiment round and y-axis is session cycles that the target blockchain environment loses its chain-availability.

That is, each experiment round will normally operate session by session until at two least nodes are unexpectedly interrupted/terminated. In the first experiment scenario, the worst case is occurred at the 38th experiment round, where the minimal session cycle that the target blockchain environment can preserve its chain-availability is 1. In addition, the best case is that the target blockchain environment can survive until the 55th session cycle at the 36th experiment round. An average session cycle is 11.545 after 200 experiment rounds are performed.

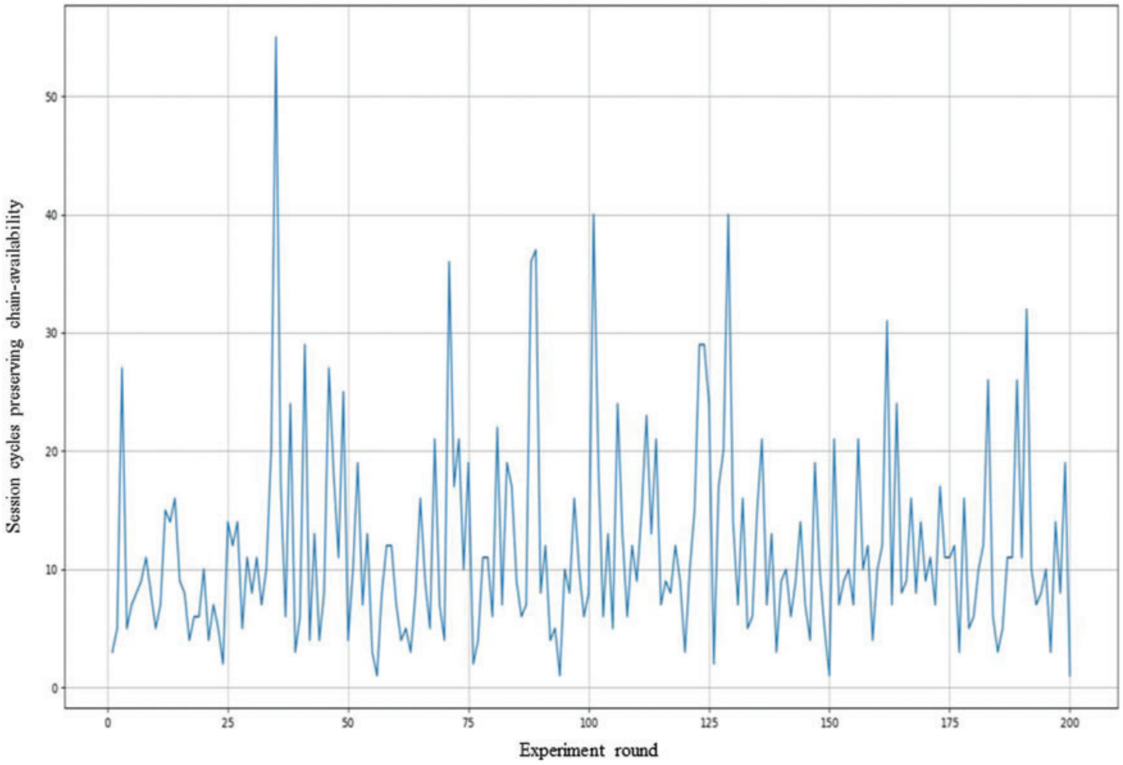

Figure 4: The results of the first experiment: Independence and Chain-availability

