On Design of Security Risk Management Framework for Permissioned Blockchain Applications

1st Shi-Cho Cha
Dept. of Information Management
National Taiwan University of Science and Technology
Taipei, R.O.C.
Email:csc@cs.ntust.edu.tw

2nd Chuang-Ming Shiung
Criminal Investigation Bureau
Taipei, R.O.C.
Email: saxbear@gmail.com

3rd Gwan-Yen Lin
Dept. of Information Management)
National Taiwan University of Science and Technology
Taipei, R.O.C.
Email: gwanlin4@gmail.com

4th Yi-Hsuan Hung
Dept. of Information Management)
National Taiwan University of Science and Technology
Taipei, R.O.C.
Email: b10709036@gapps.ntust.edu.tw

***Abstract*—As permissioned blockchain becomes a common foundation of blockchainbased applications for current organizations, related stakeholders of the applications need the means to assess the security risks of the applications. To address this issue, this study proposes a security risk management framework for permissioned blockchain applications. The framework divides itself into different implementation stacks and provides guidelines to control the security risks of the permissioned blockchain application. According to the best of our knowledge, this study is the first research that provides a means to evaluate and control 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 hopefully contribute to the adoption of permissioned blockchain technologies.**

***Index Terms*—Permisioned Blockchain, Blockchain Security, Blockchain Risk Evaluation**

I. 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 are reconsidering the value of blockchain applications [4]. Moreover, current blockchain application providers may need to convince their clients that they are not going blockchain for blockchain's sake. In addition, researches shown the propose criteria to decide whether applications are suitable to use blockchain technology. For example, McAbee et al. mentioned critical factors to determine the adoption of blockchain technology in the military intelligence process [6].

This study refers to a blockchain application as an application founded 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. We can classify blockchain networks into public and permissioned blockchains [5]. 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 application 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

This work was supported in part by the Taiwan Ministry of Science and Technology under grants XXXXXXXXXX.

evaluating whether the applications utilize the features of blockchain technology. From a technical perspective, comparing the blockchain technology with existing technologies such as PKI, 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 parties 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 claim that it utilize the above features of its blockchain network, besides the security of associated smart contracts [**?**] [**?**] [**?**] and Web applications [**?**], users are 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 especially important to permissioned blockchain applications. 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 [**?**]. Islam et al. propose to assess the sustainability of blockchain networks on their mining schemes [**?**]. 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 [**?**]. Therefore, a security risk management scheme needs to be in place to help permissoned blockchain application providers to estimate the security risks of their applications and adopt measures to control the risks.

In light of this, this study proposes a security risk management framework for permissioned blockchain applications. The framework provides guidelines to assess the security risks of permissioned blockchain application by major components of the applications and control the risks. On the other hand, users can delegate auditors to use the framework to evaluate the trustworthiness of a permissioned blockchain applications. To the best of our knowledge, this study is the first research that provides a means to assess and control the security risks of permissioned blockchain applications from a holistic point of view. If people can trust the applications that adopted the framework, the paper can hopefully contribute to the adoption of permissioned blockchain technologies.

The rest of this paper is organized as the followings: Section II introduces preliminary information on the permissioned blockchain applications and their strengths or critical features. Section III provides an overview of the proposed framework. Next, Section IV-Section IX summarizes detail controls or security risk estimation items. Conclusions are finally drawn in Section X along with recommendations for future research.

II. Preliminary

A. Implementation Stacks of Blockchain Application

This study follows the blockchain network implementation stack proposed by Wang et al. [7], which is summarized from the model proposed by Duan et al. [3], 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 is called blockchain because it 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.

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 1and 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 kind of cryptographic puzzle based on the values of the block header. A malicious intent person may need a huge amount of computing power to tamper 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. Therefore, 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: Assuming that nodes in a blockchain network have achieved consensus on the first i-1 blocks, in the block generation sub-process. One or more nodes generate the candidate i-th block first based on transactions that have not been encapsulated in existing blocks. Then, 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. Finally, if a node finds that its block data are different from others' data, the

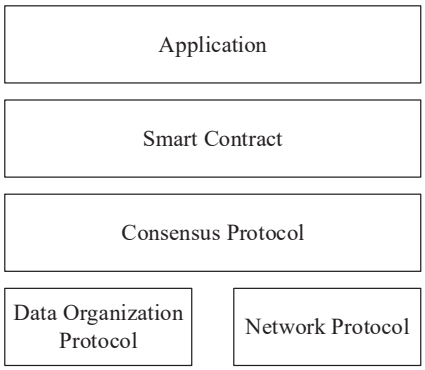

Fig. 1. The blockchain network implementation stacks proposed by Wang et al. [7]

node may initiate the conflict resolution sub-process to ensure data consistency.

Blockchain technology is first introduced to operate with cryptocurrencies like Bitcoin. The *smart contract*technology enhances the blockchain technology by enabling people to enforce a blockchain network to execute autonomous computer programs [1] [**?**]. 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 store the execution results as transaction data in the blockchain network. Consequently, any nodes equipped with the virtual machines can extract the smart contracts and associates versions of state variables and re-execute the smart contract to verify the execution results.

Finally, current blockchain networks usually have their own APIs. People can develop *applications* to send requests to the blockchain networks via the APIs.

B. 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 the previous subsection, a blockchain network has a linked list structure like block data. Also, digests are embedded in the blocks by design for tamper-proof.

Second, users may challenge blockchain technology as they can simply use digital signature technologies to ensure data integrity. In this case, blockchain technology also adopts digital signature technologies. In addition, 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, if 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 keep track of data in a server and request the server to publish data signatures periodically. Therefore, 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 sequential . 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 makes a person validate block data in a blockchain more easily. Moreover, as the ordered blocks are guaranteed, people can assert that a transaction in a block is earlier than transactions in the next block.
- *Relying on 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 section, a blockchain should tolerate partial failures. In addition, even a participating node crashes or meet network-partition failures. Once the node connects 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 can 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,

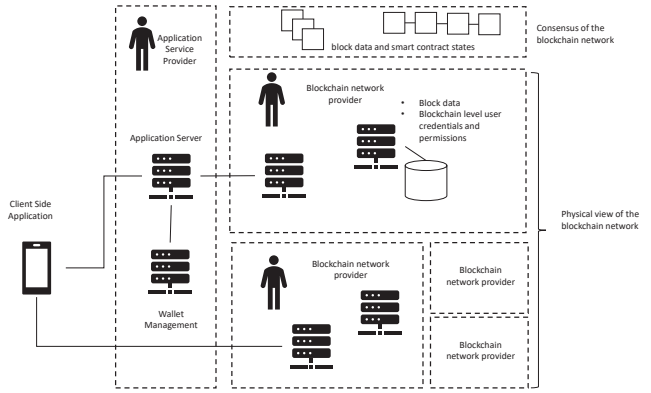

Fig. 2. Model of a Permissioned Blockchain Application in this Study

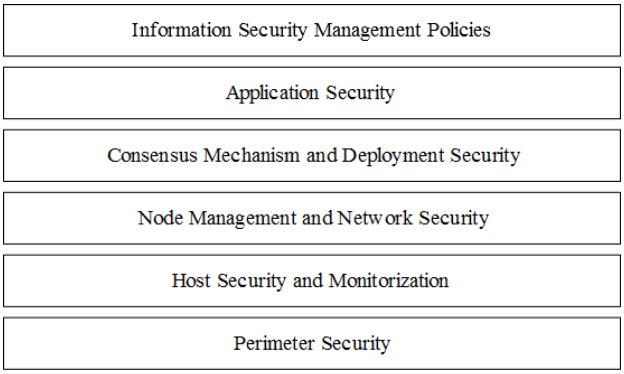

Fig. 3. Major components of the proposed framework

requests people to assess the security risks of participants in the blockchain to determine the trustworthiness of the blockchain.

C. Model of a Permissioned Blockchain Application

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 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.

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.

The application service provider may also provide client-side applications to users. 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.

III. Framework 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 service 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, that 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. Therefore, providers could deploy physical and logical check points 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 monitorization 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 blockchains' fault tolerance ability, 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

TABLE I
LIST OF CONTROLS IN PERIMETER SECURITY CATEGORY

ID	Control	Informative References
4.1	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
4.2	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

awareness, and security testing, this control mainly seeks to identify potential major threats and seeks for the prevention of security breaches.