The second experiment scenario is to investigate the chain-availability under the assumption that nodes are dependent on a PBFT-based permissioned blockchain. That is, the status of faulty nodes, such as network health and host stability, will influence the other nodes' operation. Accordingly, in our experiment, the dependences among nodes will be calculated through Jaccard similarity coefficient in terms of physical network, network address resolution, and software and hardware configurations as presented in the Tab. 9. Then, the dependences among nodes are as that in Tab. 10. For example, the dependence, i.e., (0, 1, 0.33, 0.33), between Node 3 and Node 2 is conducted as follows: (a) the dependence of physical network of nodes 2 and 3 is 0 since the physical network of nodes 2 and 3 are different, i.e., Domain A and B; (b) the dependence of physical address of nodes 2 and 3 is 100% relevant because they are in the same segment, i.e., Segment A; (c) the dependence of software configuration of nodes 2 and 3 is 0.33 with the equation $\frac{|(S2, S3, S4, S5) \cap (S1, S2, S5, S6)|}{|(S2, S3, S4, S5) \cup (S1, S2, S5, S6)|} = \frac{|(S2, S5)|}{|(S1, S2, S3, S4, S5, S6)|} = 0.33$; and (d) the dependence of hardware configuration of nodes 2 and 3 is 0.33 with the equation $\frac{|(H2, H3, H5, H6) \cap (H1, H2, H3, H4)|}{|(H2, H3, H5, H6) \cup (H1, H2, H3, H4)|} = \frac{|(H2, H3)|}{|(H1, H2, H3, H4, H5, H6)|} =$ 0.33.

Table 9: The status of nodes in the second experiment

	Physical network	Network address	Software configuration	Hardware configuration
Node 1	Domain A	Segment A	(S1, S2, S3, S4)	(H1, H2, H3, H4)
Node 2	Domain A	Segment A	(S2, S3, S4, S5)	(H2, H3, H5, H6)
Node 3	Domain B	Segment A	(S1, S2, S5, S6)	(H1, H2, H3, H4)
Node 4	Domain C	Segment B	(S4, S7, S8, S9)	(H1, H2, H5, H6)

Table 10: The dependencies among nodes in the second experiment

	Node 1	Node 2	Node 3	Node 4
Node 1	(1, 1, 1, 1)	(1, 1, 0.6, 0.33)	(0, 1, 0.33, 1)	(0, 0, 0.14, 0.33)
Node 2	—	(1, 1, 1, 1)	(0, 1, 0.33, 0.33)	(0, 0, 0.14, 0.6)
Node 3	—	—	(1, 1, 1, 1)	(0, 0, 0, 0.33)
Node 4	—	—	—	(1, 1, 1, 1)

In another example, the dependence between Node 4 and Node 1 is (0, 0, 0.14, 0.33), where (a) the dependences of physical network and network address of nodes 4 and 1 are both 0, respectively, because the node 1 is in the Domain A and Segment A and the node 4 is in the Domain C and Segment B; and (b) the dependence of software configuration of nodes 4 and 1 is 0.14 as $\frac{|(S4, S7, S8, S9) \cap (S2, S2, S3, S4)|}{|(S4, S7, S8, S9) \cup (S2, S2, S3, S4)|} = \frac{|(S4)|}{|(S1, S2, S3, S4, S7, S8, S9)|} = 0.14$; and (d) the dependence of hardware configuration of nodes 4 and 1 is 0.33 as $\frac{|(H1, H2, H5, H6) \cap (H1, H2, H3, H4)|}{|(H1, H2, H5, H6) \cup (H1, H2, H3, H4)|} = \frac{|(H1, H2)|}{|(H1, H2, H3, H4, H5, H6)|} = 0.33$ Eventually, the Tab. 11 presented the similarity coefficient among nodes in our experiment in which each coefficient is an average value of dependences among nodes as that in the Tab. 10. Note that the higher of similarity coefficient is, the more relevant and influence between nodes is. Assume that N1N2 is a similarity coefficient of nodes N1 and N2 in a specific session cycle T, the probability that node N2 being unexpectedly interrupted/terminated during session T will be increased as $P \times (1 + P_{N1}P_{N2})$ instead of the original probability P, while node $N1$ is in malfunction. Similar to the first experiment, containing four nodes, we perform 200 experiment rounds and the results are as shown in Fig. 5. Note that x-axis is the experiment round and y-axis is session cycles that the target blockchain environment loses its chain-availability. In the second experiment scenario, the worst case occurs at the 67th experiment round, where the minimal session cycle that the target blockchain environment can preserve its chain-availability is 1. Then, a best case occurs at the 19th experiment round in which the maximum session cycle is 31. An average session cycle is 9.52 after 200 experiment rounds are performed.