Finally, the *organizational security* category asks application providers to define procedures to enforce the security of their applications.

IV. PERIMETER SECURITY

Similar to the NIST cybersecurity framework [**?**], 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 Table I, 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 firewall and other network access control mechanisms, to prevent unauthorized users from accessing protected resources (Control 4.1).
- Application providers should prevent unauthorized people from entering protected areas and prevent protected resources from being taking out of protection without permission (Control 4.2).

As illustrated in Table I, application providers can apply existing guidelines or best practices to implement the controls.

V. HOST SECURITY

The *Host Security* category requests application providers to adopt appropriate host-level safeguards to protect participating nodes and associated components. As listed in Table II, 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 5.1). Moreover, application providers can deploy a host-based monitoring scheme to log and identify malicious behaviors.

TABLE II
LIST OF CONTROLS IN HOST SECURITY CATEGORY

ID	Control	Informative References
5.1	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
		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
5.2	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
5.3	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
5.4	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

- Application providers should protect block storage from being tampered with or unauthorized accessed (Control 5.2). 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 5.3). As blockchain applications triggered by user 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 5.4).

TABLE III
LIST OF CONTROLS IN NODE MANAGEMENT AND NETWORK SECURITY CATEGORY

ID	Control	Informative References
6.1	Participating node and privilege management	**PCI-DSS 3.2.1** 1.3.2 1.3.3 12.3.6
		CIS Controls v7.1 13.3
6.2	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

VI. NODE MANAGEMENT AND NETWORK SECURITY

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 Table III, the category includes the following controls:

- 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 capable of removing the node in their permissioned network.
- Application providers should monitor nodes or hosts in their permissioned blockchain applications to detect anomalies (Control 6.2). Compared to the control of logical perimeter security (Control 4.1), the control focuses on the unauthorized or suspicious operations of permitted nodes and the associated components.

VII. CONSENSUS MECHANISM SECURITY

The consensus mechanism is one of the most crucial components of 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 the security of their consensus mechanism. As consensus mechanism is specific in blockchain networks, Table IV does not list references associated with the controls in the category.

- Application providers should identify the consensus algorithms used by their blockchain network and ensure the correctness of the algorithm (Control 7.1). Researchers usually evaluate the correctness of a distributed consensus algorithm with whether the algorithm satisfies the agreement and validation requirements [**?**]. In terms of agreements, it is proving that blockchains 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 proving again that blockchains algorithms under certain degrees of viable error, acceptance can ensure 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 7.2). 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 7.3). 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. For example, if a malicious node in a blockchain network sends a useless number ofs 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 7.5). 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 7.6). Although blockchain technologies can protect data integrity and availability, they do not ensure data confidentiality and privacy [**?**]. 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. [2] have proposed IP traffic analysis schemes to identify the IP addresses of transaction generators and to link

TABLE IV
LIST OF CONTROLS IN CONSENSUS MECHANISM AND DEPLOYMENT SECURITY CATEGORY

ID	Control	Informative References
7.1	Consensus algorithm verification	
7.2	Determine the Byzantine fault-tolerant capability	
7.3	Identify the minimum resource requirement	
7.4	Deal with incidents about the consensus mechanism	
7.5	Segregation of duties	
7.6	Disclose confidentiality and privacy protection mechanism	

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.

VIII. APPLICATION SECURITY

The *application security* category introduces controls to secure the application development cycle. In this case, several software security development lifecycle (SSDLC) guidelines [**?**]. 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 the deputes 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 Table V, the category includes the following controls:

- Application providers should perform threat modeling on their applications to identify risks to the applications (Control 8.1).In this case, we can follow the DFD-based scheme proposed by Howard and Leblanc to model the application and identify the potential attacks based on the STRIDE threat model [**?**] [**?**]. For blockchain applications, Mallah and Farooq propose to evaluate the impact of potential attacks based on monetary loss, privacy, data integrity, and trust [**?**].
- Application providers should define security requirements before developing their applications or making modifications to their applications(Control 8.2). Simply speaking, the control 8.1 requires application providers to identify the threats that their applications should defend against with. Comparatively, control 8.1 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. [**?**], to define their security requirement.
- Application providers should notify the risks of using the applications and protection guidelines to users (Control 8.3). The control can be viewed as a special case of control 8.2. 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 8.4). 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 8.5). 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 tools. In the system testing or acceptance stage, complete system testing should be conducted in the testing environment from vulnerability scanning to penetration testing. (Ref Control 9.11). 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 8.6). Also, the application providers should separate the development, test, and production.
- Application providers should establish appropriate update management mechanisms (Control 8.7). Therefore, as the application providers discover vulnerabilities on their applications, they can update affected components to fix the vulnerabilities.