Table 11: The similarity coefficient among nodes in the second experiment

	Node 1	Node 2	Node 3	Node 4
Node 1	1	0.7325	0.5825	0.1175
Node 2	—	1	0.415	0.185
Node 3	—	—	1	0.0825
Node 4	—	—	—	1

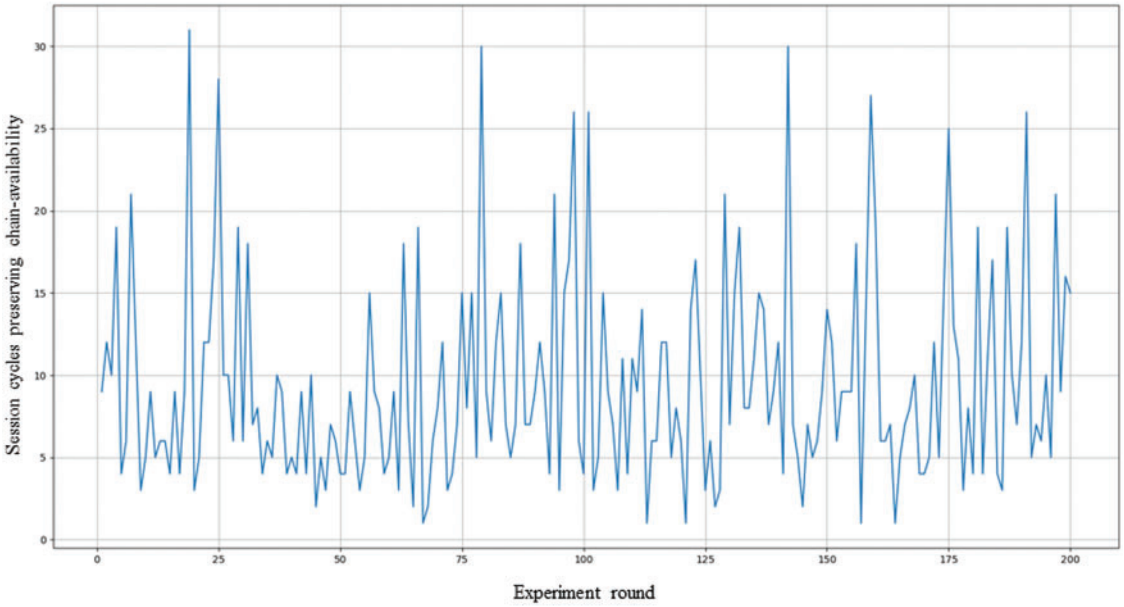

Figure 5: The results of the second experiment: Dependence and Chain-availability

The third experiment is to investigate the chain-integrity under the assumption that nodes are independent at a PBFT-based permissioned blockchain. As shown in the Tab. 8, the probability that block being maliciously modified during session T is 1%, 3% and 5% based on the risk the environment faces. Note that the chain integrity is defined as that, given a specific data block after the consensus of all nodes, more than two third data blocks maintained at their own nodes are not compromised. The experiment results are as shown in Fig. 6, where the x-axis is the experiment round and y-axis is session cycles that the target blockchain environment loss its chain-integrity. The results show that the worst case is at the 182nd experiment round, where the minimal session cycle of preserving chain integrity is 1. And, the best case is with 66 session cycles at the 82nd experiment round. An average session cycle is 20.775 after 200 experiment rounds are performed.

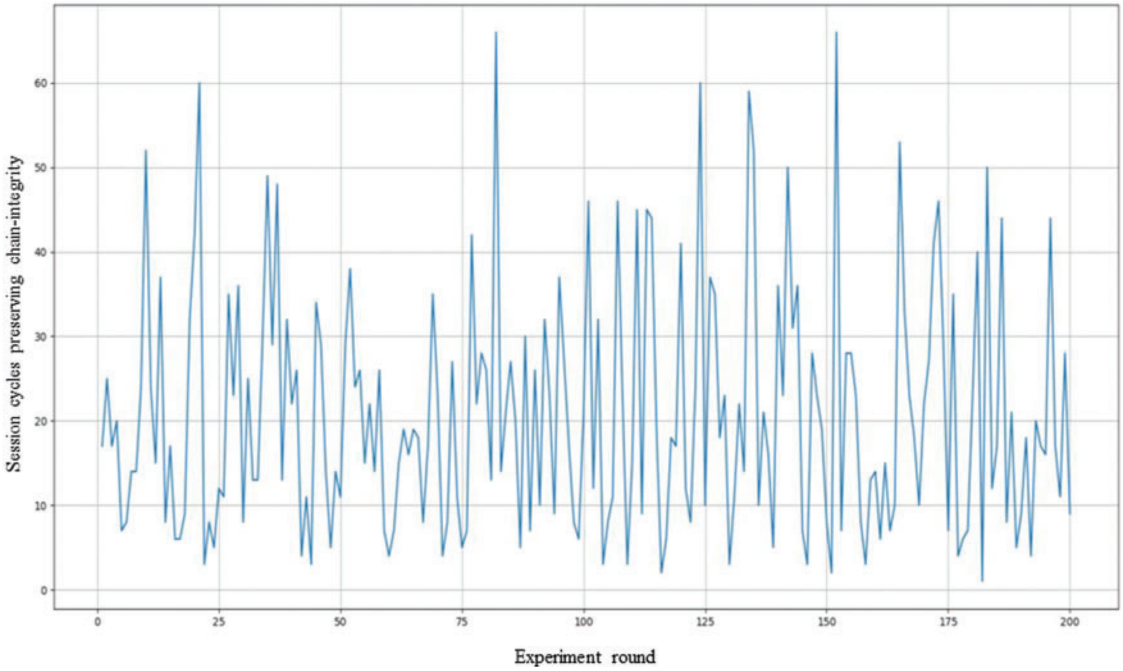

Figure 6: The results of the second experiment: Dependence and Chain-availability

The fourth experiment is to investigate the chain-integrity under the assumption that nodes are dependent at a PBFT-based permissioned blockchain. The status of nodes, dependences among nodes, and similarity coefficient between nodes are presented in the Tabs. 12–14. Note that the computing logic in these three tables as the same as that in our second experiment scenario in which Jaccard similarity coefficient is adopted. In that case, assume that $N1N2$ is a similarity coefficient of nodes $N1$ and $N2$ and N2 in a specific session cycle T, the probability that block at node N2 being maliciously modified during session T will be as $P \times (1 + P_{N1}P_{N2})$ in a specific session cycle T, the probability that block at node $N2$ being maliciously modified during session T will be as $P \times (1 + P_{N1}P_{N2})$ instead of the original probability P, while block at node N1 is being modified maliciously. After performing 200 experiment rounds, the results are as shown in Fig. 7, where the x-axis is the experiment round and the y-axis is session cycles that the target blockchain environment loses its chain-integrity. The results show that the worst case is at the 64th experiment round, where the minimal session cycle of preserving chain-integrity is 1. And, the best case is with 82 session cycles at the 61st experiment round. An average session cycle is 16.81 after 200 experiment rounds are performed.

Table 12: The status of nodes in the fourth experiment

	Network address	Host manager	Software configuration	Hardware configuration
Node 1	Segment A	Manager A	(S1, S2, S3, S4)	(H1, H2, H3, H4)
Node 2	Segment A	Manager B	(S2, S3, S4, S5)	(H2, H3, H5, H6)
Node 3	Segment A	Manager A	(S1, S2, S5, S6)	(H1, H2, H3, H4)
Node 4	Segment B	Manager A	(S4, S7, S8, S9)	(H1, H2, H5, H6)

Table 13: The dependences among nodes in the fourth experiment

	Node 1	Node 2	Node 3	Node 4
Node 1	(1, 1, 1, 1)	(1, 0, 0.6, 0.33)	(1, 1, 0.33, 1)	(0, 1, 0.14, 0.33)
Node 2	—	(1, 1, 1, 1)	(1, 0, 0.33, 0.33)	(0, 0, 0.14, 0.6)
Node 3	—	—	(1, 1, 1, 1)	(0, 1, 0, 0.33)
Node 4	—	—	—	(1, 1, 1, 1)

Table 14: The similarity coefficient among nodes in the fourth experiment

	Node 1	Node 2	Node 3	Node 4
Node 1	1	0.4825	0.8325	0.3675
Node 2	—	1	0.415	0.185
Node 3	—	—	1	0.3325
Node 4	—	—	—	1