TABLE V
LIST OF CONTROLS IN APPLICATION SECURITY CATEGORY

ID	Control	Informative References
8.1	Threat modeling	**PCI-DSS 3.2.1** 6.1 12.2
		ISO/IEC 27002:2013 14.1.1 14.2.5
8.2	Identify security requirements	**PCI-DSS 3.2.1** 6.5
		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
8.3	User notification	**ISO/IEC 27002:2013** 9.3
8.4	Code review	**PCI-DSS 3.2.1** 6.5 6.6
		CIS Controls v7.1 18.7
		ISO/IEC 27002:2013 14.2.1
8.5	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
8.6	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
8.7	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

IX. ORGANIZATIONAL SECURITY

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 9.5* 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 9.9), 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 9.1). Documenting information security policies and procedures can reduce ambiguity among associated people and provide the foundation for continuous improvement.

Second, the joint-decision-making (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 (security-related) matters (Control 9.6). 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 9.2). 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 9.4); (2) change and release management procedures (Control 9.7); (3) incident management and business continuity procedures (Control 9.8). Interested people can see the associated standards listed in Table VI

To complete the PDCA 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 9.11). Also, the parties can execute self-checking or build an internal audit program to ensure compliance with the security policies and procedures (Control 9.10). Finally, participating parties of a permissioned blockchain application should learn from past incidents or deficiencies and improve their information management system continuously (Control 9.12)

X. CONCLUSIONS AND FUTURE WORK

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.

TABLE VI
LIST OF CONTROLS IN APPLICATION SECURITY CATEGORY

ID	Control	Informative References
9.1	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
9.2	Define 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
9.3	Perform 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
9.4	Build risk management procedures	**PCI-DSS 3.2.1** 12.2
		ISO/IEC 27001:2013 6.1
9.5	Establish the joint decision-making organization and associated procedures	
9.6	Designate the information security officer	PCI-DSS 12.4.1 12.5
		CIS Controls v7.1 5.3
		ISO/IEC 27002:2013 6.1.1 7.2.1
9.7	Maintain the change management and release management procedures	**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
9.8	Establish the incident management and business continuity procedures	**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
9.9	Ensure 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
9.10	Perform internal auditing and self-check	PCI-DSS 9.1 9.2
		ISO/IEC 27002:2013 12.7.1 15.2.1 18.2.1 18.2.2 18.2.3
9.11	Execute 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
9.12	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

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 discover the framework deficiencies and improve the framework.

Second, this study is going to develop checklists based on the framework in order to help application providers to evaluate whether 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.

REFERENCES

[1] I. Bashir. *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, 2nd Edition*. Expert insight. Packt Publishing, 2018.

[2] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 15–29, New York, NY, USA, 2014. ACM.

[3] Zhangbo Duan, Hongliang Mao, Zhidong Chen, Xiaomin Bai, Kai Hu, and Jean-Pierre Talpin. Formal modeling and verification of blockchain system. In *Proceedings of the 10th International Conference on Computer Modeling and Simulation*, ICCMS 2018, page 231–235, New York, NY, USA, 2018. Association for Computing Machinery.

[4] Hanna Halaburda. Blockchain revolution without the blockchain? *Commun. ACM*, 61(7):27–29, June 2018.

[5] John Kolb, Moustafa AbdelBaky, Randy H. Katz, and David E. Culler. Core concepts, challenges, and future directions in blockchain: A centralized tutorial. *ACM Comput. Surv.*, 53(1), February 2020.

[6] Ashley McAbee, Murali Tummala, and John McEachen. Military intelligence applications for blockchain technology. In *System Sciences (HICSS), 2019 52th Hawaii International Conference on*, pages 6031–6040.

[7] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370, 2019.