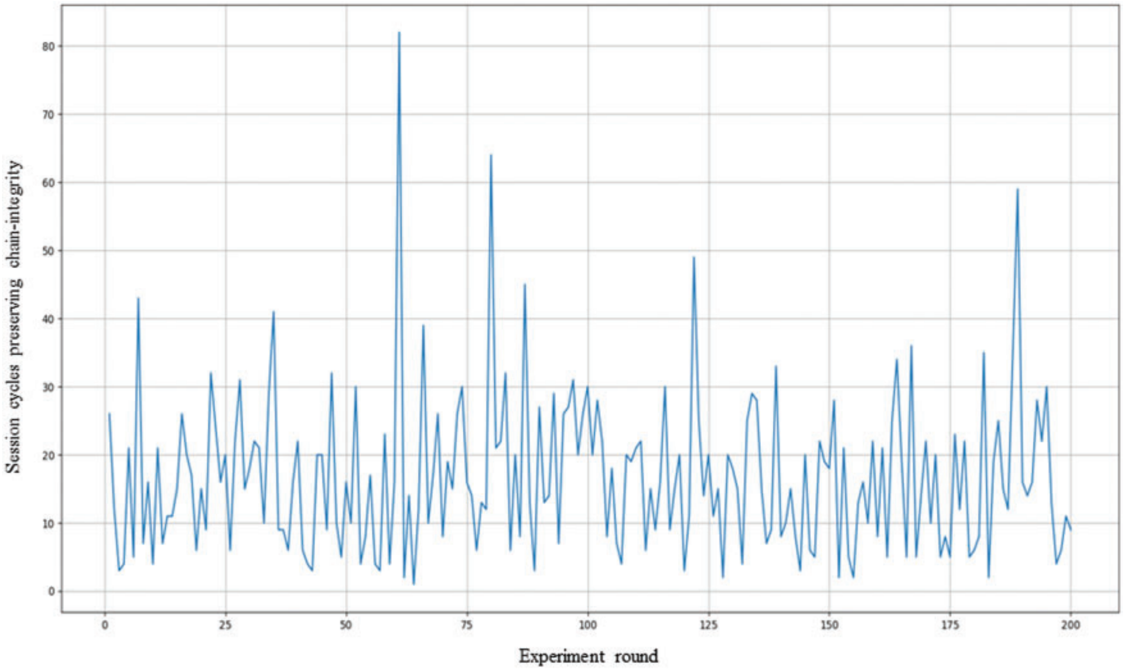

Figure 7: The results of the fourth experiment: Dependence and Chain-integrity

5 Conclusions

This study proposes a security risk management framework for permissioned blockchain applications. Based on the implementation stacks of permissioned blockchain application, the framework defines 6 categories. The framework then provides controls by the categories. The controls can be viewed as the best practices to achieve permissioned blockchain application security. Therefore, application providers can use the framework to perform gap analysis on their existing systems and controls and understand the risks of their applications. The application providers can then follow the controls in the framework to improve security of their existing applications. Furthermore, users can delegate auditors to evaluate the security risks of a permissioned blockchain application to determine whether or not to trust the application. This study maps the controls to existing information security standards and guidelines. The mapping results show that this study is the first research that provides a means to protect security of permissioned blockchain applications from a holistic point of view.

This study has certain limitations that point the way toward future research. First, this study has not validated the framework with real permissioned blockchain applications. While applying the framework to the real-world case, we can find out the framework deficiencies and improve the framework. Second, this study is going to develop checklists based on the framework to help application providers to evaluate their permissioned blockchain applications. The checklists should provide a standard means for application providers or auditors to determine security risks of permissioned blockchain applications. Last but not least, current organizations usually need to follow several information security standards, such as ISO/IEC 27001, ISO/IEC 22301, ISO/IEC 27017, and other standards. Therefore, the proposed framework should consider the integration with existing standards.

Acknowledgement: We thank for the handling editor and anonymous reviewers' comments for making this manuscript better.

Funding Statement: This work was supported by the Ministry of Science and Technology, Taiwan, under grants MOST 110-2218-E-011-007-MBK, MOST 111-2218-E-011-012-MBK, MOST 109-2221-E-011-110-MY2, MOST 109-2221-E-259-011-MY2, MOST 110-2629-E-259-001, MOST 110-2926-I-259-501, and MOST 110-2634-F-A49-004.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Halaburda, "Blockchain revolution without the blockchain?," *Communications of the ACM*, vol. 61, no. 7, pp. 27–29, 2018.

[2] A. McAbee, M. Tummala and J. McEachen, "Military intelligence applications for blockchain technology," in *Hawaii Int. Conf. on System Sciences (HICSS)*, Grand Wailea, Hawali, pp. 6031–6040, 2019.

[3] J. Kolb, M. AbdelBaky, R. H. Katz and D. E. Culler, "Core concepts, challenges and future directions in blockchain: A centralized tutorial," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–39, 2020.

[4] L. Muller, A. Glarner, T. Linder, S. D. Meyer, A. Furrer *et al.,* "Conceptual framework for legal and risk assessment of crypto tokens–Classification of decentralized blockchain-based assets," 2018. [Online]. Available: https://regisbarondeau.com/dl190.

[5] N. Islam, Y. Marinakis, S. Olson, R. White and S. Walsh, "Is blockchain mining profitable in the long run?," *IEEE Transactions on Engineering Management*, pp. 1–14. 2021. https://doi.org.10.1109/TEM.2020.3045774.

[6] N. V. Kuchin and N. G. Butakova, "Vulnerability analysis of corporate blockchain systems to network attacks," in *2021 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus 2021)*, Moscow and St. Petersburg, Russia, pp. 2366–2371, 2021.

[7] S. C. Cha, C. M. Shiung, G. Y. Lin and Y. H. Hung, "A security risk management framework for permissioned blockchain applications," in *2021 IEEE Int. Conf. on Smart Internet of Things (SmartIoT 2021)*, pp. 301–310, 2021.

[8] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato *et al.,* "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[9] Z. Duan, H. Mao, Z. Chen, X. Bai, K. Hu *et al.,* "Formal modeling and verification of blockchain system," in *10th Int. Conf. on Computer Modeling and Simulation (ICCMS 2018)*, New York, USA, 2018.

[10] I. Bashir, "Smart contracts," in *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*, 2nd Revised edition. Birmingham Mumbai: Packt Publishing, 2018.

[11] T. M. Hewa, Y. Hu, M. Liyanage, S. S. Kanhare and M. Ylianttila, "Survey on blockchain-based smart contracts: Technical aspects and future research," *IEEE Access*, vol. 9, pp. 87643–87662, 2021.

[12] M. P. Barrett, "Framework for improving critical infrastructure cybersecurity Version 1.1," Apr. 2018, Accessed: May 06, 2022. [Online]. Available: https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity-version-11.

[13] G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, *Distributed systems: Concepts and design*, 2nd ed., USA: Addison-Wesley Publishing Company, 2011.

[14] D. Yang, J. Gavigan and Z. Wilcox-O'Hearn, "Survey of confidentiality and privacy preserving technologies for blockchains," Technical report R3, 2016.

[15] A. Biryukov, D. Khovratovich and I. Pustogarov, "Deanonymisation of clients in bitcoin P2P network," in *2014 ACM SIGSAC Conf. on Computer and Communications Security (CCS'14)*, New York, USA, pp. 15–29, 2014.

[16] B. D. Win, R. Scandariato, K. Buyens, J. Gr´egoire and W. Joosen, "On the secure software development process: clasp, SDL and touchpoints compared," *Information and Software Technology*, vol. 51, no. 7, pp. 1152–1171, 2009.

[17] M. Howard and D. E. Leblanc, *Writing secure code*, 2nd ed., USA: Microsoft Press, 2002.

[18] J. H. Lee, "Application of the security assessment method to blockchain systems," *Systematic Approach to Analyzing Security and Vulnerabilities of Blockchain Systems*, Master Thesis of the Massachusetts Institute of Technology, pp. 112–118, 2019.

[19] R. A. Mallah and B. Farooq, "Actor-based risk analysis for blockchains in smart mobility," in *3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock '20)*, New York, USA, pp. 29–34, 2020.

[20] R. Zhang, R. Xue and L. Liu, "Security and privacy on blockchain," *ACM Computer Surveys*, vol. 52, no. 3, pp. 1–34, Article No.: 51, 2020.